Universidad de Granada

Tesis Doctoral

SISTEMA INTELIGENTE DE CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS PARA EL ANÁLISIS Y PREDICCIÓN DE LA MOVILIDAD MEDIANTE SOFT COMPUTING

ANTONIO FERNÁNDEZ ARES

Editor: Universidad de Granada. Tesis Doctorales
Autor: Antonio Jesús Fernández Ares
ISBN: 978-84-1306-326-3
URI: http://hdl.handle.net/10481/57417

UNIVERSIDAD
DE GRANADA

SISTEMA INTELIGENTE DE CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS PARA EL ANÁLISIS Y PREDICCIÓN DE LA MOVILIDAD MEDIANTE SOFT COMPUTING

ANTONIO FERNÁNDEZ ARES

Directores
PEDRO CASTILLO VALDIVIESO
MARÍA ISABEL GARCÍA ARENAS

Programa de Doctorado
TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Departamento de Arquitectura y Tecnología de los computadores
Escuela Técnica Superior de Ingenierías en Informática y Telecomunicaciones
TIC024 - Software Libre para Optimización, Búsqueda y Aprendizaje

Junio 2019

A la memoria de Padre,
mi primer y más duro jefe.

A mi sobrina María del Mar,
porque cuando quiero sonreír
no tengo más que pensar en ti.

RESUMEN

Las ciudades del futuro necesitan fuentes de datos veraces y confiables de las que nutrirse para poder gestionar de forma eficiente sus recursos. Los desplazamientos de los ciudadanos y sus vehículos suponen uno de los aspectos más críticos a abordar y gestionar por las ciudades y sus dirigentes. Sin embargo los métodos existentes para la obtención esta información resultan muy costos de implantar y mantener, son invasivos a la privacidad individual o resultan muy intrusivos con las infraestructuras existentes.

Por otro lado, el abaratamiento y optimización energético/espacial del hardware ha facilitado el desarrollo de la computación ubícua, donde los ordenadores se han incorporado de forma natural a la vida cotidiana de los ciudadanos. Su principal valor de uso explota principalmente su capacidad de comunicación, promovido por el empleo de redes inalámbricas. Esta capacidad se ha visto mercantilizada por los dispositivos denominados inteligentes y sus servicios ofertados, habiendo alcanzando a prácticamente la totalidad de la población y sus pertenencias, dando lugar al Internet de las Cosas.

Esta tesis ofrece una fuente de datos para las ciudades inteligentes capaz de proporcionar información sobre el movimiento de personas y vehículos, basándose en la captación de las comunicaciones inalámbricas emitidas de forma inadvertida por los dispositivos inteligentes portados en el día a día por los ciudadanos. El empleo de esta fuente de datos provee de la información necesaria para que las ciudades sean capaces de adquirir conocimiento mediante la aplicación de técnicas de Soft Computing con el fin de realizar una gestión más eficiente de sus recursos, ya sea prediciendo las necesidades futuras o reaccionando frente a anomalías.

En esta tesis se somete a estudio la viabilidad, veracidad, privacidad, eficiencia, consistencia y aplicabilidad de esta fuente de datos a las ciudades del futuro. Para la realización de dichos estudios se desarrolla un prototipo funcional de sistema de monitorización de bajo coste basado en la fuente de datos propuesta.

PUBLICACIONES

Este trabajo de tesis se avala con los siguientes artículos en revista:

A. J. Fernandez-Ares, A. M. Mora, S. M. Odeh, P. Garcia-Sanchez y M. G. Arenas. "Wireless monitoring and tracking system for vehicles: A study case in an urban scenario". En: *SIMULATION MODELLING PRACTICE AND THEORY* 73 (APR de 2017), 22-42. ISSN: 1569-190X. DOI: {10.1016/j.simpat.2016.11.004}.

A. Fernandez-Ares, A. M. Mora, M. G. Arenas, P. Garcia-Sanchez, G. Romero, V. Rivas, P. A. Castillo y J. J. Merelo. "Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system". En: *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE* 76 (NOV de 2017), 163-179. ISSN: 0167-739X. DOI: {10.1016/j.future.2016.11.021}.

Adicionalmente, se ha publicado un capítulo de libro:

Antonio Fernández-Ares, Antonio Miguel Mora-García, María Isabel García-Arenas, Pablo García-Sánchez, Gustavo Romero, Suhail M Odeh y Pedro A Castillo. "A novel wireless mobility monitoring and tracking system: Applications for smart traffic". En: *International Journal of Conceptual Structures and Smart Applications (IJCSSA)* 4.2 (2016), págs. 55-71.

Las siguientes participaciones en congresos:

M. G. Arenas y col. "Traffic flow forecasting in real world". En: *INTERNATIONAL WORK-CONFERENCE ON TIME SERIES (ITISE 2014)*. 1st International Work-Conference on Time Series (ITISE), Granada, SPAIN, JUN 25-27, 2014. Univ Granada, Fac Sci; Univ Granada, Dept Comp Architecture & Comp Technol; Univ Granada, CITIC. 2014, 1436-1445. ISBN: 978-84-15814-97-9.

Antonio Fernandez-Ares, Maribel Garcia Arenas, Antonio M. Mora, Pedro A. Castillo y J. J. Merelo. "Comparing Wireless Traffic Tracking with Regular Traffic Control Systems for the Detection of Congestions in Streets". En: *SMART CITIES, SMART-CT 2016*. Ed. por Alba, E and Chicano, F and Luque, G. Vol. 9704. Lecture Notes in Computer Science. 1st International Conference on Smart Cities (Smart-CT), Malaga, SPAIN, JUN 15-17, 2016. 2016, 42-51. ISBN: 978-3-319-39594-4; 978-3-319-39595-1. DOI: {10.1007/978-3-319-39595-1_5}.

Antonio Fernandez-Ares, Maria Garcia-Arenas, Pedro A. Castillo y Juan J. Merelo. "Impact of Protests in the Number of Smart Devices in Streets: A New Approach to Analyze Protesters Behavior". En: *SMART CITIES*. Ed. por Alba, E and Chicano, F and Luque, G. Vol. 10268. Lecture Notes

in Computer Science. 2nd International Conference on Smart Cities (Smart-CT), Malaga, SPAIN, JUN 14-16, 2017. 2017, 75-85. ISBN: 978-3-319-59513-9; 978-3-319-59512-2. DOI: {10.1007/978-3-319-59513-9_8}.

Aunque no estén directamente relacionados con la temática de la tesis, las técnicas y procedimientos aprendidos han sido también empleados en artículos de otras líneas de investigación del autor:

Antonio Fernández-Ares, AM Mora, Pablo García-Sánchez, Pedro A Castillo y JJ Merelo. "Analysing the influence of the fitness function on genetically programmed bots for a real-time strategy game". En: *Entertainment Computing* 18 (2017), págs. 15-29.

Antonio M. Mora, Antonio Fernandez-Ares, Juan J. Merelo, Pablo Garcia-Sanchez y Carlos M. Fernandes. "Effect of Noisy Fitness in Real-Time Strategy Games Player Behaviour Optimisation Using Evolutionary Algorithms". En: *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY* 27.5, SI (SEP de 2012), 1007-1023. ISSN: 1000-9000. DOI: {10.1007/s11390-012-1281-5}.

Como colaboración en artículos de revista de otros investigadores[1]:

Pedro A. Castillo, Antonio M. Mora, Hossam Faris, J. J. Merelo, Pablo Garcia-Sanchez, Antonio J. Fernandez-Ares, Paloma De las Cuevas y Maria I. Garcia-Arenas. "Applying computational intelligence methods for predicting the sales of newly published books in a real editorial business management environment". En: *KNOWLEDGE-BASED SYSTEMS* 115 (JAN 1 de 2017), 133-151. ISSN: 0950-7051. DOI: {10.1016/j.knosys.2016.10.019}.

P. de las Cuevas, A. M. Mora, J. J. Merelo, P. A. Castillo, P. Garcia-Sanchez y A. Fernandez-Ares. "Corporate security solutions for BYOD: A novel user-centric and self-adaptive system". En: *COMPUTER COMMUNICATIONS* 68.SI (SEP 1 de 2015), 83-95. ISSN: 0140-3664. DOI: {10.1016/j.comcom.2015.07.019}.

Adicionalmente, en otras líneas de investigación no directamente relacionadas con la tesis, se han publicado los siguientes trabajos en congresos:

A. Fernandez-Ares, P. Garcia-Sanchez, A. M. Mora, P. A. Castillo y J. J. Merelo. "There Can Be only One: Evolving RTS Bots via Joust Selection". En: *APPLICATIONS OF EVOLUTIONARY COMPUTATION, EVOAPPLICATIONS 2016, PT I*. Ed. por Squillero, G and Burelli, P. Vol. 9597. Lecture Notes in Computer Science. 19th European Conference on the Applications of Evolutionary Computation (EvoApplications), Porto, PORTUGAL, MAR 30-APR 01, 2016. Edinburgh Napier Univ, Inst Informat & Digital Innovat; EvoCOMNET Track, World Federat Soft Comp;

1 ↑La inclusión de estos artículos es únicamente a nivel curricular. Ninguno de las partes de dichos artículos son empleados en ninguna de las partes de esta tesis Doctoral.

Camara Municipal Porto; Turismo Porto; Univ Coimbra. 2016, 541-557. ISBN: 978-3-319-31204-0. DOI: {10.1007/978-3-319-31204-0_35}.

A. Fernandez-Ares, A. M. Mora, J. J. Merelo, P. Garcia-Sanchez y C. Fernandes. "Optimizing Player Behavior In A Real-Time Strategy Game Using Evolutionary Algorithms". En: *2011 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC)*. IEEE Congress on Evolutionary Computation. IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, JUN 05-08, 2011. IEEE; IEEE Computat Intelligence Soc. 2011, 2017-2024. ISBN: 978-1-4244-7835-4.

A. Fernandez-Ares, A. M. Mora, J. J. Merelo, P. Garcia-Sanchez y C. M. Fernandes. "Optimizing Strategy Parameters in a Game Bot". En: *ADVANCES IN COMPUTATIONAL INTELLIGENCE, IWANN 2011, PT II*. Ed. por Cabestany, J and Rojas, I and Joya, G. Vol. 6692. Lecture Notes in Computer Science. 11th International Work-Conference on Artificial Neural Networks (IWANN), Torremolinos, SPAIN, JUN 08-10, 2011. Univ Malaga; Univ Granada; Univ Politecnica Catalunya. 2011, 325-332. ISBN: 978-3-642-21498-1.

Antonio Fernández-Ares, Pablo García-Sánchez, Antonio Miguel Mora, Pedro A Castillo y JJ Merelo. "There can be only one: Evolving rts bots via joust selection". En: *European Conference on the Applications of Evolutionary Computation*. Springer, Cham. 2016, págs. 541-557.

Antonio Fernandez-Ares, Antonio M. Mora, Maribel Garcia-Arenas, Juan Julian Merelo Guervos, Pablo Garcia-Sanchez y Pedro A. Castillo. "Co-Evolutionary Optimization of Autonomous Agents in a Real-Time Strategy Game". En: *APPLICATIONS OF EVOLUTIONARY COMPUTATION*. Ed. por EsparciaAlcazar, AI. Vol. 8602. Lecture Notes in Computer Science. 17th European Conference on Applications of Evolutionary Computation (EvpApplications), Granada, SPAIN, APR 23-25, 2014. Univ Granada, Free Software Off; Granada Excellence Network Innovat Lab; Edinburgh Napier Univ, Inst Informat & Digital Innovat; World Federat Soft Comp. 2014, 374-385. ISBN: 978-3-662-45523-4; 978-3-662-45522-7. DOI: {10.1007/978-3-662-45523-4_31}.

Pablo Garcia-Sanchez, Antonio Fernandez-Ares, Antonio M. Mora, Pedro A. Castillo, Jesus Gonzalez y Juan Julian Merelo Guervos. "Tree Depth Influence in Genetic Programming for Generation of Competitive Agents for RTS Games". En: *APPLICATIONS OF EVOLUTIONARY COMPUTATION*. Ed. por EsparciaAlcazar, AI. Vol. 8602. Lecture Notes in Computer Science. 17th European Conference on Applications of Evolutionary Computation (EvpApplications), Granada, SPAIN, APR 23-25, 2014. Univ Granada, Free Software Off; Granada Excellence Network Innovat Lab; Edinburgh Napier Univ, Inst Informat & Digital Innovat; World Federat Soft Comp. 2014, 411-421. ISBN: 978-3-662-45523-4; 978-3-662-45522-7. DOI: {10.1007/978-3-662-45523-4_34}.

J. J. Merelo, Pedro Castillo, Israel Blancas, Gustavo Romero, Pablo Garcia-Sanchez, Antonio Fernandez-Ares, Victor Rivas y Mario Garcia-Valdez. "Benchmarking Languages for Evolutionary Algorithms". En: *APPLI-*

CATIONS OF EVOLUTIONARY COMPUTATION, EVOAPPLICATIONS 2016, PT II. Ed. por Squillero, G and Burelli, P. Vol. 9598. Lecture Notes in Computer Science. 19th European Conference on the Applications of Evolutionary Computation (EvoApplications), Porto, PORTUGAL, MAR 30-APR 01, 2016. Edinburgh Napier Univ, Inst Informat & Digital Innovat; EvoCOMNET Track, World Federat Soft Comp; Camara Municipal Porto; Turismo Porto; Univ Coimbra. 2016, 27-41. ISBN: 978-3-319-31153-1. DOI: {10.1007/978-3-319-31153-1_3}.

J. J. Merelo, Zeineb Chelly, Antonio Mora, Antonio Fernandez-Ares, Anna I. Esparcia-Alcazar, Carlos Cotta, P. de las Cuevas y Nuria Rico. "A Statistical Approach to Dealing with Noisy Fitness in Evolutionary Algorithms". En: *COMPUTATIONAL INTELLIGENCE, IJCCI 2014*. Ed. por Merelo, JJ and Rosa, A and Cadenas, JM and Dourado, A and Madani, K and Filipe, J. Vol. 620. Studies in Computational Intelligence. 6th International Joint Conference on Computational Intelligence (IJCCI), Rome, ITALY, OCT 22-24, 2014. Inst Syst & Technologies Informat, Control & Commun; IEEE Computat Intelligence Soc; Int Federat Automat Control; ACM Special Interest Grp Artificial Intelligence; Assoc Advancement Artificial Intelligence; Asia Pacific Neural Network Assembly; European Soc Fuzzy Log & Technol; Int Neural Network Soc; Int Fuzzy Syst Assoc. 2016, 79-95. ISBN: 978-3-319-26393-9; 978-3-319-26391-5. DOI: {10.1007/978-3-319-26393-9_6}.

JJ Merelo, Pedro Castillo, Israel Blancas, Gustavo Romero, Pablo García-Sanchez, Antonio Fernández-Ares, Víctor Rivas y Mario García-Valdez. "Benchmarking languages for evolutionary algorithms". En: *European Conference on the Applications of Evolutionary Computation*. Springer, Cham. 2016, págs. 27-41.

Juan J. Merelo y col. "The Uncertainty Quandary: A Study in the Context of the Evolutionary Optimization in Games and Other Uncertain Environments". English. En: *TRANSACTIONS ON COMPUTATIONAL COLLECTIVE INTELLIGENCE XXIV*. Ed. por Nguyen, NT and Kowalczyk, R and Filipe, J. Vol. 9770. Lecture Notes in Computer Science. GEWERBESTRASSE 11, CHAM, CH-6330, SWITZERLAND: SPRINGER INTERNATIONAL PUBLISHING AG, 2016, 40-60. ISBN: 978-3-662-53525-7; 978-3-662-53524-0. DOI: {10.1007/978-3-662-53525-7_3}.

Oráculo: ¿Qué es esto?
Neo: Una ristra de morcillas.
Oráculo: No. No. Son unos y ceros, pero mu bien curados y mu bien colocaos.

— ¿Matrix?

AGRADECIMIENTOS

Durante mucho tiempo estudié la posibilidad de poner un espejo en esta página, de forma que cualquiera que leyese esta tesis sintiese mi agradecimiento. Si estás aquí, mirando esto, muchas gracias. Ya hayas hecho más fácil o más difícil mi existencia, habré aprendido algo gracias a ti. Es por ello que gracias. He intentado que no haya nombres en estos agradecimientos para no olvidarme explícitamente de nadie. He intentado que todo el mundo, tenga una línea en la que sentirse acogido. Muchos en más de una.

A mis directores de tesis:

Que han sabido lidiar con mi cabezonería manifiesta y me han permitido, en mayor o menor medida, hacer lo que he querido como he podido. He intentado, como es imperativo en mi existencia, ser el estudiante de doctorado que el día de mañana me gustaría tener. Espero haber estado a la altura.

A mi familia:

Tanto la de sangre como la política. Os pido perdón por mi ausencia, por mi cabeza en las nubes y por mis sermones intentando explicar mi investigación.

Te pido perdón sobri, por no haberte podido ver tanto como a ambos nos hubiese gustado. Te quiero más de lo que nunca pensé que podría quererte.

A mis hermanos, por ser el raro y el friki de la familia y por los conflictos que ello os haya podido causar, os quiero y admiro más de lo que pensáis.

A mi madre, por que ser mi madre no ha tenido que ser fácil, gracias por enseñarme a valerme por mi mismo y no depender nunca de nadie.

A mis cuñadas, tanto a la que considero parte arterial de mi familia como a la que poco a poco estoy conociendo, os pongo en bandeja de plata los chistes de cuñado que sin duda espero que hagáis siempre a mi costa.

A mi familia arterial, gracias por acogerme y hacerme sentir uno más de la familia sin serlo.

Al resto de mi familia, para la cual estoy fuera del radar desde hace tiempo y si saben que existo es por los detalles que les mando cuando algo reseñable les acontece. Me tenéis siempre aquí, aunque esté ausente.

A aquellos que me regalaron un *GENIO 3000* por mi comunión, que dudo que sospechasen que lo que más me gustaba era el intérprete de *BASIC*.

A las mascotas a mi alrededor, que son como de la familia. Lo cual incluye a Canela Trauma, a Codi el perro, a Arco Iris Marshal Estrella y a Fito.

Finalmente, a mi padre. Gracias por forjar la persona en la que me he convertido. Lamento en el alma que no hayas podido ver los pocos logros que he cumplido. Aprendí de ti tanto lo que quería como lo que no quería ser en la vida; no considero que haya lección más importante. Nunca entendiste que lo yo hacía todo el día delante de un ordenador, pese a tu prohibición, no era jugar, sino aprender. Aún a día de hoy, nunca he dejado da aprender.

A mis amigos:

A los pocos que os puedo contar con los dedos de una mano (¿en binario?) y os considero casi de mi familia.

A los que a pesar de estar tan lejos os siento siempre tan cerca.

A los que a pesar de estar tan cerca, la rutina nos mantenga tan lejos.

A los que os llevo diciendo que iré a veros cuando termine la tesis, desde que empecé la tesis. Me quedé sin escusa. Ahora si que si, que quiero conocer a vuestras casas y mascotas.

A los que aunque nos veamos poco, cuando nos juntamos es como si nos siguiésemos viendo todos los días.

A los compañeros de carrera, que lo pasamos tan bien y tan mal juntos que eso une. ¿Alguien dijo TEPT?

A los inumerables compañeros de piso que he tenido. Gracias por aguantarme. Perdón por ser un desastre. Los que me consideréis amigo, os toca aguantarme, que no os pido perdón por ser un desastre.

A los compañeros del rol, porque ser un vampiro edonista o un guerrero zumbao era increiblemente divertido a vuestro lado.

A la elite sonyer, a pesar de que tenga la consola cogiendo polvo.

A aquellos que gritan *Lok'tar ogar* alzando armas junto a Green Jesus.

A quien el día de mañana me aguante, que tiene mérito. Que chica, ya puedes ir apareciendo carajo, que ahora tengo algo más de tiempo libre.

Al grupo de investigación TIC-24:

Gracias por acogerme, por enseñarme, por aguantarme[2] y permitirme haber estado aquí sin ser el más brillante.

Gracias al patriarca del grupo, por confiar en mi para la beca de iniciación a la investigación. Y por mostrarme que hay alguien más cabezón que yo.

Gracias a mis hermanos mayores de la investigación. Gracias por dejarme seguir vuestros pasos. Sin vosotros no hubiese sido posible nada de esto. Todo lo que sé sobre la investigación lo he aprendido de vosotros. Os debo tanto, que nunca podré sentir pagada mi deuda. Os prometo que volveré a los juegos con vosotros.

2 ↑Lo de aguantaros yo a vosotros, ya lo hablamos otro día :P

Gracias a mi tíos lejanos de la investigación, a vosotros que a veces me iba a vuestros despacho a que me aconsejarais y pediros opinión, sin que tuvieseis ninguna obligación. A los que os veía poco pero siempre aprendía algo de vosotros.

Gracias a mis primillos de investigación, a los que habéis ido y venido. Con los que he compartido el juego de aprender a investigar. A los que espero ver pronto en esta misma situación.

Gracias a los padres de mi línea de investigación. Espero haber puesto el esfuerzo suficiente en ella. Espero que os sintáis orgullosos tanto de mi, como de mi trabajo. Llegué a ella casi de casualidad, sin pretenderlo y al instante la hice mía. Gracias por brindarme tal diamante en bruto.

A mis compañeros del D1.7 (que tiene líneas temporales):

A mis compañeros originales del *D1.7*. No podéis imaginar lo que aprendí de vosotros. Os miraba y os veía como caballeros de brillante armaduras y afiladas espadas vencedores de mil y una batallas. Yo me sentía como un niño con un palo. Gracias por tratarme siempre como a un igual. Gracias por brindarme vuestros conocimientos, vuestros consejos y vuestra experiencia. Gracias por ser el reflejo en el espejo de lo que yo quería ser.

A mis compañeros del *D1.7* de la era llamada *De las Guerras Nerfs*. Nunca pensé que podría trabajar tanto en un sitio y pasarlo tan bien al mismo tiempo. Por esas horas que hemos echado currando como cabrones por imitación del entorno. Por esas risas que se oían desde la conserjería. Por esa escalada armamentística *nerf* que se nos fue totalmente de las manos, pero nos brindó tantos momentos gloriosos.

A mis compañeros del *D1.7* de la era posterior llamada la era *Pokemon GO*. Perdón por mi lío de mesa y mi barrera protectora de cacharros a mi alrededor. Gracias por entender que trabajar con hardware tiene estas cosas. Gracias por soportarme en mis momentos más oscuros. Gracias por las canciones, los dibujos en la pizarra, las chorradas en los grupos y las salidas a las incursiones.

A mis últimos compañeros del *D1.7*. Gracias por permitirme la osadía de intentar aconsejaros. Gracias por tener en cuenta mi opinión y buscarla. Gracias por hacerme sentir, sin merecerlo, como un caballero de brillante armadura.

A mis compañeros de trabajo:

Gracias a mi jefe en Lenguaje y Sistemas. Siempre lamenté que fuesen causas externas y de salud la que nos separasen. Gracias por enseñarme que podía aportar algo en la Universidad.

Gracias a mis jefes y compañeros en el área de Comunicaciones en Sistemas de cierta entidad bancaria que ya no existe. Fuisteis mi primer trabajo serio. Seguí vuestro consejo y olvidé todo lo que había aprendido de redes en la carrera, nunca os confesé lo sencillo que era. Gracias por intentarlo. Gracias

por adoptarme y enseñarme a que os dedicabais cada uno de vosotros, aunque escapase a mis competencias. Gracias por la confianza depositada.

Gracias a mis jefes y compañeros del antiguo CEVUG. Llegué a vosotros por casualidad, y todos hicimos lo posible porque no me fuese. Gracias por enseñarme mi vocación de docente, de casualidad. Gracias por arroparme, sin saberlo, en uno de mis mayores momentos de desamparo. Gracias por todo lo que aprendí de vosotros. Gracias por atarme a Granada.

Gracias a los profesores de Arquitectura y Tecnología de los Computadores. Yo era del grupo de investigación minoritario, pero siempre he encontrado todas vuestras puertas abiertas cuando he necesitado algo. Gracias por permitirme estar, en cierta manera, allí.

Gracias a los mal llamados *becarios* del departamento, aunque alguno peine ya acreditación de titular y tenga currículum más potente que Saitama. Yo no era más que el ocasional, pero siempre tuve un hueco en vuestra mesa. Siempre sentí que podía formar parte de vosotros.

Gracias a los profesores de Teoría de la Señal, Telemática y Comunicaciones. O bueno, debería decir compañeros, no me acostumbro a esto. O bueno, más bien ex-compañeros, o igual he vuelto, quien sabrá lo que me deparará el futuro para cuando lea la tesis. Gracias por no rechazarme como sustituto. Gracias por acogerme y brindarme la oportunidad de dar clase en esta Universidad a la que tanto apego tengo. Gracias por permitirme brevemente cumplir un sueño. Gracias.

A la ETSIIT, la UGR y demás organismo institucional:

A los mandamases de la *ETSIIT*, por liarme tantas veces para participar en eventos de difusión y divulgación. Gracias por permitirme tirar agua sobre la mesa para inculcarle a los chavales el pensamiento algorítmico.

A los mandamases del *CITIC*, por no echarme de allí durante tantos años. Gracias porque aún cuando os habéis visto obligado a echarme, me habéis permitido tener un laboratorio donde tener todos mis trastos.

A los mandamases del programa de doctorado. Gracias por tener la puerta siempre abierta. Gracias por encontrar soluciones a todos los problemas. Gracias por una labor invisible pero ardua.

Al personal de conserjería y limpieza, los que concentran todo el poder de la universidad. ¡Les he visto echar a catedráticos de sus despachos! Gracias por todo. Por entender mi caos. Por la cantidad de paquetes que recibo. Ahora os confieso que no todos ellos son por trabajo. La última antena, era en realidad una espada de Juego de Tronos.

Al personal de cafetería, que llevo 15 años ocupándole un sitio sin consumir nunca nada, porque siempre voy de compañía. Gracias por permitirlo. Espero que sea por muchos años.

Al personal de comedores, por echarme de comer durante tantos años. En serio, llevo 15 años de platos alpujarreños y sopas de mi pueblo (aunque cada vez tiene que ser un pueblo distinto, porque nunca vuelve a ser lo mismo).

Al personal de seguridad, por no detenerme ni apalearme a pesar de mis pintas agravadas por mi paraguas-katana o el andar subido a escaleras manipulando cajas estancas del sótano. Perdón por saltar la valla del CITIC de madrugada en tantas ocasiones.

A la antigua y a la nueva *OSL*. Igual no siempre coincidamos (escribió el doctorando en Sublime Text), pero gracias por todo lo que habéis intentado enseñarme.

Organismos e instituciones que han permitido poner nodos de monitorización:

Gracias a la Dirección General de Tráfico por permitirnos colocar los nodos. A sus empleados, por esas tostadas de jamón que había que cortar cinco veces para comerlas después de una dura mañana de averiguar porque algo no funcionaba; por enseñarme tanto y tratarme siempre con cariño. Ambos nos quedamos con las ganas de poner más cacharros en las alturas. Espero, que tan solo sea de momento.

Gracias al Área de Movilidad del Ayuntamiento de Granada y a la empresa *ACISA* por permitirnos poner los nodo en la ciudad de Granada. Por abrirnos hasta los semáforos para ello. Por permitirnos tanto, a pesar de ofrecer nosotros tan poco.

Gracias a la *ETSIIT* por permitir hacer las primeras pruebas en ella. Muchos bedeles aún creen que era broma lo de que el cacharro que estaba poniendo servía para espiarles. ¡Sorpresa!

Gracias al *CITIC* por permitirme mantener un nodo allí escondido. El día que me ponga a cruzar datos, habrá una prueba fehaciente de la cantidad de horas que allí se trabaja.

Gracias a *Nazaries IT* por facilitarnos las comunicaciones `M2M`. Siento no haber cumplido con las promesas que se os hicieron. Gracias por darnos más incluso de los que nos merecíamos.

Profesores:

A mi profesor de primaria, por entender que aquellos galimatías que escribía en papel era el código *BASIC* que luego usaría para comprobar si me salían bien los ejercicios de matemáticas. Y en lugar de prohibirlo, incentivarlo.

A mi profesor *el Grande*, en serio, si eres el tutor de un grupo no puedes convertir todas tus horas en Historia. Gracias por dejarme tanto tiempo para pensar en mis cosas.

A los profesores de mi instituto que me dijeron que nunca llegaría a nada academicamente hablando, con especial mención a mi profesora de Lengua y Literatura. Gracias por decirme que no podía hacerlo. Gracias por humillarme porque no podía hacerlo. Gracias por lo fuerte que mi hicistéis para poder hacerlo.

A la orientadora que me dijo que no estudiase informática, que me fuese a trabajar a una tienda vendiendo ratones y esas cosas que me gustaban.

A algunos profesores de la universidad que marcaron un antes y después en mi carrera. Ya hablasen de pájaros caídos, me perdieran exámenes, me llevasen tocando la guitarra a la velada musical, me buscasen en los foros, se me quedasen dormidos en tutorías o hablasen detrás de la foto de Wario. Sin vosotros, nada de esto hubiese sido posible.

Por motivos totalmente opuestos, gracias también a los que se negaban a reconocer sus errores, a los que desconocen que es una tabla hash, a los que el mismo ejercicio a lo largo de los años le da resultado distintos, a los que te piden preguntas de desarrollo a responder en un cuadrado de 1cm de alto o a los que se presentan como adalides de las matemáticas cuyo cometido divino es impedir que surjan ingenieros.

A los que les debo salud:

A los posaderos de la Posada, por esas tapas sazonadas de auténtico cariño que tanto hacen por nuestra salud. Y también la mental.

A Mercadona y al señor Hacendado, sustento de mi alimentación. Por sus empleados que se preocupan por mi alimentación, cuando he tenido que comer rápido y mal o me llevado latas de `BURN` a diario.

A Domino's Pizza, Burguer King, Mc Donald's, el Kebab de la esquina,... por hacer que hubiese más yo. Aunque igua a los 110 kilos tendríamos que haber parado. Y no, otra empresa con la que solía hacer locuras los martes, tú ya no te mereces mi agradecimiento.

Al Gimnasio Beone y sus monitores. Por hacer que haya menos yo, algo más de 40 kilos menos de yo. Y ahora, nuevamente, cada vez más de mi, pero en durito. Gracias por saber cuando exigirme. Gracias por saber cuando pararme. Gracias por tener siempre una cara amable. Gracias por sacarme al final del día una sonrisa. No olvidaré el día que casi muero allí vestido de Batman.

Mención aparte merecen la sauna y la piscina del Beone, que me han mantenido siempre a flote. Si la gente supiese cuantas vidas han salvado indirectamente, las nominarían al Nobel de la Paz.

Casi me olvido de la gente de comedores de la universidad. Gracias de nuevo. Sigo sin saber de que pueblo es la sopa.

Gentecilla en general:

A todos aquellos que he conocido en congresos y saraos varios.

A todo aquel con el que haya estado más de una vez en un bar de tapas.

A todos mis alumnos, que soportan con entereza mi capacidad de hablar sin cansarme. Que les suelto cada chapa...

Al indigente que pide en la esquina del materno, al que saludo todas y cada una de las mañanas.

Ocio y cosas random:

Gracias *CD Projekt Red*, *Blizzard* y *Bethesda* por hacer esos mundos de fantasía en los que me he perdido cuando necesitaba desconectar.

Bioware, a ti no te perdono lo del último *Mass Effect*.

Game Freak tu contento me tienes con el último Pokémon.

A *Niantic*, por hacerme perder en el mundo real para poder capturar un Charmander por las calles de Graná.

A *Stan Lee* directa e indirectamente por las películas de Marvel.

A *Tony Stark* alias *Iron Man*, te quiero 3000.

A *Emma Watson*, te he dejado una nota en la segunda página.

A *R.R.Martin*. Yo ya he terminado la tesis, ahora te toca a ti terminar los libros. Arregla el desastre que han hecho *D&D*. ¡Demando juicio por combate! Dile a *Emilia Clarke* que tiene una nota en la segunda página.

A *David Broncano* y *Berto Romero*, por las horas de risa que me habéis proporcionado, aunque os vea por Youtube y no pilléis ni un duro.

A mi cabezonería. Mi principal virtud. Mi mayor defecto. Mi combustible y mi kriptonita.

A quien herede mi legado / le caiga el marrón:

Por último, a alguien que no sé si siquiera llegará a existir, pero es mi única esperanza de que alguien lea esta tesis entera: quien continúe mi línea de investigación en el grupo. Espero que todo el conocimiento, trabajo y esfuerzo puesto en esta tesis te sirva de algo.

ÍNDICE GENERAL

ÍNDICE DE EXPERIMENTOS Y ESTUDIOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ALGORITMOS Y CÓDIGOS

LISTA DE TÉRMINOS

B

BOARD 18
En el ámbito de la computación ubícua, dispositivo electrónico de grandes dimensiones interactuable por varias personas.

D

DISPOSITIVO INTELIGENTE 7
Ver smartdevices.

I

INTERNET OF THINGS (IoT) 19
Paradigma basado en el que las cosas o `things` pueden comunicarse y colaborar entre ellas mediante su conexión a un red.

P

PAB 18
En el ámbito de la computación ubícua, dispositivo electrónico del tamaño de la palma de la mano.

S

SMARTCAR 4
Automóvil inteligente.

SMARTCITY 4, 17, 29, 32, 33
Ciudad inteligente, ver Sección 2.2 para una descripción formal.

SMARTPHONE 4
Teléfono móvil inteligente.

SMARTWATCH 4
Reloj inteligente.

T

TAB 18
En el ámbito de la computación ubícua, dispositivo electrónico de escasos centímetros que puede ser llevado encima o vestido.

W

WPAN 72
Personal Area Network (PAN), Red de Área Personal, es una red de computadoras para la comunicación entre distintos dispositivos cercanos al punto de acceso..

WWAN 72
Una red de área local inalámbrica, también conocida como WLAN (del inglés wireless local area network), es un sistema de comunicación inalámbrico para minimizar las conexiones cableadas..

Parte I

Introducción y Revisión Bibliográfica

1

INTRODUCCIÓN

Cualquier tecnología avanzada es equivalente a la magia.

— Sir Arthur C. Clarke

Este capítulo de Introducción presenta la propuesta de la tesis y la motivación de la misma. Asimismo se enuncian las hipótesis que son estudiadas y demostradas a lo largo de esta tesis. Para probar la veracidad de dichas hipótesis se plantean los objetivos a cumplir. Finalmente, se presenta la estructura de la tesis para los siguientes capítulos.

Índice del capítulo

[266] Thomas J. Watson
[213] Popular Mechanics
[144] Ken Olsen

En los años 50 se pensaba que el mundo del futuro nunca estaría preparado para más de cuatro o cinco ordenadores [266] y que estos pesarían varias toneladas [213]. En la década de los 70, no se encontraba ninguna razón ni económica ni social para que los ciudadanos tuviesen un ordenador en su casa [144]. Sin embargo actualmente, más de medio siglo después y rompiendo todas las predicciones, cada vez más y más gente en todo el mundo lleva diariamente un pequeño ordenador consigo. Ya sea en su mochila, en su bolsillo, en su muñeca, vistiéndolo o incluso "conduciendo" uno. En la práctica, constantemente, todo el mundo dispone de un computador cerca.

Estos pequeños ordenadores portables, son llamados Smartphones, Smartwatches o Smartcars y se han incorporado en la vida rutinaria de cada vez más gente. Estos dispositivos son denominados Smart o inteligentes no tanto por su capacidad de procesar grandes cantidades de datos o por puras cuestiones de Marketing, si no por su capacidad para poder comunicarse y compartir información con dispositivos de igual o distinta naturaleza. La intercomunicabilidad que ofrecen, frente a otras características, es la que les confiere el valor de dispositivo inteligente o Smart.

Se vive un escenario que incluso para las mentes más brillantes de hace tan sólo medio siglo hubiese resultado utópico o más propio de la ciencia ficción que de una sociedad real: una sociedad intercomunicada en todo momento gracias a sus dispositivos inteligentes. Sin embargo, la mayoría de estas personas se mueven por ciudades ordinarias, muchas de las cuales han cambiado poco o nada en este último medio siglo. La necesidad de elevar dicha inteligencia a las propias ciudades, y por tanto el surgimiento del concepto de Smartcity, era cuestión de tiempo.

Aunque actualmente la definición de Smartcity es aún difusa, y será ampliamente presentada en el transcurso de esta tesis, la inmensa mayoría de los autores coincide en señalar que la finalidad de una Smartcity es la de mejorar la calidad de vida de sus habitantes, haciendo uso para ello de la aplicación de las nuevas tecnologías o TICs. Esos autores coinciden en que para la concepción de una Smartcity, resulta imprescindible tener información real y fiable del funcionamiento de la propia ciudad. Eso, en gran parte, implica comprender como los habitantes de dicha ciudad se mueven y desplazan en ella, o incluso, entre ellas.

1.1 PROPUESTA DE TESIS

Esta tesis se sitúa en la convergencia entre los `Smartdevices` y las `Smartcities`, proponiendo que con la integración de ambos conceptos es factible el estudio y análisis de la movilidad de masas de personas y vehículos de forma anónima, transparente, no intrusiva y de bajo coste.

Para ello se sustenta en la proliferación de los dispositivos inteligentes, proponiendo monitorizar a las personas y vehículos por medio de dichos dispositivos. Se propone hacer uso de la lectura o captación de las señales de comunicación que legalmente pueden ser monitorizadas y que de forma periódica, automática e inadvertida emiten los dispositivos inteligentes para establecer comunicaciones con otros dispositivos cercanos. Más concretamente, empleando las comunicaciones Bluetooth [255] y WiFi [253] que se han convertido en las dos tecnologías de comunicación inalámbrica más extendidas dentro de los dispositivos inteligentes, hasta el punto de ser consideradas estándares de facto en las comunicaciones inalámbricas de rango personal.

[255] 802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems

[253] 802.11-1999 - Standard for Information Technology

Por medio de la captación de dichas comunicaciones, se propone *geoposicionar* y enmarcar temporalmente los reconocimientos de todos los dispositivos relevantes en las inmediaciones. Esta monitorización, refleja la detección de un dispositivo concreto en un punto geográfico determinado, durante un periodo de tiempo finito y acotado.

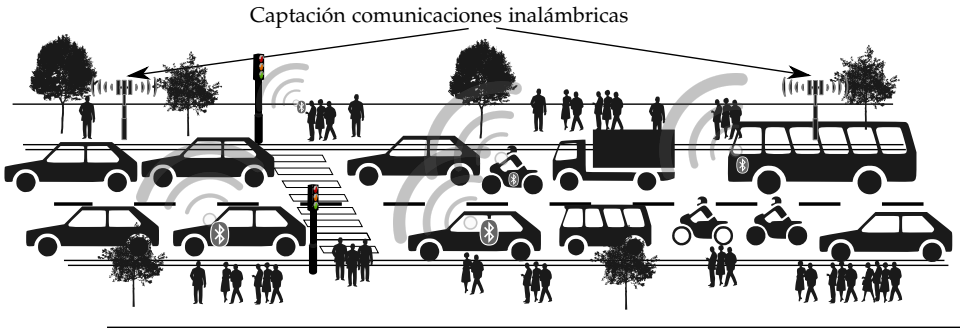


Figura 1.1
Ejemplo de la monitorización por captación de comunicaciones inalámbricas propuesta en esta tesis, aplicada al estudio de la movilidad de personas y vehículos en una Smartcity.

Haciendo uso de este mecanismo de monitorización, ejemplificado en la Figura 1.1, se pretende dotar a las ciudades y organismos que las gestionan de información sobre los movimientos de las personas y vehículos en dicha ciudad. Esta información ha de ser confiable, veraz y en tiempo cercano al real para resultar relevante en el entorno de una Smartcity.

1.2 MOTIVACIÓN

La realización de estudios de movilidad, así como la extracción de información relevante a partir de los datos de movilidad que ofrecen las tecnologías actuales, resulta complejo y en la mayoría de las ocasiones existen limitaciones impuestas por la propia tecnología. Esto es debido a que la mayoría de las tecnologías empleadas actualmente para monitorizar el flujo de tráfico de personas y vehículos, tanto en carreteras, calles o edificios resultan excesivamente caras para resultar viable una implantación en masa. Además, muchas de estas tecnologías sólo son capaces de recoger información sobre la magnitud de movimientos absolutos, siendo incapaces de volver a identificar a la misma persona o vehículo en visitas recurrentes, ya sea a corto, medio o largo plazo. Por esta limitación, no resultan viables para identificar y estudiar las rutas o caminos que el flujo (de personas y vehículos) está siguiendo y por tanto, no proporcionan información necesaria hoy en día en este ámbito.

Contar con fuentes de información sobre el estado del tráfico y del tránsito de personas se antoja clave en el contexto actual, donde el concepto de `Smartcities` está emergiendo. Una fuente de datos capaz de proveer información histórica confiable y veraz sobre cuánto, cuándo y cómo se mueven las personas y vehículos de una ciudad; se convertiría en una herramienta indispensable a la hora de elaborar políticas urbanísticas y de ordenación territorial, así como para la coordinación de las áreas de movilidad de las ciudades. De esta forma, estos organismos podrían conocer de manera inmediata el impacto de las decisiones y variaciones realizadas sobre las vías de comunicación de la ciudad. E incluso, sería abordable, poder delegar la toma de estas decisiones a un agente externo que se nutriese de la información generada por esta fuente de datos.

Esto abre la puerta a la aplicación de técnicas de softcomputing sobre la información histórica y generada en tiempo real por la fuente de datos. Supuesto el éxito en la aplicación de dichas técnicas, se habilitaría una vía para poder gestionar de manera óptima las redes de desplazamiento de las ciudades, tarea vital para la gestión eficiente de una `Smarticity`. Estas técnicas pueden consistir en algoritmos de predicción para estimar magnitudes y direcciones de los flujos de tráfico. O pueden basarse en heúristicas para el cálculo instantáneo de velocidades, tiempos de viaje, detección de atascos u anomalías o elección de mejores rutas. También pueden detectar patrones de comportamiento que ayuden a la comprensión de los hábitos en los desplazamientos. Este tipo de problemas componen un campo de investigación que ha sido y está siendo muy estudiado en el marco teórico. Sin embargo, para su aplicación en entornos reales, requieren de la obtención de información del tráfico y del tránsito de forma cercana al tiempo real. Esto motiva la necesidad de obtener información sobre el movimiento de personas y vehículos mediante el empleo de nuevas técnicas y tecnologías, como la propuesta en esta tesis.

De igual manera la aplicación de esta fuente de datos, y los estudios de monitorización y movilidad nutridos por ellos, pueden ser aplicados a otros escenarios no únicamente a la gestión de una `Smartcity`. Por ejemplo, históricamente el comercio y el marketing han estado interesados en los hábitos generales de las masas de personas, con el único fin de incrementar las ventas. Ser capaz de obtener, por ejemplo, la duración de las visitas en un lugar de ocio, supone disponer de una información muy valiosa desde el punto de vista del marketing. Además, aplicando los mismos principios de monitorización, movilidad y técnicas de Softcomputing, se puede estudiar a los clientes o visitantes basándose en sus desplazamientos, y obtener así conocimiento sobre ellos en base a esta información.

Se concreta, por tanto, que la motivación de esta tesis doctoral es la de estudiar si mediante la captación de las comunicaciones inalámbricas emitidas por los dispositivos inteligentes y el análisis posterior de los datos obtenidosse puede proveer de una fuente de datos viable y confiable para la monitorización de personas y vehículos. De esta forma se espera ofrecer a las administraciones y empresas información de relevancia sobre la movilidad de masas y conocimiento derivado a partir de la aplicación de técnicas de softcomputing. Proveer de esta información y conocimiento supondría un valor añadido para estos organismos, ya que les acercaría al concepto de Smartcity, pudiendo hacer un mejor uso de los recursos a administrar basándose en el conocimiento adquirido.

1.3 HIPÓTESIS

Si bien en las secciones anteriores se han introducido muchos de los aspectos relevantes a tratar en esta tesis, en esta sección se introducen las suposiciones o marcos de estudio a tratar en la misma.

Esta tesis se sostiene sobre tres hipótesis fundamentales, cada una de las cuales compone un escenario de trabajo por si mismo. Sin embargo, estas hipótesis no son independientes, sino que esta tesis construye sobre las bases sentadas en cada una de dichas hipótesis, tal y como se recoge en la figura 1.2. De esta forma, se hace necesario el estudio e investigación de cada una de dichas hipótesis, pues sus resultados sustentan los principios del planteamiento de las siguientes hipótesis.

Hipótesis I: Sobre la captación, reconocimiento y monitorización de las comunicaciones inalámbricas de los dispositivos inteligentes.

La primera hipótesis planteada en esta tesis es que no existen impedimentos (ni tecnológicos ni legales) para realizar una monitorización basada en la captación de las comunicaciones inalámbricas de los dispositivos inteligentes.

Para ser factible esta monitorización, los dispositivos inalámbricos han de realizar las comunicaciones a ser capturadas de forma periódica e inadvertida

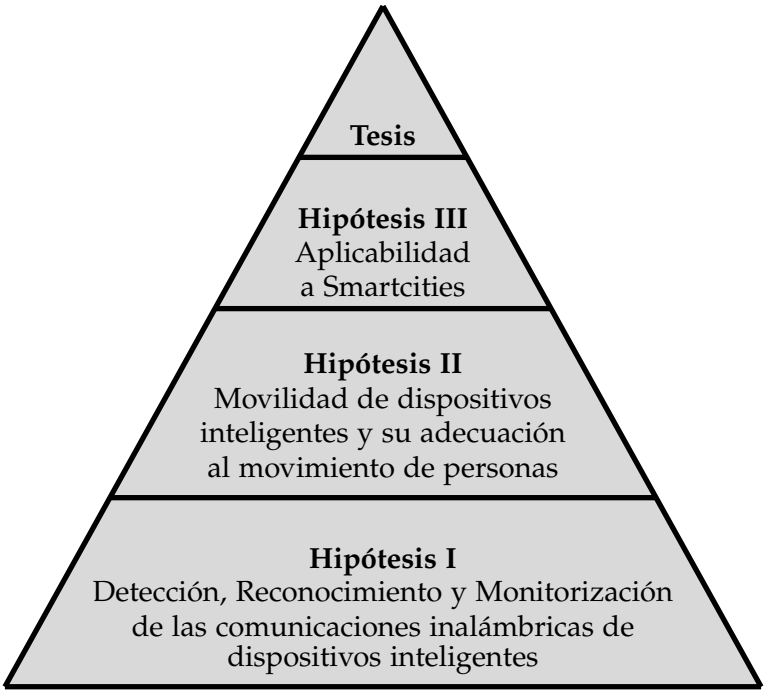


Figura 1.2
Hipótesis planteadas en esta tesis y sus dependencias entre ellas.

para los usuarios. Dichas comunicaciones deben de poder ser capturadas sin vulnerar la legislación vigente y con los medios tecnológicos actuales. Además esas comunicaciones capturadas deben de facilitar mecanismos para permitir distinguir de forma unívoca a un dispositivo emisor determinado, con el fin de poder reconocer que las comunicaciones capturadas posteriormente pertenecen a ese dispositivo concreto.

La captura de estas comunicaciones debe nutrir la monitorización de los dispositivos inteligentes, siendo capaz de enmarcar temporalmente la estancia del dispositivo inteligente en las inmediaciones. Esto implica ser capaz de acotar el espacio de búsqueda a un área concreta, y de discernir cuándo un dispositivo llega y cuando se va.

Hipótesis II: Sobre el estudio de la movilidad de los dispositivos inteligentes y la adecuación al movimiento de personas y vehículos

Supuesta la hipótesis anterior basada en que es posible la monitorización, esta hipótesis plantea que dicha monitorización es capaz de proporcionar información útil y confiable sobre como las personas y vehículos se desplazan por las ciudades.

Esta hipótesis supone, en primer lugar, que es posible estudiar la movilidad de los dispositivos inteligentes en dos o más puntos geógraficos en base a la monitorización individual en cada uno de los puntos. Esta monitorización puntual, según la hipótesis planteada, debe permitir la elaboración de estudios de movilidad cotejando los dispositivos monitorizados por cada uno de

los puntos. Y ofreciendo por tanto información sobre el desplazamiento de un sitio a otro de los dispositivos inteligentes.

En segundo lugar, esta hipótesis plantea que existe una correspondencia directa entre la afluencia de dispositivos inteligentes monitorizados en un sitio (o que se han desplazado de un sitio a otro) y la cantidad de gente o vehículos que hay realmente en dicho sitio. Si bien la monitorización y estudios de movilidad son realizados en base a los dispositivos inteligentes, esta hipótesis plantea que existe una extrapolación o correspondencia que permite inferir el número de personas o vehículos en base a los dispositivos inteligentes detectados. A su vez afirma que los factores externos que afecten al número de personas o vehículos afectará en igual medida y magnitud al número de dispositivos inteligentes detectados.

Hipótesis III: Sobre la aplicabilidad de la información generada al ámbito de a una Smartcity

Esta última hipótesis plantea que la información generada es aplicable al ámbito de una Smartcity. Esta hipótesis supone por tanto que los estudios de la movilidad de personas y vehículos generan información útil que puede ser aprovechada por los organismos e instituciones encargados de la gestión de las vías de comunicación de las ciudades.

Debido a que supone como válida la aplicación de dicha información al contexto de una Smartcity, supone también que dicha información puede generar estructuras de datos que permitan el estudio analítico. Y a su vez, que dichas estructuras y análisis habilitan la aplicación de técnicas de SoftComputing que permiten extraer conocimiento de dicha información.

1.4 OBJETIVOS

El principal objetivo de esta tesis es la de estudiar si una fuente de datos que provea de información sobre monitorización basada en la captura de comunicaciones inalámbricas de dispositivos inteligentes, puede ser empleada para el estudio de la movilidad tanto de personas como de vehículos y servir de herramienta para que las emergentes Smartcities obtengan, en tiempo cercano al real y con un bajo coste, información y conocimiento sobre los flujos de personas y vehículos que se desplazan por ella.

Para demostrar la validez de esta fuente de datos, en la Sección 1.3 de este capítulo se han elaborado las tres hipótesis que se presentan en esta tesis; hipótesis que han de ser demostradas. Los tres primeros objetivos obedecen por tanto a la necesidad de validar mediante el estudio, investigación y experimentación cada una de dichas hipótesis planteadas.

Para cumplir dichos objetivos, y satisfacer por consiguiente las hipótesis, se hace necesario disponer de una herramienta con la que poder experimentar y que haga uso de la fuente de datos de monitorización que se propone. Es por ello, que el cuarto objetivo implica la elaboración de un prototipo que haga uso de la tecnología y metodología que han de ser estudiadas y validadas.

El prototipo y su elaboración es un objetivo derivado de la necesidad de poder demostrar la viabilidad de la fuente de datos y demostrar las hipótesis que han sido planteadas en esta tesis. Por ello debe de entenderse como una herramienta desarrollada para habilitar los estudios experimentales e investigación aplicada para las hipótesis planteadas en esta tesis.

Objetivo I: Captación, reconocimiento y monitorización de las comunicaciones inalámbricas de los dispositivos inteligentes

El primer objetivo de esta tesis, relativo a la primera hipótesis, consiste estudiar y validar los fundamentos teóricos y tecnológicos que sustentan la fuente de datos que se propone para la monitorización: la captación de las comunicaciones BT y WiFi de los dispositivos inteligentes.

Este primer objetivo implica la realización de los estudios de los mecanismos de captación de comunicaciones inalámbricos amparados por las normativas vigentes. Se abordan también los principios que habilitan el reconocimiento del mismo dispositivo en distintas captaciones tanto a corto como a largo plazo.

Con el cumplimiento de este objetivo, se habilita la posibilidad de aplicar una monitorización veráz a los dispositivos inteligentes. Dicha monitorización debe consistir en enmarcar temporalmente la estancia o paso de un dispositivo determinado por punto geográfico acotado.

Este objetivo se aborda de forma teórica en el Capítulo 4, se materializa en el desarrollo en el Capítulo 5 y es sometido a estudio experimental en la Sección 6.1.

Objetivo II: Movilidad de dispositivos y adecuación al movimiento de personas

El segundo objetivo de esta tesis, relativo a la segunda hipótesis, consiste en estudiar si la monitorización de dispositivos inteligentes es útil para el estudio de la movilidad. Estos estudios de movilidad implican tanto el ser capaz de cuantificar los desplazamientos de un punto geográfico a otro como la adquisición de información añadida sobre dicho desplazamiento. Como por ejemplo el tiempo necesario para realizarlo o la variación de dicho tiempo respecto al tiempo necesario habitualmente.

Este objetivo abarca también los estudios relativos a la aplicabilidad y extrapolación de los resultados de movilidad de dispositivos al movimiento de personas y vehículos. Para tal fin, se hace necesario comparar los resultados de dichos estudios de movilidad con fuentes externas de datos, consistentes en otros estudios de movilidad realizados y cedidos por organismos e instituciones públicos, haciendo uso de otras técnicas de monitorización.

Se propone, por tanto, estudiar la correlación existente entre ambas fuentes de datos, así como la causalidad frente a factores externos, como por ejemplo la hora del día, el día de la semana o las condiciones meteorológicas. Estos factores externos afectan al tránsito y tráfico de personas y vehículos, y por tanto deben afectar en igual medida a ambas fuentes de datos, por lo que es esperable que ambas reflejen una fluctuación similar ante el mismo factor externo.

En el Capítulo 3 se presentan las métricas empleadas habitualmente en los estudios de movilidad. La Secciones 5.1 y 5.11 detallan como se aplican dichas métricas al sistema de monitorización propuesto. Finalmente, en la Sección 6.2 se presentan estudios de movilidad llevados a cabo por medio del sistema de monitorización propuesto.

Objetivo III: Aplicación del sistema de monitorización al ámbito de las Smartcities

Supuestos cumplidos los anteriores objetivos, es decir, supuesto que la monitorización de dispositivos de comunicaciones inalámbricas permite el estudio de la movilidad de las personas y vehículos, queda evaluar si los resultados de dichos estudios proporcionan un entorno de información útil para las Smartcities. O lo que es lo mismo, estudiar la veracidad de la tercera hipótesis.

Validar dicha hipótesis implica mostrar que los estudios de movilidad que pueden realizarse en base a la monitorización realizada proveen de estructuras de datos que permiten obtener información veraz sobre el es-

tado del tránsito y tráfico. Para tal fin se generarán estructuras de datos muy empleadas habitualmente en los problemas de movilidad, como son las series temporales, las matrices entrada-salida o los grafos dirigidos de desplazamiento.

Dichas estructuras serán analizadas y resumidas con el fin de obtener información que resulte útil para la gestión del tráfico. Como por ejemplo, conocer las predilecciones del tráfico en un cruce y cómo influyen factores externos en dicha elección. También conocer cuál es el tiempo medio para realizar un desplazamiento entre varias rutas u obtener información sobre las velocidades de desplazamiento. Además, se estudiará la influencia de factores externos en dichas magnitudes. Por ejemplo, la influencia de un fin de semana o festivo en el compartamiento del tráfico.

Además se estudiará si las estructuras de datos generadas y la información obtenida son aptas para la aplicación de técnicas y heurísticas de SoftComputing, con el fin de obtener una capa de conocimiento. Por ejemplo, se aplicarán algoritmos de predicción a las series temporales con el fin de estimar el volumen del tráfico o el tiempo necesario para un desplazamiento futuro. Se realizarán estudios para reconocimiento de anomalías en tráfico, asi como técnicas de reconocimiento de patrones para clasificar a las distintas personas o vehículos en base a sus movimientos.

En el Capítulo 2 se presentan las necesidades a las que tienen que hacer frente las ciudades inteligentes del furuto. La Sección 5.12 y siguientes presentan las técnicas de Softcomputing empleadas. La Sección 6.3 presenta la aplicación de las técnicas presentadas para la extracción de conocimiento a los datos obtenidos por el sistema de monitorización propuesto.

Objetivo IV: Prototipo funcional del Sistema

Por último, tal y como se ha indicado anteriormente, para satisfacer los objetivos propuestos en esta tesis se hace necesario disponer de un prototipo funcional que haga uso de monitorización mediante captación de comunicaciones inalámbricas de dispositivos inteligentes.

Este prototipo será pues una herramienta resultado de la investigación realizada, con el fin de disponer de un entorno sobre el que realizar los estudios, investigaciones y experimentación presentados en esta tesis.

El desarrollo del prototipo se encuentra detallado en el capítulo 5, donde se recogen todos los aspectos relacionados con el prototipo de sistema desarrollado.

.

1.5 ESTRUCTURA DE LA TESIS

Una vez presentada la propuesta de tesis, su motivación, las hipótesis que plantea y los objetivos a cumplir en la misma, se presenta la estructura del resto de este documento.

El Capítulo 2 recoge los antecedentes en los que se sustenta esta tesis. Presenta un breve contexto de los dispositivos inteligentes y las ciudades inteligentes.

El Capítulo 3 presenta la revisión bibliográfica. Aborda las técnicas más extendidas en la bibliografía para la monitorización tando de vehículos como de personas y las magnitudes y métricas que son empleadas en los estudios de monitorización

El Capítulo 4 recoge la fundamentación teórica de la captación de las comunicaciones inalámbricas. Presenta en primer lugar una breve introducción a las comunicaciones inalámbricas, para luego dedicarse a cada una de las tecnologías de transmisión inalámbricas empleadas.

El Capítulo 5 es el capítulo más extenso de esta tesis, debido a que recoge todo aspecto relacionado con el prototipo de sistema de monitorización propuesto. En la Sección 1.5.1 se recogen ciertas recomendaciones de lectura orientadas a dicho capítulo.

El Capítulo 6 recoge los experimentos que validan de forma empírica las hipótesis planteadas de esta tesis. Se presentan agrupados en tres secciones, una sección para cada hipótesis planteada.

El Capítulo 7 expone las conclusiones de la tesis, basándose en todo el trabajo presentado anteriormente.

1.5.1 *Recomendaciones de lectura*

Debido a que la propuesta de tesis abarca numerosos campos y aréas, ha resultado en un documento excesivamente extenso. Si bien, no todas las partes del documento son requeridas para adquirir una concepción de la propuesta de tesis o pueden no resultar atractivas para el lector por no estar relacionadas con su campo de investigación o no ser de vital importancia para la propuesta de tesis.

El Capítulo 2 se centra en dos aspectos, los dispositivos inteligentes y las ciudades inteligentes. En los dispositivos inteligentes (Sección 2.1, se muestra su desarrollo desde los fundamentos de la computación ubícua impulsada por el Internet de las Cosas. Las ciudades inteligentes se aborda de forma muy ligera, centrándose en los retos y problemas que tienen que enfrentarse las ciudades europeas respecto a la gestión del tráfico. Si bien su lectura sirve para ofrecer un contexto de los antecedentes en los que se sustenta la tesis, no resultan imprescidibles.

EL Capítulo 3 presenta las métricas empleadas en los sistemas de monitorización comunes en la bibliografía. Los sistemas estudiados son puestos siempre en comparativa con la captación de comunicaciones inalámbricas. Este capítulo obedece a la necesidad de emplazar el sistema de monitorización propuesto frente a los sistemas actualmente en uso.

El Capítulo 4 presenta los fundamentos teóricos que permiten la captación de comunicaciones inalámbricas (en el caso de Bluetooth BR/EDR y WiFi) o bien que la imposibilitan (casos de Bluetooth LE, NFC, RFID, telefonía inalámbrica). Si bien resulta un capítulo denso, se hace necesario abordar este enfoque teórico con el fin de presentar el funcionamiento de ambos protocolos de comunicación inalámbrica explotados por el sistema de monitorización propuesto. Además, en este capítulo se presentan los conceptos de legalidad de captación de cada uno de los protocolos.

El Capítulo 5 aborda demasiados aspectos, muchos de los cuales no tienen que ser de relevancia para todos los lectores. Resulta imprescindible la Sección 5.1 donde se presentan los fundamentos del sistema de monitorización propuesto. La Sección 5.2 de requisitos resulta interesante desde el punto de vista de la ingeniería del software y de los restos que se pretende abordar en el prototipo, como la necesidad de resultar de bajo costo.

La Sección 5.3 descompone el sistema en distintos componentes, que son abordados de forma individual en las siguientes secciones del capítulo:

La Sección 5.4, relativa al hardware del sistema, resulta interesante a bajo nivel porque presenta la elección de cada uno de los componentes, así como costo y consumo energético.

La Sección 5.5, relativa al sistema operativo, resulta interesante desde el punto de vista de la arquitectura de computadores pues se presenta las configuraciones y resoluciones para lidiar con la computación en un

dispositivo empotrado, intentando garantizar la robusted e integridad del sistema.

La Sección 5.6, relativa al software de monitorización, abarca los numeros módulos desarrollados para la monitorización que constituyen el sofware `RAZIEL`. Resulta muy relevante la descripción de la arquitectura propuesta. El interés de cada uno de los componentes descritos, varia en función del interés del lector. Adicionalmente

La Sección 5.7, relativa al servidor de cómputo que centraliza la captura de información resulta relevante desde el punto de vista de la ingeniería de servidores. Se presentan brevemente las configuraciones para lograr una alta eficiencia, así como los distintos servicios desarrollados que ofrece el servidor de cómputo.

La Sección 5.8, relativa a la `API` de comunicación entre los nodos y el servidor, se centra principalmente en la securización del canal de comunicaciones entre estos.

La Sección 5.9, relativa al almacenamiento de datos, aborda la descripción de los métodos de alto rendimiento empleados para facilitar la explotación de los datos en producción. Resultará de mayor interés para aquellos interesados en los aspectos más relacionados con el *Big Data* y la gestión eficiente de los datos.

La Sección 5.10, relativa al procesamiento eficiente de datos, resulta de interés para aquellos interesados en la algorítmica y la optimización de procedimientos.

La Sección 5.11, relativa al análisis, presenta los fundamentos de las series temporales, las matrices de entrada y salida y los conjuntos de datos clasificables, desde un enfoque teórico.

La Sección 5.12, relativa al aprendizaje automático, presenta los métodos empleados para la clasificación, predicción y detección de anomalías. Su interés se acerca más a la inteligencia artificial.

La Sección 5.13, relativa al almacenamiento en la nube, resulta de interés para el campo del `cloud computing` y el almacenamiento en bases de datos no relacionanles.

Finalmente, la Sección 5.14, relativa a la difusión y publicación de resultados, presenta brevemente algunos de los desarrollos web, móvil, documental y por redes sociales para la publicación de la información generada por el sistema de monitorización. Sirve de galería de las distintas aplicaciones alternativas desarrolladas para tal fin.

El Capítulo 6 recoge los experimentos relativos a la demostración de las hipótesis. Los experimentos que no se enmarcan directamente en alguna de las hipótesis (p.e. los relativos a la eficiencia de métodos o del sistema) se han emplazado en las secciones correspondientes. Se ha intentando, en la medida de lo posible, incluír unicamente los experimentos más relevantes para la demostración de las hipótesis. Muchos de los experimentos presentados se

han realizado con varios conjuntos de datos de distintos nodos a lo largo del tiempo, pero en la medida de lo posible se ha escogido aquel conjunto de datos que se realizase en mayor profundidad, con el fin de no presentar las mismas técnicas variando únicamente el conjunto de datos.

Los experimentos relativos a la primera hipótesis (Sección 6.1) pertenecen en su mayoría a documentos internos y pruebas de control realizados durante la investigación que no han sido publicados con anterioridad. Resultan relevantes para determinar el comportamiento de los dispositivos inteligentes, respecto a la captación de las comunicaciones inalámrbicas.

Los experimentos relativos a la segunda hipótesis (Sección 6.2) han sido publicados en su mayoría en los informes y entregables de los proyectos en los que el sistema de monitorización propuesto ha estado involucrado. Resultan relevantes para la monitorización, tanto de personas como de vehículos, y sus estudios analíticos y descriptivos.

Por último, los experimentos relativos a la tercera hipótesis (Sección 6.3) han sido publicados en revistas con índice de impacto así como en aportaciones a congresos. Resultan relevantes en el campo de las ciudades inteligentes y en la apliación de técnicas de Softcomputing para la clasificación, predicción y detección de anomalías.

El capítulo 7 de conclusiones presenta de forma breve y concisa los resultados de esta tesis doctoral, validando las hipótesis planteadas y concluyendo sobre la viabilidad de la captación de las comunicaciones inalámbricas.

2

ANTECEDENTES

Esta ciudad no se levantó con hormigón y acero.
¡Se levantó con ideas!

— Andrew Ryan (Bioshock)

En este capítulo se abordan las definiciones de los conceptos mencionados durante el capítulo de Introducción, así como el contexto histórico en el que que se sientan las bases de la captación de comunicaciones inalámbricas para la monitorización de personas y vehículos, como se propone en esta tesis.

En primer lugar se presentan los dispositivos inteligentes o `smartdevices`, maxima representación actual del concepto de Computación Ubicua, cuyo surgimiento y despegue ha sido influenciado por la corriente del Internet de las Cosas. Se presentará como dichos dispositivos han sido popularizados hasta el punto de que la mayor parte de los habitantes posee al menos uno.

A continuación se describe la problemática a la que se encuentran las ciudades debido al aumento de población que están sufriendo y al impacto medioambiental que este aumento de población genera. Si bien se presentarán las numerosas problemáticas a las que se enfrentan las ciudades del futuro, para este trabajo resultan más relevantes las asociadas con la movilidad y el transporte, centradas en el marco europeo. Se presentarán por tanto los índices de crecimiento que están experimentando las ciudades en Europa, así como un contexto histórico de las mismas. Concienciar sobre este transfondo resulta crítico como medio para entender que la efectividad y eficiencia de la gestión de las ciudades europeas puede ser dificilmente mejorable (o abordable) por medio del cambio de sus infraestructuras físicas debido a los orígenes antiguos de la mayoría de las ciudades. Se debe optar por tanto una mejora de la gestión y de la administración de los recursos, tornándose las Tecnologías de la Información y Comunicación claves en este rol, dando lugar a las Ciudades Inteligentes o `Smartcities`. Ciudades que cuentan con redes de sensores desplegadas obteniendo información que sirve de base de conocimiento para la mejora de su propia eficiencia.

Índice del capítulo

2.1 DE LA COMPUTACIÓN UBÍCUA E INTERNET DE LAS COSAS: EL SURGIMIENTO DE LOS SMARTDEVICES.

En el año 1991 los investigadores de la universidad de Cambrige compartían una única cafetera para varias plantas de uno de los principales edificios. Los investigadores se frustraban a menudo por el hecho de tener que subir multitud de escalones y encontrarse que la cafetera estaba vacía.

Un equipo interdisciplinar de expertos en distintos campos deseosos de café recién hecho, instalaron una videocámara rudimentaria que emitía por la red interna del edificio unas tres imágenes por minuto del estado de la cafetera, suficiente para determinar el nivel de café remanente en la jarra [108].

[108] The Internet of things: Networked objects and smart devices

Esta cafetera es considerada hoy día la primera manifestación de un objeto cotidiano dotado de cierta inteligencia, conectado a un red para compartir su información. Con esta cafetera, empezó la edad dorada de los dispositivos inteligentes.

2.1.1 *Los orígenes: Computación ubícua*

La computación ubícua es entendida como la integración de los aspectos relacionados con la informática al entorno habitual de las personas, de forma que los computadores no se perciban como objetos diferenciados de los objetos cotidianos [292].

[292] The Computer for the 21st Century

Mark Weiser se vio influenciado por la novela Ubik[1] en que se presentaba un futuro en el que todo objeto manufacturado (desde las puertas al papel higiénico) era inteligente e interconectado. En el año 1988 influenciado por esta distopía empieza a concebir el concepto de Computación Ubícua, formalizándola en trabajos posteriores [289-293]. En estos trabajos, propone tres modelos básicos que debían ser considerados para desarrollar sistemas ubícuos.

[289-293] The computer for the 21st century, Some computer science issues in ubiquitous computing, Ubiquitous computing, The Computer for the 21st Century, The origins of ubiquitous computing research at PARC in the late 1980s

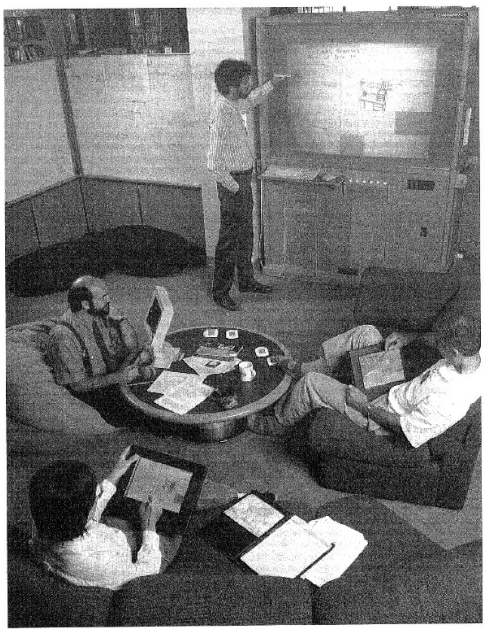
tabs o dispositivos de escasos centímetros que podían ser llevados encima o vestidos por las personas.

pabs o dispositivos del tamaño de la palma de la mano.

boards o dispositivos de grandes dimensiones interactuables por varias personas al únisono.

En la figura 2.1 se puede observar como las personas fotografiadas interactuan con los computadores de forma natural y cotidiana, tanto en el concepto original de computación ubícua publicada en el año 1991 como en la actualidad.

1 ↑Novela de Philip K Dick disponible en http://amzn.to/2jS1OKF



(a) Concepto de 1991

(b) En la actualidad

Figura 2.1
(a) Fotografía que ilustra la concepción de la computación ubícua, donde las personas interactuan con computadores más cercanos a los objetos cotidianos de las personas.
Fuente: The Computer for the 21st Century [292] - Figura 1.
(b) Fotograma de la serie de televisión Modern Family en la que refleja como las familias interacturan con computadores de forma cotidiana.
Fuente: Modern Family. Derechos de ABC y 20th Century Fox.

La proliferación de los dispositivos ubícuos ha superado toda expectativa. Smartphones, tablets, ordenadores portátiles, videoconsolas, frigoríficos y televisores inteligentes,... son dispositivos que se han implantado en la sociedad hasta un punto que hubiese resultado una utopía para Mark Weiser[2].

Sin embargo, se han alejado del concepto de la computación ubicua original, pues son dispositivos que se han centrado más en las capacidades de comunicación que en el acercamiento de la computación a la persona de a pie. Esta cualidad, la de interconexión, se ha visto impulsada por el surgimiento del Internet de las Cosas o IoT, que se presentará a continuación.

2 ↑Mark Weiser falleció en el año 1999

2.1.2 *IoT: Internet de las cosas.*

El concepto de Internet de las Cosas (Internet of Things o IoT) es un paradigma emergente en el año 1999 en el escenario de las comunicaciones inalámbricas modernas [20]. La idea básica del paradigma es la existencia de una enorme variedad de `Cosas o Things` con capacidad para comunicarse, de forma que estas `Cosas o Things` pueden interaccionar y colaborar las unas con las otras. El concepto de `Cosa o Thing` se centra en objetos muy simples y cotidianos como etiquetas, sensores, actuadores, teléfonos móviles, electrodomésticos, vehículos... [99]. Por lo general, dispositivos que acceden a la red sin mediación de una orden directa de una persona, es decir, de forma autónoma o no interactiva.

[20] The Internet of Things: A survey

[99] The internet of things: 20th Tyrrhenian workshop on digital communications

Las primeras definiciones del concepto de IoT son atribuidas a Kevin Ashton de Auto-ID [15], laboratorio del MIT centrado en la investigación de las redes RFID [228] y las tecnologías de sensores emergentes.

[15] **title**

[228] Radio frequency identification (RFID)

El potencial en la poca precisión del concepto de `cosa o thing` abre las puertas al desarrollo de una gran cantidad de aplicaciones en infinidad de escenarios distintos con cualquier `thing` imaginable. Ya sea en nuestro hogar, nuestro lugar de trabajo, nuestra cafetería o pub favorito, o incluso en los desplazamientos que realizamos de un sitio a otro, multitud de pequeños objetos están a nuestro alrededor, susceptibles de ser dotados de inteligencia y capacidad de comunicación. Al otorgarle a estos objetos la capacidad de comunicarse, se genera una cantidad de información latente, a la espera de ser explotada mediante diversas aplicaciones.

En el campo del IoT se distinguen cuatro áreas de aplicación del paradigma [20][3]:

[20] The Internet of Things: A survey

- Área del transporte y la logística.
- Área de la salud.
- Área de los entornos inteligentes.
- Área personal y social.

Sin embargo ha sido en el dominio personal y social donde el IoT ha influido en el nacimiento comercial de los denominados Dispositivos Inteligentes o SmartDevices.

3 ↑Estas áreas serán rescatadas cuando se aborde el tema del desarrollo del prototipo de sistema de monitorización usando la fuente de datos propuesto, pues este este prototipo será a su vez un dispositivo del área del transporte y la logística del IoT.

2.1.3 *Dispositivos inteligentes*

Un Dispositivo Inteligente o Smartdevice es un dispositivo electrónico con posibilidad de interconexión a otros dispositivos o redes de dispositivos que puede funcionar tanto de forma interactiva como de forma autónoma [96, 175, 214].

[96, 175, 214] Microsensors, MEMS and smart devices, From smart devices to smart everyday objects, Ubiquitous computing: smart devices, environments and interactions

Un dispositivo inteligente dispone de un sistema hardware y de recursos software usualmente estáticos, decididos en el diseño del mismo. Esto no cierra la posibilidad de conexión con dispositivos hardware y plugins de naturaleza Plug and Play [205], o con soluciones Sotfware integradas a posteriori.

[205] Plug-and-play

Entre los dispositivos inteligentes más populares existentes actualmente se encuentran los smartphones, los phablets, los tablets, los smartwatches, smartbans, smart tvs y demás electrodomésticos inteligentes.

Toda esta explosión de dispositivos smart se debe gracias al origen comercial de los dispositivos Smartphones, que fueron los dispositivos que iniciaron la popularización del término inteligente o `smart` asociado a un dispositivo electrónico.

2.1.3.1 *Smartphones*

Originariamente el `smartphone` fue concebido como la unión del teléfono móvil y de la PDA o `Personal Digital Assistant`. Comercialmente, el primer dispositivo en hacer uso del término `smartphone` fue el `Ericcson GS88` en el año 1992, aunque por las características se suele denominar al `IBM Simon` del mismo año como el primer `smartphone`, aunque no usaba dicho término en su publicidad.

Sin embargo, no sería hasta el año 2000 con el lanzamiento del sistema operativo `Windows Pocket PC` cuando estos dispositivos se empezarían a popularizar en el entorno empresarial. Entre el año 2002 y 2004 la compañía HTC lanzó en Europa los terminales Wallaby, Falcon e Himalaya (Figura 2.2). Durante los siguientes años los smartphones se siguieron popularizando en el entorno empresarial de la mano de `RIM`[4] con su línea `Blackberry` con `Blackberry OS` y `Palm Inc.` con numerosos smartphones y PDAs haciendo uso de `Palm OS`.

Sin embargo, `Windows Pocket PC` era un sistema operativo excesivamente similar en el uso a los habituales PC de sobremesa con Windows, resultando poco funcional debido a una excesiva dependencia al uso de un puntero o stylus en la navegación por la interfaz.

`Blackberry OS` estaba excesivamente centrado en el entorno empresarial y dependía para las comunicaciones de un servidor `BES`[5] desplegado en

4 ↑Research In Motion
5 ↑Blackberry Enterprise Server

(a) HTC Wallaby (b) HTC Falcon (c) HTC Himalaya

Figura 2.2
Primeros Smartphones presentados por HTC usando Windows Pocket PC.

la infraestructura propia de cada empresa, lo que impedía acceder a sus servicios de comunicación a la población no empresarial.

`Palm OS` por su parte, cayó en el olvido debido a maniobras empresariales que no agradaron a la comunidad originadas por la compra de la empresa desarrolladora del código fuente `PalmSource` por la japonesa `ACCESS` y el cambio de nombre del sistema operativo a `Garnet OS`. Todas estas fluctuaciones e inseguridades, hicieron que los dispositivos Palm no afianzasen la confianza de los consumidores.

Pero el mercado de los smartphones se revolucionó en el año 2007 cuando dos de las empresas más importantes del sector de las TICs anunciarón por separado los dos principales sistemas operativos que compiten por el mercado aún hoy en día. Por un lado, la empresa Apple presentó el `iPhone` haciendo uso del sistema operativo `iOS`. Por el otro `Google` en colaboración con `HTC` presentaron el `HTC Dream` usando el Sistema Operativo `Android` (Figura 2.3). Estos dos sistemas operativos fueron desarrollados por y para smartphones y orientados hacia la población general más que para un uso empresarial. Además estaban diseñados para llegar a las masas, haciendo uso de interfaces simples, diseñadas para ser manejadas mediante el empleo de los dedos para tocar en los elementos de la propia interfaz a través de una pantalla táctil capacitiva.

(a) iPhone (b) HTC Dream

Figura 2.3
iPhone y HTC Dream, primeros dispositivos en hacer uso de los sistemas operativos iOS y Android respectivamente, que popularizaron los smartphones a las masas.

Con el lanzamiento de estos dos terminales, comenzó la invasión de los smartphones en el mundo. Según un estudio realizado por Pew Research Center[215] se estima que el 43 % de las personas en edad adulta del mundo posee un `smartphone`. España es uno de los países del mundo donde más implantados están estos dispositivos (Figura 2.4) siendo el sexto país del mundo y el máximo exponente de la Unión Europea.

[215] Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies

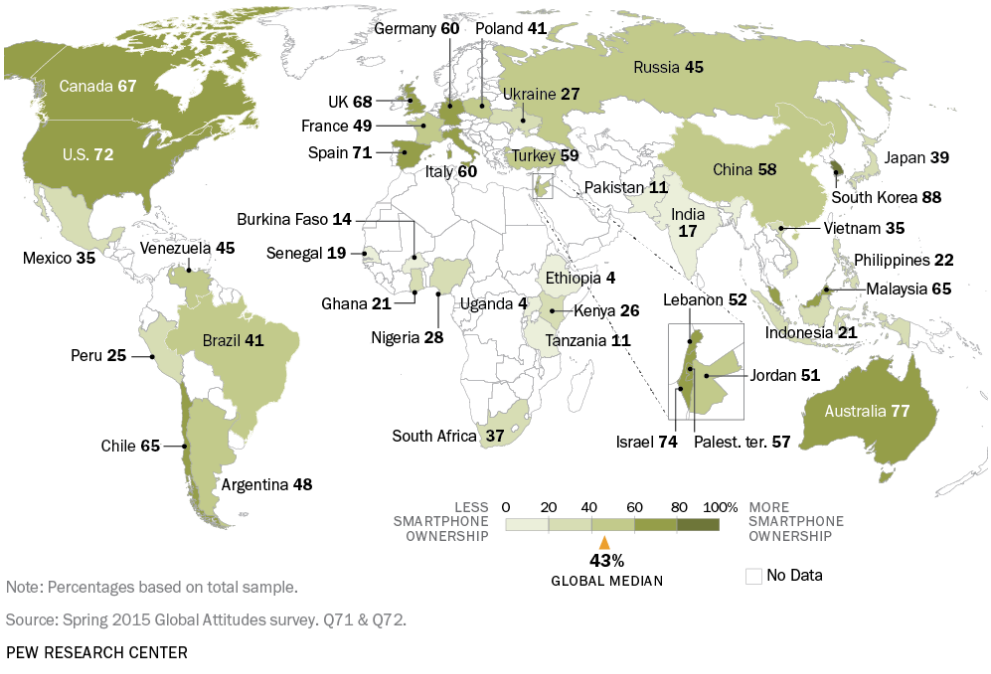
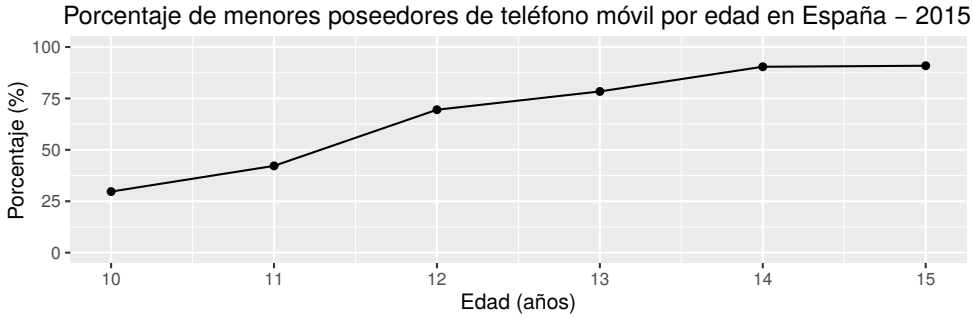


Figura 2.4
Implantación de los smartphones en todo el mundo según el estudio de Pew Research Center[215]

Según datos del Institudo Nacional de Estadística Español [79] el 94.7 % de los hogares dispone de al menos un `smartphone`. El 83 % de las personas

[79] Nota de Prensa sobre Equipamiento y Uso de Tecnologías de Información y Comunición en los Hogares

entre 16 y 74 años accede habitualmente a Internet mediante su `smartphone` fuera de su vivienda habitual o lugar de trabajo. Además, la telefonía móvil es una tecnología muy adoptada por la población joven según se puede ver en la Figura 2.5, donde se muestra como a partir de los 14 años más del 90 % de los jóvenes españoles posee su propio teléfono móvil.



Fuente: INE 2015.

Figura 2.5
Implantación de los teléfonos móviles en España para los menores de edad.

Según la Fundación Telefónica en su Informe Anual del año 2015 sobre la Sociedad de la Información en España [264] de todos los teléfonos móviles vendidos durante el 2015 dentro de nuestras fronteras, un 87 % son `smartphones` y un 90 % de los usuarios de `smartphones` se conecta diariamente a Internet.

[264] Informe Anual sobre la Sociedad de la Información en España

Los `smartphones` (y demás dispositivos inteligentes) se han arraigado hasta tal punto en la sociedad, que están surgiendo nuevas patologías mentales asociadas a su adicción como la **nomofobia** [149] consistente en el temor a verse despojado del smartphone.

[149] Nomophobia: Dependency on virtual environments or social phobia?

En la sociedad actual es cada vez más frecuente la gente que porta diariamente su `smartphone` así como los demás dispositivos inteligentes que posea. Resultando impensable (e incluso patológico) el hacer cualquier desplazamiento por la calle sin llevar el `smartphone`.

2.2 DE LAS CIUDADES ACTUALES A LAS SMARTCITIES

En esta sección se presenta una breve historia de las ciudades europeas necesaria para entender las limitaciones a las que se encuentran para solventar los problemas emergentes en el Siglo XXI. Se presentaran dichos problemas, centrados en la gestión eficiente y sostenible de los recursos. Finalmente se presentarán las Smarctities como vía de solución a dichos problemas, presentando su definición más aceptada y las características que ofrecen. Siendo la más destacable de ellas la implantación de redes de sensores con los que monitorizar todos los aspectos relevantes de la ciudades.

Parte de los problemas que están experimentando las ciudades actualmente son debidos al aumento de la población en las mismas, viviéndose un nuevo éxodo rural. Sirva de ejemplo señalar que, según los últimos informes de la Unión Europea mediante la `Urban Audit` [274], en la Unión viven más de 510.1 millones de personas, de las cuales más de un 35.2 % se encuentra en 86 ciudades de más de 500.000 habitantes[6]. En España, la cifra aumenta al 48.5 % del total viviendo en grandes ciudades. Si contamos núcleos urbanos no únicamente ciudades, en Europa la cifra asciende al 73 %, y está previsto que para el 2050 se supere el 80 % [244], porcentaje que ya se ha superado en el contiene Americano.

[274] Urban Audit

[244] Smart Cities: La transformación digital de las ciudades

Sin embargo las ciudades americanas se encuentran mejor preparadas para la concetración y conciliación de grandes masas de personas, puesto que el desarrollo histórico ha sido muy diferente al sufrido por las europeas [234].

[234] Las formas de crecimiento urbano

Las ciudades americanas se orientan al aprovechamiento vertical del espacio con distribuciones de las edificaciones basadas en planos regulares de calles perpendiculares, que resultan modélicamente perfectas. El sobredimensionamiento de los elementos urbanos es una señal característica, con grandes avenidas y calles de multiples carriles que sirven de arterias para distribuir el tráfico. La presencia de varios medios de transporte públicos es habitual, habiendo influido estos medios fuertemente en la expansión de las áreas urbanas, al crecer en torno a ellos.

Al contrario, las ciudades europeas tienden a crecer horizontalmente anexándo los núcleos urbanos colindantes. La calles de las ciudades europeas presentan planos irregulares ancestrales y su trazado resulta dificilmente alterable debido a la preservación del patrimonio urbanístico. Las infraestructuras del transporte, al contrario que en las ciudades americanas, ha tenido que adaptarse a ciudades ancestrales y la preservación del patrimonio. Esto ha provocado que la distribución de los edificios y calles, así como el aprovechamiento del espacio, sea radicalmente distintos entre las ciudades europeas y las ciudades americanas (Figura 2.6).

6 ↑Aún así, está previsto que la UE pierda próximamente más de 10 % de su población como consecuencia de la salida del Reino Unido de la Unión.

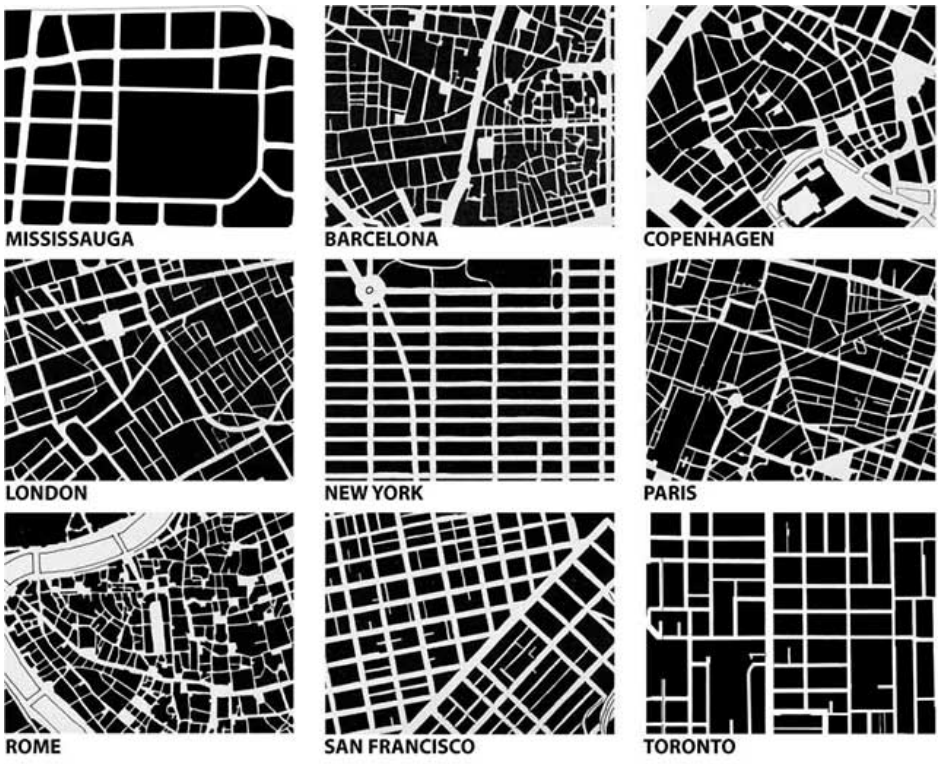


Figura 2.6
Planos comparativos entre diferentes ciudades europeas y americanas, donde se apreciar la diferencia de distribución entre ellas.
Fuente: Las formas de crecimiento urbano [234]

La naturaleza irregular de las ciudades europeas en contraposición con las americanas, hace que la gestión del tráfico que se mueve por ellas resulte en muchas ocasiones mucho más difícil de gestionar de forma eficiente [40].

[40] Ungulate traffic collisions in Europe

2.2.1 *Revisión histórica de las ciudades europeas*

Si bien las primeras ciudades del mundo se originaron hace más de 6000 años a orillas de los grandes ríos[7], el concepto actual de ciudad se debe a la herencia dejada por el impacto de la Antigua Grecia y al Imperio Romano. Sin embargo tanto en Europa como en España existían ciudades fundadas por los pueblo prerrománicos, muchas de las cuales aún perduran: Cádiz fundada en el Siglo XI a.C por los Tirios, Sevilla en el sigo VIII a.C por los Tatessos o Lisboa en el Siglo VII a.C por los Celtas.

Las ciudades romanas presentaban un plano ortogonal, con dos calles principales denominadas `cargo` (en dirección Norte-Sur) y `decumano` (dirección este-oeste). En el cruce de estas dos calles se emplazaban los edificios comunales más importantes. El resto de calles se trazaban paralelas a estas, alojando el resto de edificios y las viviendas, disminuyendo la calidad de las mismas a medida que se le alejaban de la intersección de las dos calles principales. Las ciudades eran abiertas y de calles amplias, pues el propio

7 ↑Nilo, Tigris, Eufrates, Gangers, Indo y Amarillo

poder intimidatorio del Imperio Romano servía para protegerlas frente a posibles invasores. Muchas ciudades europeas fundadas por los romanos conservan en la actualidad su casco antiguo romano.

Si bien tras la caída del Imperio Romano en el 476 (hecho que marca el comienzo de la Edad Media en Europa) las ciudades caen en decadencia, estas volverían a cobrar importancia a partir del Siglo XII. Las ciudades medievales se rodearían por murallas de piedra y eran trazadas mediante un plano irregular, con calles estrechas y sinuosas para dificultar un posible asedio o invasión. La defensa de las mismas era la prioridad en las decisiones urbanísticas, más que el transporte de mercancías o los factores estéticos.

Durante la Edad Moderna, debido al descubrimiento de América, las zonas de guerra y conflicto se alejaron de las principales ciudades europeas, por lo que la preocupación urbánistica se centró en los factores estéticos y en la mejora de la calidad de las ciudades, construyéndose por ejemplo grandes jardines, estanques, monumentos, edificios culturales y religiosos que forman un conglomerado culturan incrustado en el núcleo de las ciudades.

La industralización a finales del siglo XVIII tuvo un gran impacto en las ciudades europeas debido a la migración del campesinado para servir de mano de obra en las nuevas factorias e industrias, lo que provocó el crecimiento de las ciudades, tanto en habitantes como en extensión. Muchas ciudades para facilitar su expansión derriban las murallas heredadas del medievo. De igual manera su trazado fue modificado aunque de forma más planificada que en el medievo, incrementándose las diferencias entre los barrios obreros y los barrios burgueses. El rápido crecimiento acelerado de las ciudades sumado a la industrialización permitió mejorar las infraestructuras de las ciudades europeas, hasta el momento inexistentes o de orígenes romanos. Esta planificación y expansión permitieron abrieron las puertas a la inversión de capital y esfuerzo en las estructuras necesarias para el despliegue de los servicios básicos como agua corriente, electricidad o alcantarillado. Se empiezan a ver también los primeros tranvías y líneas de metro, con el fin agilizar los desplazamientos de la emergente masa de personas. Además, la preservación del patrimonio urbanístico resultaba casi inexistente, permitiéndose alterar y modificar el plano de las ciudades sin miramientos si las necesidades lo imponían, creciendo las ciudades horizontalmente sin límites ni restricciones, pero conservando un casco urbano clásico reservado a las clases más altas de la sociedad.

Sin embargo el crecimiento de las ciudades europeas se vio frenado por los conflictos bélicos del Siglo XIX[8]. La situación empeoró durante la primera mitad del Siglo XX debido a las dos Guerras Mundiales y las guerras civiles que tenían como campo de batalla las ciudades europeas.

Si bien el crecimiento de las ciudades europeas quedó en punto muerto, las vías de transporte entre ellas cobraron vital importancia debido a las necesidades bélicas y logísticas del movimiento de tropas y recursos con los que

8 ↑Por citar algunos: las Guerras Napoleónicas y las Guerras de Independencia Hispanoamericana

abastecerlas. Además los caminos empezaron a ver sustituidos los vehículos de tracción animal por los vehículos de motor. Así por ejemplo, España ya contaba en el año 1900 con 36.306 kilómetros de carreteras asfaltadas [275][9].

[275] Historia de los caminos de España

Durante la segunda mitad del siglo XX debido a la rápida recuperación económica durante la postguerra europea, se vive un nuevo resurgimiento de las ciudades. Las ciudades europeas se ven influenciadas por el concepto Americano de Área Metropolitana, donde la ciudad se vuelve el foco principal de las actividades sociales y económicas. Esto provoca que los sistemas de movilidad interurbana se transformen, ampliándose por ejemplo los servicios de tránsporte público colectivo. En los transportes interurbanos se empiezan a ver los primeros sistemas de vías rápidas y especilizadas como autovías y autopístas. Esto es debido a dos factores socioeconómicos: el transporte tanto de materías primas como de productos manufacturados por carretera y los vehículos de motor privados [170]. Sin embargo la preservación del patrimonio urbanístico cobra mayor importancia, debido a una necesidad de ensalzamiento de los origenes de cada nación, pues la preservación de las calles y las eficiones antiguas se vio fuertemente potenciado por estos sentimientos. De esta forma el plano de las ciudades europeas y sus infraestruturas no fueron modificados, perdurando hoy día.

[170] Ciudad y urbanismo a finales del siglo XX

2.2.2 *Las ciudades europeas actuales y los problemas a los que tienen que hacer frente en el siglo XXI*

Lamentablemente la mayoría de las ciudades europeas han cambiado poco en su distribución y sus infraestructuras durante el Siglo XXI. Las diversas épocas historias del plano de las ciudades europeas pueden leerse como anillos de un árbol, existiendo claras distinciones entre planos ortogonales y planos irregulares. Además, las infraestructuras de las ciudades europeas parten de orígenes medievales o románicos, al contrario que las modernas ciudades norteamericanas fundadas muchas de ellas colonialmente. El sentimiento cultural de la preservación del patrimonio impide las grandes alteraciones en los núcleos urbanos, que han tenido que permanecer inmutables pese a obedecer su planificación a las necesidades de muchos siglos atrás, no a las necesidades actuales y futuras de las ciudades.

Esta antiguedad afecta al rendimiento eficiente de las ciudades europeas. El redimiento de una ciudad depende tanto de su plano e infraestructuras (o capital físico), como de la alta disponibilidad, la calidad del conocimiento sobre la problemática asociada, las vías de comunicación entre servicios y los medios sociales ofrecidos. Estos recursos (denominados capital no físico) resultan decisivos para la mejora de la gestión de las ciudades europeas [44], debido a que cambiar las infraestructuras físicas hereadas de la mayoría de las ciudades resulta prácticamente inviable por el alto coste asociado, así como por el valor histórico y cultural de las mismas. De esta forma, por ejemplo, reestructurar y reorganizar la distribución y plano de las calles de

[44] Smart Cities in Europe

9 ↑Aunque sólo tres vehículos a motor.

las ciudades no es una solución factible ni abordable para mejorar el tráfico o reducir los atascos en la mayoría de escenarios. Se hace necesario, por tanto, mejorar el capital no físico de las ciudades como vía para mejorar su eficiencia .

Las ciudades europeas se encuentran frente a un problema de gestión eficiente, limitadas por las pocas variaciones que pueden realizar en su capital físico, siendo el capital no físico el único en el que fácil y económicamente pueden implantarse mejoras. En la Figura 2.7 se pueden observar los problemas o retos de eficiencia a los que según IBM tendrán que hacer frente las ciudades del futuro.

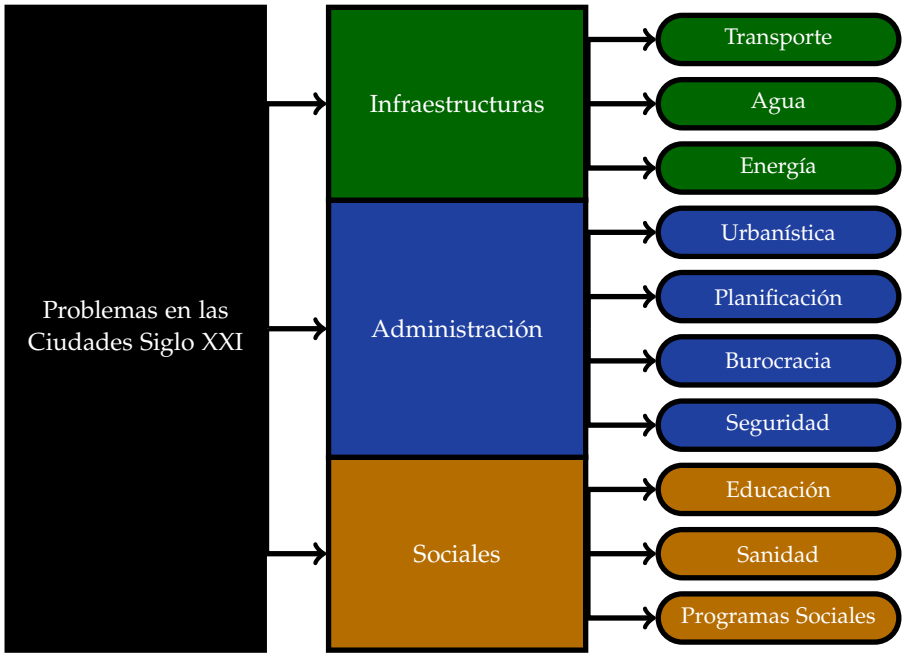


Figura 2.7
Problemas a los que se enfrentan las ciudades del Siglo XXI y que pueden ser más eficientes por mediación de las Smartcities.
Fuente: IBM http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/

Estos retos abarcan problemáticas asociadas a la gestión de recursos e infraestructuras, la planificación y gestión de las administraciones públicas y aspectos directamente relacionados con la sociedad. En cada una de estas problemáticas, se pueden adoptar distintas actuaciones con el fin de mejorar su eficiencia.

Transporte

Las ciudades europeas presentan unos planos irregulares, el trazado sus calles es hererado de necesidades de transporte muy antiguos y se encuentran muy alejados de los volúmenes requeridos por las ciudades actuales y futuras necesitan. En ciudades en las que cada vez se concentran mayor cantidad de personas, el número de desplazamientos es igualmente creciente. Debido a la capacidad de expansión limitada de las vías, en la mayoría de las ciudades europeas, se presentan atacos que saturan las arterias de comunicación de las ciudades. Disponer de una gestión del tráfico eficiente implica ofrecer

rutas más cortas y optimizadas, menores tiempos de desplazamiento, etc. Además, cuanto menor sea el tiempo que esté un vehículo en circulación, menor impacto tiene en la ciudad y menores son sus emisiones contaminantes. Finalmente una gestión eficiente del tráfico repercute en la calidad de vida de los ciudadanos en numerosos aspectos como su salud, la concilización familiar o la incentivación del consumo y el ocio.

Gestión enérgtica y agua

La gestión de la energía consumida por las ciudades y los abastecimientos de agua requeridos, suponen un problema por el costo medioambiental asociados a estos recursos. Hacer una géstión eficiente de la energía y el agua necesarios por una ciudad supone un menor impacto medioambiental, lo cual resulta deseable en todos los casos.

Urbanística

Los edificios de las ciudades del futuro y su distribución deben de ser desarrollados pensando en la gestión eficiente de sus recursos, en la automatización de su mantenimiento y preservación, en la accesibilidad universal y su propia capacidad de autogestión. Se entienden como recursos no solamente los proporcionados por las infraestructuras de las ciudades, sino también los inherentes al edificio como la distribución eficiente del espacio, la luz natural o los materiales empleados.

Planificación y Burocracia

El ciudadano del futuro desea conocer de primera mano y en tiempo real las decisiones de planificación adoptadas por las administraciones dirigentes de sus ciudades. Las ciudades del futuro tienen que hacer una gestión de la información que generan de forma eficiente y transparente para el ciudadano. Además el ciudadano bien informado debe de ser un actor principal en las decisiones de planificación sobre su ciudad. Finalmente las ciudades del futuro tienen la obligación de presentar de la forma más rápida y cómoda los trámites y servicios que ofrece a la ciudanía.

Seguridad

Las ciudades del futuro tienen que disponer de medios y protocolos para reaccionar en casos de emergencias de diferente naturaleza, ya sean desastres naturales, accidentes, incendios, robos, desapariciones, secuestros, etc. En este ámbito las ciudades del futuro deben ofrecer espacios seguros y ser capaces de proteger las infraestructuras y los ciudadanos teniendo tiempos de detección y reacción eficaces con los que solventar las emergencias que surjan.

Educación

Las ciudades y sus administraciones son actores importantes en la educación tanto formativa como cultural de los ciudadanos. Este aspecto cobra mayor importancia en las ciudades del futuro, que tienen que ofrecer una amplia gama de entornos formativos, desde guarderías a universidades, e incluso actividades formativas culturales. Además debido al impacto de las nuevas tecnologías en los ciudadanos, las ciudades y sus administraciones deben ofrecer las pautas y guías de empleo de esas tecnologías por parte sus ciudadanos mediante la formación de las habilidades digitales.

Sanidad y programas sociales

Las ciudades del futuro también participan en la gestión de los servicios sanitarios y programas de acción social. La responsabilidad de gestión eficiente de la información, la demanda asistencial de profesionales, las gestión de incidencias, programaciones de intervenciones médicas, el empleo de recursos, la difusión de campañas de prevención y concienciación ciudadana y muchos otros casos, supone todo un reto para las ciudades del futuro.

La Unión Europea ha volcado sus esfuerzos en el objetivo de preparar a sus ciudades para hacer frente a todos estos problemas; siendo la vía para lograrlo conseguir una mejora significativa de las infraestructuras no físicas, para de esta forma, optimizar y resolver sus limitaciones de eficiencia sin alterar las infraestructuras físicas. Más concretamente, la Unión Europea en el denominado `OECD EUROSTAT Oslo Manual` [196], señala a las Tecnologías de la Información y la Comunicación (o TICs) como el eje central para la mejora de la eficiencia de las ciudades, instando a las TICs a ofrecer indicadores y herramientas para la innovación de los núcleos urbanos con el fin de prepararlos para resolver los problemas futuros.

[196] *Oslo Manual*

Si bien la Unión Europea marca que las TICs tiene un papel protagonista en el devenir de las ciudades en el futuro, existen discrepancias sobre la funcionalidad y características que deben proporcionar a las ciudades. De esta forma han surgido cuatro enfoques diferentes sobre como las TICs deben ser abordadas y aplicadas en el entorno de las ciudades [23]:

[23] *Smart cities of the future*

Ciber cities

Deben su nombre al ciberespacio y la cibernética. Supone la administración y control de las ciudades basada en el retorno de información periódica. Sin embargo, también tiene carácter negativo por su asociación a los cibercrímenes y al cibercontrol, que implica la rotura de derechos y libertades mediante la constante indefiticación, rastreo y control de los movimientos y actividades de los ciudadanos.

Digital cities

Deben su nombre a la digitalización de las ciudades. Implica la creación de ciudades virtuales, realizando la transformación de las ciudades físicas a

ciudades digitalizadas, donde gran parte de la vida en ella y sus interacciones se realiza mediante avatares en una simulación digital.

Intelligent cities

Deben su nombre a la inteligencia colectiva de ciudadanos, inteligencia distribuida, colaboración online, crowdsourcing y capital social de las ciudades. Implica dotar de medios a los ciudadanos para un autocontrol y regulación de forma descentralizada de sus necesidades.

Smart cities

Deben su nombre a los smarphones y smartdevices presentados anteriormente. Supone desplegar por las ciudades sensores, sistemas embebidos, entornos inteligentes e instrumentación que permita recoger información constante y en tiempo real sobre el estado de la ciudad, obteniendo conocimiento sobre esa información y aplicándolo para lograr la mejora de la eficiencia de la ciudad. Eficiencia que se retornaría en mejores servicios para la ciudadanía, así como menores costes tanto económicos, temporales y medioambientales.

A pesar de que todos ellos marcan escenarios donde las TICs pueden ofrecer mejoras en la gestión de las ciudades, el movimiento que más apoyo e interés está recibiendo por los investigadores y las administraciones es el de las Ciudades Intelligentes o SmartCities.

2.2.3 *La solución de los problemas modernos mediante el uso de las TICs: SmartCities*

El concepto de Ciudad Inteligente o `Smartcity` se remonta al Infome Brundtland [59], cuya elaboración fue solicitada por la Comisión Mundial de Medio Ambiente y Desarrollo de Naciones Unidas en el año 1985. En dicho informe se señala la problematica a la que se enfrentarán las ciudades durante el Siglo XXI y la necesidad de hacer uso de un desarrollo sostenible y eficiente de los recursos.

[59] Our common future: Report of the World Commission on Environment and Development

Empresas del sector de las TICs como IBM, CISCO, Siemens o Microsoft alumbraron el concepto de `Smartcity` como solución al problema de la gestión de la eficiencia de las futuras ciudades. Debido a que cada empresa ofrece su propia solución `Smartcity` y a la novedad del término, su definición es aun difusa. Partiendo del trabajo de Taewoo Nam [190] que busca una única definición de `Smartcity` y añadiendo citas adicionales, se presentan las siete definición de `Smartcity` más influyentes[10] y aceptadas actualmente.

[190] Conceptualizing smart city with dimensions of technology, people, and institutions

Definición 1 *Una smart city utiliza la tecnología para prestar de forma más eficiente los servicios urbanos, mejorar la calidad de vida de los ciudadanos y transfor-*

10 ↑Para la unificación del idioma de esta memoria de tesis, se ha optado por presentar una traducción casi literal de las definiciones presentadas en idiomas distintos al castellano

mar la relación entre entidades locales, empresas y ciudadanos facilitando una nueva forma de vivir la ciudad. Gildo Seisdedos et al.[244].

[244] Smart Cities: La transformación digital de las ciudades

Definición 2 *El uso de la tecnología con inteligencia computacional (Smart Computing) para las infraestructuras críticas y servicios de una ciudad (que incluye la administración de la ciudad, la educación, la sanidad, la seguridad, el urbanismo, el transporte y los servicios) de forma más inteligente, interconectada y eficiente. Doug Washburn et al.[288].*

[288] Helping CIOs understand "smart city" initiatives

Definición 3 *Una ciudad bien gestionada, con la vista en el futuro en lo económico, lo social, lo gubernamental, la movilidad, el urbanismo y la calidad de vida de sus habitantes.Rudolf Giffinger et al.[98].*

[98] Smart cities ranking: an effective instrument for the positioning of the cities?

Definición 4 *Una ciudad que monitoriza e integra las condiciones de todas aquellas infraestructuras críticas, incluyendo carreteras, puentes, túneles, railes, metros, aeropuetos, puertos marítimos, comunicaciones, agua, electricidad e incluso grandes edificios de forma que se pueda lograr una mejor optimización de los recursos, planificar tareas de mantenimiento y monitorizar los aspectos de seguridad mientras se maximizan los servicios a los ciudadanos. B Bowerman et al.[32].*

[32] The vision of a smart city

Definición 5 *Una ciudad sensorizada, interconectada e inteligente. La sensorización habilita la captura e integración de datos vivos del mundo real (...) Interconexión significa la integración de todos esos datos en una única plataforma de computación empresarial (...) Inteligencia se refiere a la inclusión de análisis complejos, modelado, optimización y visualización (...). Colin Harrison et al.[114].*

[114] Foundations for smarter cities

Definición 6 *Una ciudad que proporciona inspiración, comparte la cultura, el conocimiento y la vida. Una ciudad que motiva a sus habitantes a crear y hacer florecer sus propias vidas. Patrice Rios [227].*

[227] Creating"The Smart City"

Definición 7 *Una ciudad donde las Infraestructuras Común de Telecomunicaciones se esfuerzan para lograr la liberdad de expresión y la accesibilidad pública de la información y servicios. Helen Partridge [204].*

[204] Developing a human perspective to the digital divide in the'smart city'

Presentadas estas definiciones resulta fácil establacer cuales son las características que los diferentes autores otorgan a una `Smartcity`. Más concretamente, a como las Smartcities tiene que nutrirse de información en tiempo real sobre lo que está ocurriendo en la ciudad. Si bien hay autores que enfocan las Smartcities desde un punto más social y consideran que la interveción y opinión del ciudadano es la mejor fuente de datos, gracias al Internet de las Cosas (que ha sido presentado en la sección 2.1.2) existe otro enfoque más tecnológico. Según este enfoque, la información de las ciudades debe de ser obtenida por toda una red de sensores desplegada en las ciudades.

2.2.3.1 *Redes de sensores para ciudades inteligentes*

Una red de sensores (o Sensor Network) es una red de pequeñísimos ordenadores denominados nodos equipados con sensores, colaborando en una tarea común [14]. Tiene su origen en iniciativas militares, por lo que debido a ello es complicado trazar sus orígenes, aunque el concepto pudo verse originado gracias al del polvo inteligente (o `Smart Dust`) [139, 287].

[14] Wireless integrated network sensors: Low power systems on a chip

[139, 287] Next century challenges: mobile networking for "Smart Dust", Smart dust: Communicating with a cubic-millimeter computer

El objetivo de una red de sensores aplicados a una ciudad inteligente es la de medir parámetros cuyo estudio pueda ser influyente en su gestión, brindando una información actual valiosa para la gestión eficiente. Idealmente todos los datos recogidos son transmitidos en tiempos cercanos al real y se encuentran disponibles tanto para las administraciones como para los ciudadanos.

Entre las características deseables de una red de sensores [139] se encuentran el bajo consumo energético, las limitaciones del hardware y su procesamiento que lleva asociado por el bajo coste de su manufacturación. Además de otros requisitos de cualquier elemento hardware/software como son la tolerancia a fallos.

[139] Next century challenges: mobile networking for "Smart Dust"

2.2.4 *Resumen*

En esta sección se han presentado los problemas a los que se encuentran las ciudades actuales a resolver durante el siglo XXI, problemas causados por la gestión de los recursos medioambientales y el aumento de población previsto de las ciudades. Por las caracteríricas de las ciudades europeas debido a su historia, la mejor resolución de estos problemas radica en la mejora de la eficiencia en la gestión y administración de los recursos e infraestructuras. Se ha encontrado en las TICs una vía para lograr esta mejora en la eficiencia por medio de la conversión de las ciudades actuales en ciudades inteligentes o `smarticities`. A pesar de las discrepancias en la definición de qué es o no es una ciudad inteligente, todos los autores señalan en que esta debe nutrirse de información. Y la mayoría de los autores otorga la potestad de la generación de dicha información a las redes de sensores. Otorgar nuevas fuentes de datos a dichas redes de sensores como la presentada en esta tesis, permite ampliar el especto de magnitudes que pueden ser relevantes para la gestión eficiente de las ciudades. Así como estudiar aspectos nuevos de las mismas.

3

REVISIÓN BIBLIOGRÁFICA

El mejor ordenador debe ser como un sirviente invisible y callado.

— Mark Weiser

En este capítulo se presentan las magnitudes y métricas que han sido empleadas por las administraciones e investigadores para el estudio de la movilidad, muchas de ellas heredadas desde la concepción del automóvil y aplicadas posteriormente a la movilidad de personas.

Se enumeran a su vez las fuentes de datos y las tecnologías que son empleadas actualmente para la obtención de información para el estudio de la movilidad tanto de vehículos como de personas. Se presentan sistemas convencionales, así como sistemas basados en principios funcionales similares al de la fuente de datos presentada en esta tesis: la captación de comunicaciones inalámbricas.

Índice del capítulo

3.1 INTRODUCCIÓN A LA MONITORIZACIÓN

El primer automóvil fue lanzado en el año 1885, desarrollado por el equipo liderado por Karl Benz[41, 76]. Durante los primeros años del Siglo XX, los automóviles se popularizaron llegando al mercado y siendo su presencia en las calles cada vez más habitual. Su impacto fue tal en la sociedad y en la estructura de las ciudades, que se empezó a considerar un problema gestionar el tráfico en las mismas, empezando a ser abordado desde el punto de vista de un problema de ingeniería [209]. A mediados del Siglo XX, se habían sentado las bases de los que serían los primeros sistemas de control del tráfico [269].

[41, 76] Mixed blessing: the motor in Britain, World history of the automobile

[209] Traffic engineering: theory and practice

[269] Road traffic and its control

Con el tiempo, las magnitudes [105] y medidas del tráfico [142] se han ido estandarizando, haciendo que la naturaleza del sistema de monitorización sea irrelevante para los sistemas de control del tráfico y sus investigadores, siempre y cuando dicho sistema de monitorización sea capaz de proporcionar las mágnitudes y medidas que son ampliamente empleadas. Igualmente en los últimos años, se han aplicado estos mismos principios a la monitorización de personas cuando la tecnología existente lo ha permitido.

[105] Traffic Control Systems Handbook

[142] Measures of Effectiveness

En esta sección se recogen dichos principios y fundamentos de la monitorización del tráfico que permanecen aún vigentes, y que son aplicables a otros entornos de monitorización, como la monitorización de personas o dispositivos inteligentes que esta tesis propone.

Se presentarán en primera instancia las magnitudes relativas al tráfico en las que se ha centrado el estudio del mismo. A continuación se presentarán las métricas basadas en dichas magnitudes que son empleadas en los estudios relativos a la gestión del tráfico y el estudio de la eficacia de la implantación de nuevas estrategias.

Se completa la sección con una comparativa de los diferentes y más extendidos sistemas de monitorización del tráfico empleados en las ciudades y carreteras, mostrando sus principales beneficios y limitaciones de uso.

A continuación se detalla como los sistemas de monitorización de vehículos se han adaptados para la monitorización de personas, presentándose tres de los sistemas de monitorización de personas más extendidos.

Finalmente, se presentarán sistemas de monitorización que hacen uso de la captación de comunicaciones inalámbricas, al igual que el propuesto en esta tesis. Se mostrarán alguno de los nuevos estudios emergentes sobre la temática, así como algunas de las soluciones comerciales de `Smartcities` que ofertan productos opacos que ofrecen este tipo de monitorización.

3.2 MAGNITUDES EN EL ESTUDIO DEL TRÁFICO

Para estudiar la movilidad de vehículos, se hace necesario definir unas magnitudes o características que resuman el tráfico de forma precisa. Estas variables de control, son ampliamente usadas por los organismos y administraciones, y proporcionan un lenguaje común a los equipos interdisciplinares que trabajan en la monitorización, administración y control de tráfico. Una nueva tecnología de monitorización debe proporcionar estas magnitudes si se desea que su aplicación al mundo real sea fácilmente implantable. Y además, los algoritmos y metodologías existentes para la gestión y control del tráfico que hagan uso de esas magnitudes, también deben seguir siendo compatibles con la información obtenida por dicho nuevo sistema de monitorización. Las magnitudes relativas al tráfico frecuentemente usadas [105] se recogen a continuación.

[105] Traffic Control Systems Handbook

3.2.1 *Presencia de vehículo*

Esta magnitud refleja la presencia de un vehículo determinado en un punto concreto acotado temporalmente. Aunque es de carácter muy simple, se emplea en multitud de escenarios y supone una primera aproximación funcional muy útil al control de tráfico.

Se presenta en múltiples escenarios en los que es necesario determinar la presencia o no de un vehículo. Los más comunes son los sensores de ocupación cada vez más extendidos en los parkings, semáforos o puestos de peaje. Un ejemplo de este escenario, se puede observar en la Figura 3.1 presentando una disposición típica de un parking, donde se controla el número de plazas de aparcamiento libres en base a los vehículos detectados ocupando plaza.

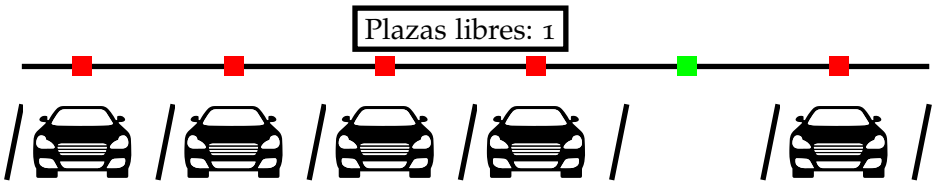


Figura 3.1
Detector de presencia de vehículo usado para indicar los sitios disponibles y ocupados en un aparcamiento.

3.2.2 *Densidad / Volumen del flujo de tráfico*

La densidad o volumen del tráfico se refiere al número de vehículos circulando por un punto determinado de la carretera durante un periodo de tiempo concreto. Viene definido por la Ecuación 3.1, donde Q es el volumen o densidad del tráfico expresado en vehículos por hora. N es el número de vehículos detectados por el sensor durante un periodo de tiempo T.

$$Q = \frac{N}{T}(Vehículos/tiempo) \tag{3.1}$$

3.2.3 *Ocupación*

La ocupación de una vía o carretera es el porcentaje de tiempo que un punto concreto de la misma está ocupado por un vehículo. Se calcula con la Ecuación 3.2 donde θ es la ocupación de la vía expresada en porcentaje. T es el intervalo de tiempo especificado expresado en segundos, t_i es el tiempo muestreado del sensor en segundos, D es el tiempo necesario para la recogida de la muestra y desconexión[1] de la misma y N es el número de vehículos detectados en el tiempo T.

$$\theta = \frac{100}{TL}\sum_{i=1}^{N}(t_i - D)(\,\%) \tag{3.2}$$

La ocupación de la vía es un parámetro de alta transcendencia en el estudio del tráfico, ya que sirve como indicativo de la utilización real de la vía lo que habilita la detección de atascos, retenciones o usos de la vía anómalos.

3.2.4 *Velocidad*

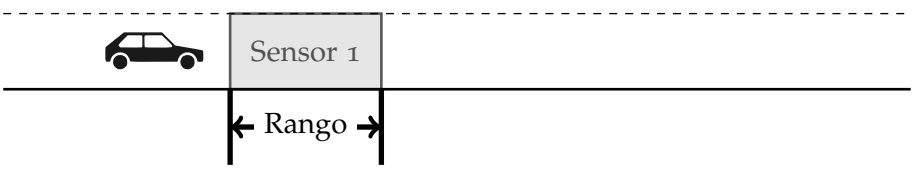
La velocidad de un vehículo en movimiento se define como la distancia recorrida por dicho vehículo en una unidad de tiempo determinada. En los sistemas de monitorización de tráfico se puede calcular tanto empleando un único sensor como haciendo uso de dos o más sensores sucesivos. En ambos casos el cálculo es el mismo, siendo el mostrado en la Ecuación 3.3. Esta ecuación está expresada en millas por hora debido a que la bibliografía existente suele ser de procedencia americana. El equivalente de este cálculo expresado en kilómetros por hora se presenta en la Ecuación 3.4.

$$V = \frac{3.6 \times 10^6 D}{5280(t_1 - t_0)}(Millas/horas) \tag{3.3}$$

1 ↑Es decir, el tiempo mínimo necesario para que el sensor pase de detección a no detección.

$$V = \frac{3.6 \times 10^3 D}{(t_1 - t_0)} (Kilómetros/hora) \tag{3.4}$$

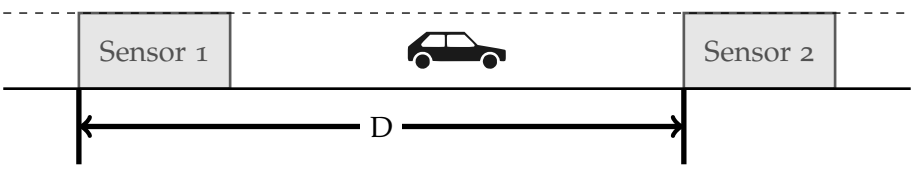
En el caso de hacer el cálculo con un único sensor, nos encontramos con un escenario como el reflejado en la Figura 3.3, donde D es la media de las longitudes de los vehículos más el rango efectivo de detección del sensor, expresado en pies (ecuación 3.3) o metros (ecuación 3.4). El instante t_0 es la marca de tiempo del sensor en el momento de su activación, expresado en milisegundos. El instante t_1 es la marca de tiempo del sensor en el momento en que el este se desactiva.



$$Velocidad = \frac{Longitud\ vehículo + Rango}{Tiempo\ Final\ Sensor - Tiempo\ Inicio\ Sensor} (Km/h)$$

Figura 3.2
Cálculo de velocidades haciendo uso de un único sensor o detector.
Fuente Original: Detector Technology Evaluation[174]

El cálculo de la velocidad con un solo detector implica componentes externos, que añaden ruido a la medición: la longitud media de los vehículos y el rango de detección del sensor. Por esta razón que existe una segunda aproximación al cálculo de la velocidad mucho más precisa, empleando dos sensores desplegados consecutivos. En ese caso, d es la distancia entre un sensor y el otro (en pies o metros en función de la ecuación). El instante de tiempo t_0 en el que el primer sensor se activa detectando un vehículo. El instante de tiempo t_1 es el instante de tiempo en el que el segundo sensor detecta al mismo vehículo. Un ejemplo gráfico de este cálculo se puede observar en la Figura 3.2.



$$Velocidad = \frac{D}{Tiempo\ Inicio\ Sensor\ 2 - Tiempo\ Inicio\ Sensor\ 1} (Km/h)$$

Figura 3.3
Cálculo de velocidades haciendo uso de dos sensores o detectores secuenciales.
Fuente Original: Detector Technology Evaluation[174]

El cálculo de la velocidad haciendo uso de dos o más sensores consecutivos presenta también sus desventajas. La principal de ellas, es que se debe garantizar que el vehículo detectado por ambos sensores es el mismo. Esto se resuelve en la mayoría de los sistemas, reduciendo la distancia D al mínimo, garantizando que ningún otro vehículo ha podido interponerse entre ambos sensores en tan corta distancia.

Existe una tercera alternativa para el cálculo de la velocidad (Ecuación 3.3), haciendo uso del volumen de tráfico Q y la ocupación θ, pues existe un factor constante C que puede ser medido y cuantificable de forma experimental.

$$V = C \times \frac{Q}{\theta}(kilómetros/hora) \tag{3.5}$$

3.2.5 *Densidad*

La densidad K se calcula como el número de vehículos por milla/kilómetro de carril. Se calcula como muestra la Ecuación 3.6, siendo Q el volumen del tráfico (en vehículos por hora) y $\overline{U}s$ la velocidad media espacial, expresada en $Millas/h$ o $Kilómetros/h$ dependiendo del sistema empleado.

$$Q = K \times \overline{U}s(kilómetros/hora) \tag{3.6}$$

La densidad K puede ser calculada directamente en función de el número de vehículos N detectados durante un instante de tiempo T y las velocidades V_i individuales de todos los vehículos detectados tal y como se muestra en la Ecuación 3.7.

$$K = \left(\frac{1}{T}\right) \sum_{i=1}^{N} \left(\frac{1}{V_i}\right) (vehículos/hora) \tag{3.7}$$

A pesar de que la densidad es un factor muy importante y decisivo en el control del tráfico, la mayoría de los sistemas de monitorización no hacen uso de esta magnitud debido a que no es fácilmente interpretable y requiere recopilar los tiempos individuales de los vehículos.

3.2.6 *Determinación del avance del tráfico*

El concepto de avance del tráfico se determina mediante el tiempo que transcurre entre dos vehículos consecutivos detectados, generalmente, en un carril determinado. Un ejemplo gráfico se puede ver en la Figura 3.4.

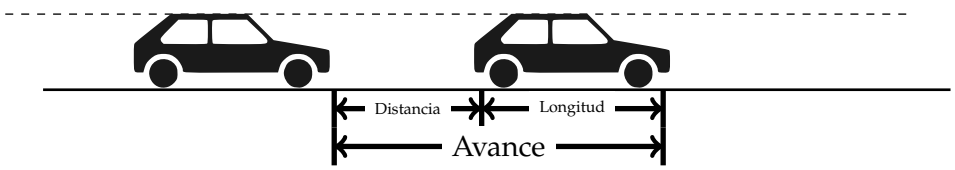


Figura 3.4
Determinación del avance del tráfico de vehículos.
Fuente original: Detector Technology Evaluation [174]

3.2.7 *Longitud de las colas*

Número de vehículos detenidos en un carril en una señal de stop o paso de peatones. Esta magnitud es difícilmente obtenible por la mayoría de los sistemas de monitorización existentes, debido a que están limitados a la observación de una región espacial limitada. Es por ello, que esta magnitud suele saturar a partir de un determinado valor de vehículos, indicando que hasta donde el sensor puede incidir, existen vehículos detenidos.

3.3 MEDIDAS DE EFECTIVIDAD EN EL CONTROL DEL TRÁFICO

Uno de los principales usos de los sistemas tanto de monitorización como de control de tráfico, es el de optimizar las vías de comunicación. El objetivo de esa optimización puede ser algo tan concreto como reducir los atascos y retenciones en una calle, optimizar los tiempos de los semáforos para reducir las paradas, incrementar la accesibilidad de las vías para incrementear la velocidad de las mismas o disminutir el tiempo necesario para desplazarse de un sitio a otro. Sin embargo, este tipo de objetivos que en apariencia son fáciles de definir, resultan complicados de medir y cuantificar.

[69, 209] Manual of Uniform Traffic Control Devices for Streets and Highways, Traffic engineering: theory and practice

Las medidas de efectividad (`Measures of effectiviness` o MOE)[69, 209] proveen un conjunto de métricas con la que cuantificar el ajuste de las estrategias adoptadas para conseguir el objetivo deseado. De esta forma realizando el cálculo de las MOEs antes y después de la implantación de una nueva medida relativa al tráfico, se puede estudiar el impacto que ha supuesto dicha estrategía al tráfico real de vehículos. Y verificar si está mejorando el tráfico adecuándolo al objetivo marcado.

Dado que las MOEs están directamente relacionadas con la efectividad de las estrategías implicadas, su selección supone un problema crítico. Un criterio deseable para la selección de dichas MOEs incluiría o tendría presente los siguientes requerimientos:

- Trato simple de las limitaciones propias de la precisión y exactitud del sistema de monitorización.
- Sensibilidad a cambios pequeños para un mejor estudio de la influencia de la estrategia en el tráfico.
- Posibilidad de medición en una escala cuantitativa dentro de un tiempo, coste y mano de obra asumibles.

Estos requerimientos en las MOE implican directamente al sistema de monitorización que se emplee para la obtención de las magnitudes del tráfico con las que se calcularán dichas medidas de efectividad. La adecuación del sistema de monitorización, se considera por tanto también crítico para la tarea de la gestión y estudio del tráfico.

[105] Traffic Control Systems Handbook

Las medidas de efectividad más comunes [105] se presentan a continuación.

3.3.1 *Tiempo total de viaje*

El tiempo total de viaje (Total Travel Time o TTT) es uno de las principales medidas de efectividad, empleada tanto en el tráfico urbano como en el tráfico en vías de alta capacidad. Se expresa en vehículos/hora y representa el producto del total de los vehículos circulando por la vía durante un tiempo determinado y el tiempo de viaje promedio empleado por dichos vehículos.

El tiempo de viaje promedio tt_j expresado en horas de una tramo de vía j se calcula según la ecuación Ecuación 3.8:

$$tt_j = \frac{X_j}{u_j} \tag{3.8}$$

Donde X_j es la distancia del tramo de vía expresada en kilómetros y u_j es la velocidad promedio de los vehículos circulando sobre dicho tramo. Se define el tiempo total de viaje como la Ecuación :

$$TTT_j = N_j \times tt_j = \frac{N_j \times X_j}{u_j} \tag{3.9}$$

Donde N_j es el número de vehículos circulando por dicho tramo durante el periodo de tiempo determinado. El tiempo total de viaje en vehículos/hora para todas las secciones de un vía puede ser calculado como muestra la Ecuación 3.10:

$$TTT = \sum_{j=1}^{K} TTT_j \tag{3.10}$$

Siendo K el número de tramos que componen dicha vía.

3.3.2 *Viajes totales*

El número de viajes totales es otra MOE usada para evaluar el impacto de nuevas estrategias en el tráfico. Se expresa en unidades de vehículos por kilómetros y representa el producto del número total de vehículos usando la vía durante un periodo de tiempo acotado por la distancia del viaje de dichos vehículos.

El total de viajes (Total Travel o TT) en vehículos/kilómetros para una tramo j de una vía se calcula como muestra la Ecuación 3.11:

$$TT_j = X_j \times N_j \tag{3.11}$$

Donde X_j es la longitud total del tramo de vía expresado en kilómetros y N_j es el número de vehículos circulando en dicho tramo durante un periodo de tiempo acotado.

Las Ecuaciones 3.13 y 3.9 sugieren que el total de viajes o TT expresado en vehículos por kilómetro sobre un tramo j puede ser obtenido partiendo del

TTT_j y la velocidad promedio en dicho tramo, tal y como se presenta en la Ecuación 3.12:

$$TT_j = TTT_j \times V_j \tag{3.12}$$

Donde TTT_j es la MOE introducia en el apartado anterior y V_j es la velocidad promedio de los vehículos circulando sobre el tramo j durante un periodo de tiempo acotado.

El TT o tiempo de viaje para todos los tramos de una vía se calcularía como:

$$TT = \sum_{j=1}^{K} TT \tag{3.13}$$

Donde K es el número de tramos que componen dicha vía.

3.3.3 *Número y porcentaje de paradas*

Esta MOE se emplea para el control de la calidad del flujo del tráfico sobre todo en entornos urbanos. El número de paradas puede ser obtenido mediante el empleo de vehículos flotantes o mediante la observación directa de las intersecciones existentes. El sistema de control del tráfico empleado, debe registrar las paradas de los vehículos. Expresa el porcentaje de paradas (principalmente por el efecto de semáforos o intersecciones) que el vehículo encuentra en su trajecto y que se ha visto obligado a realizar.

Una estrategia que optimice el flujo del tráfico, minimizaría esta medida. Siendo un tráfico ideal aquel que circula sin la necesidad de realizar ninguna detencción.

3.3.4 *Retraso de la circulación*

Empleada también en entornos urbanos, esta medida se define como el incremento de tiempo sufrido en el tiempo de viaje correspondiente a una velocidad de circulación por debajo de esperada, denominada velocidad base. Circulaciones por debajo de dicha velocidad se considerarían que están retrasando el desplazamiento.

Para las intersecciones urbanas, el retraso se define también como el tiempo perdido por aquellos vehículos detenidos en las intersecciones mediante semáforos[35].

[35] *Manual of traffic engineering studies*

3.3.5 *Velocidad promedio*

Es uno de los MOE más descriptivos y se emplea tanto en el estudio de nuevas estrategias tanto en entornos urbanos como en vías de alta capacidad como carreteras y autovías. Muestra correspondencia directa con una de las magnitudes del tráfico presentadas anteriormente (Velocidad en Sección 3.2.4).

Además, muestreos puntuales de la velocidad promedio en un punto concreto o el estudio de las velocidades individuales de los vehículos pueden detectar áreas problemáticas y proveer de nuevas medidas avanzadas[142] para estudiar el rendimiento de las estrategias.

[142] Measures of Effectiveness

Ratio de accidentes

Minimizar el ratio de accidentes y su mortalidad es uno de los objetivos principales en las estregias relativas a la gestión del tráfico. El ratio de accidentes está usualmente expresado en términos de accidentes por millón de vehículos en entornos urbanos. En vías de alta capacidad, se expresa en accidentes por cada 100 millones de vehículos por kilómetro recorrido.

3.3.6 *Rendimiento*

Si bien el rendimiento está en apariencia relacionado con la velocidad con la que circulan los vehículos y el número de estos, se emplea de manera diferente, permitiendo comparar como han afectado distintas estretegias de gestión del tráfico a un mismo escenario. Se calcula puntualmente como presenta la Ecuación 3.14.

$$Rendimiento = \frac{Km\ por\ Vehículo\ por\ unidad\ de\ tiempo}{Horas\ por\ vehículo\ por\ unidad\ de\ tiempo} \tag{3.14}$$

La Figura 3.5 representa el rendimiento de dos estrategias de gestión en un mismo escenario, uno empleado como referencia (curva A) y otro haciendo uso de una estregia para mejorar la fluidez del tráfico (curva B). Esas curvas representan el mejor ajuste matemático de los datos representados por un conjunto de medidas realizadas individualmente. El rendimiento del tráfico se determina por la pendiente de la recta en un punto dado de la curva. Cuando el tráfico se incrementa, el rendimiento del tráfico empieza a decrecer.

Esta aproximación permite el estudio del rendimiento del tráfico de forma precisa, midiendo los beneficios de las estrategias adoptadas en un escenario en diferentes entornos. Por ejemplo, en las curvas mostradas en la Figura 3.5, si el objetivo de la estrategia emprendida en B es el de minimizar la congestión del tráfico, examinando la curva B se puede observar que la

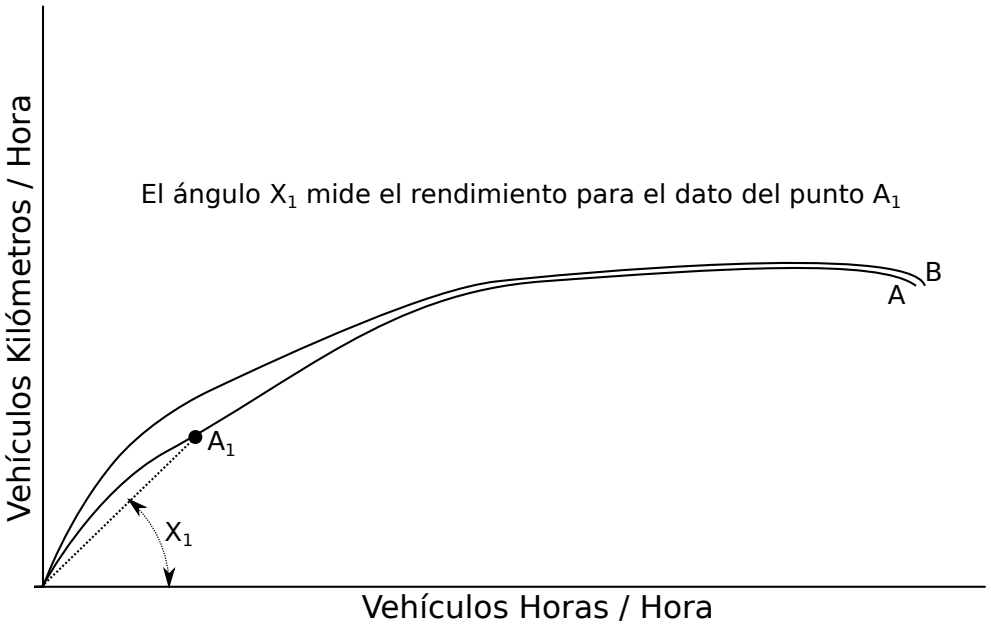


Figura 3.5
Curvas de Rendimiento, MOE empleada para la comparación analítica del impacto de diferentes estrategias.
Fuente Original: [105].

mejora conseguida es mínima en las zonas de tráfico congestionado, por lo que la estregia empleada no ha satisfecho el objetivo propuesto.

3.4 TECNOLOGÍAS EMPLEADAS PARA LA DETECCIÓN TRÁFICO

Históricamente se han empleado una gran cantidad de tecnologías para la detección y monitorización del tráfico de vehículos [105, 150]. Estos sistemas pueden ser clasificados según la inmediatez de la obtención y la exhaustividad de los datos, así como por lo intrusivo que resulta el sistema para la vía [174].

[105, 150] Traffic Control Systems Handbook, Sensor technologies and data requirements for ITS

[174] Detector technology evaluation

Inmediatez

Atendiendo a la inmediatez de la recolección de los datos, los sistemas se clasifican como Sistemas de Recolección Directa donde la fuente obtiene los datos experimentalmente midiendo una magnitud física; y los sitemas de Recolección Indirectos, donde los datos se obtienen tras un procesamiento algorítmico posterior.

Por ejemplo, un dispositivo situado en la carretera contando ejes necesita un procesamiento posterior para inferir el número de vehículos, por lo que se clasifica como un sistema de recolección indirecto.

Exhaustividad

En función de lo exhaustiva que sea la fuente de datos, los sistemas se pueden clasificar como sistemas casi exhaustivos en donde la magnitud reflejada por la fuente de datos es igual a la real; o sistemas no exhaustivos, en donde la magnitud de la fuente de datos está en otra escala, se encuentra desplazada o solo recoge una muestra estadística del tráfico real.

Por ejemplo, un sistema de monitorización que solo sea capaz de reconocer vehículos que cumplan una condición o restricción específica, como disponer de un hardware adicional o superar un peso umbral, es considerado un sistema de monitorización no exhaustivo.

Intrusividad

Finalmente los sistemas de monitorización son clasificados en función de su intrusividad con la vía donde son instalados. Los métodos intrusivos son instalados directamente en el pavimento, ya sea sobre la carretera o por debajo de ella mediante la altareación de la misma.

El principal problema de los sistemas intrusivos, es que afectan al tráfico [174], debido principalmente a que para su implantación y tareas de mantenimiento suelen necesitar cortar el acceso al tráfico. Los sistemas no intrusivos no requieren contacto directo con la calzada, por lo que causan un impacto mínimo en el flujo del tráfico.

[174] Detector technology evaluation

Existe una tercera categoría en esta clasificación, en los que es el propio vehículo el que dispone del sistema de monitorización, no requiriéndose ninguna interacción con la calzada. Estos sistemas se denominan de `vehículo flotante`. En la Figura 3.6 se puede observar algunos de los sistemas de monitorización existentes clasificados según está topología.

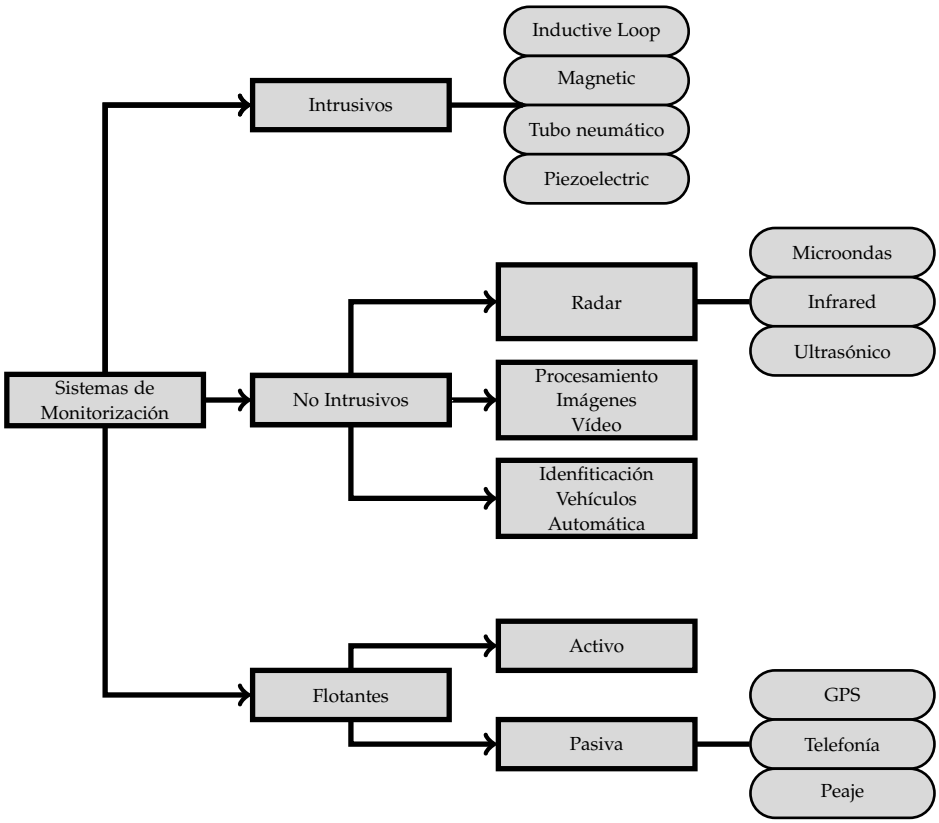


Figura 3.6
Clasificación de los sistemas de monitorización en base a su intromisión.
Fuente Original: Detector Technology Evaluation [174] - Figure 4.31.

A continuación, se presentarán las principales tecnologías y sistemas de monitorización empleados actualmente para el control del tráfico de vehículos [174]: los tubos neumáticos, las bobinas de inducción y los sistemas de captación de imágenes de vídeo [230].

[174] Detector technology evaluation

[230] The geography of transport systems

3.4.1 *Tubos neumáticos*

Los tubos neumáticos son dispositivos de medición del paso de vehículos basados en el conteo del número de ejes que circulan por una vía. Es un sistema intrusivo, ya que es necesario que el sistema de monitorización esté en contacto directo con el firme de la carretera.

Sus principios fundamentales son bastante simples, pues se componen de un tubo de goma cerrado que se coloca transversalmente sobre la calzada [137] tal y como se muestra en la Figura 3.7. El tubo es flexible y se encuentra relleno de un fluido (generalmente aceite o aire) a una presión constante, denominada presión de calibración. Cuando un vehículo circula por encima del tubo, el peso del vehículo transmitido al tubo mediante las ruedas incrementa la presión del aire o fluido. Esta variación en la presión hace que una membrana se extienda o retraiga en función de la presión a la que se encuentre el tubo. Superado un umbral de presión, la membrana alcanza su máxima separación

[137] Gestión Técnica Tráfico - Temario Oposiciones

provocando el cierre un circuito eléctrico, compuesto por dos elementos conductores, uno alojado al final del tubo y otro en la propia membrana.

Figura 3.7
Fotografía de un tubo neumático instalado en la calzada.
Fuente: Jean-Francois Rheault, Eco-Counter.

Al unirse los dos elementos conductores un dispositivo contador se acciona por el cierre de dicho circuito y contabiliza el paso del eje de un vehículo (Figura 3.8). Sistemas más modernos, disponen también en un reloj que refleja el instante de tiempo durante el que se ha producido el cierre del circuito. Actualmente se disponen de aforadores que permiten conectar diversos tubos neumáticos y establecer mediante electrónica las variaciones de presión umbrales, prescindiendo del mecanismo de la membrana y por lo tanto haciéndolos más robustos y más fácilmente calibrables.

Este tipo de sistemas no son inmediatos, debido a que se contabilizan ejes de vehículos y posteriormente se establece que al detectar el paso de dos ejes consecutivo (delantero y trasero) se contabiliza el paso de un vehículo. Este modo de conteo es uno de las mayores limitaciones de esta tecnología, que solamente es capaz de contar ejes. Dado que existen vehículos con más de dos ejes (como autobuses, furgonetas, camiones,...) el sistema introduce pequeños errores de conteo. Estos errores se pueden intentar corregir mediante post-procesamiento, haciendo uso de los instantes de tiempos de los datos obtenidos, la distancia de seguridad y la longitud media entre los ejes de los vehículos, para aproximar si mediciones consecutivas son fruto del mismo vehículo o de otro vehículo distinto. En estos casos, se muestran informaciones de dos tipos de vehículos vehículos cortos y vehículos largos (de más de 5 metros).

Este tipo de tecnología es muy útil en instalaciones provisionales o de corta duración, ya que los tubos son emplazados de forma rápida y sin requerir obra en la calzada. Además, pueden funcionar de forma aislada y autónoma haciendo uso de una batería. Son bastante portables, lo que permiten que pueda variar el sitio donde se implanta en función de donde se desee medir.

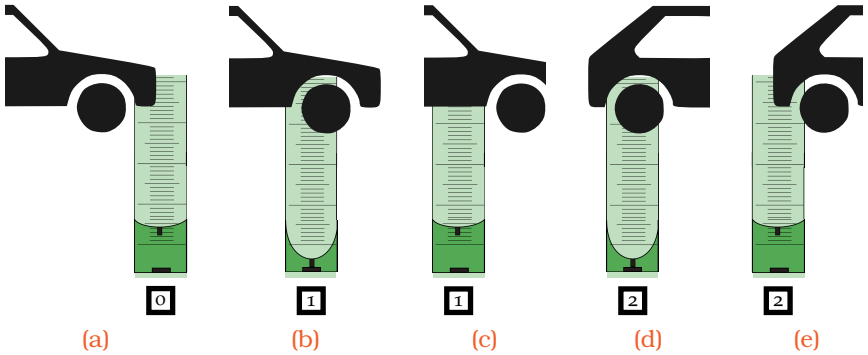


Figura 3.8
Funcionamiento de un tubo neumático. En (a) el vehículo se aproxima al tubo neumático, que se encuentra con la presión base de calibración, por lo que la membrana se encuentra en posición de equilibrio. En (b) al circular el vehículo por encima, el peso transmitido por las ruedas hacen aumentar la presión del fluido en el interior del tubo, esto hace que la membrana se expanda. Al superar la variación de presión umbral, la membrana toca el interior del tubo, cerrando el circuito, lo cual activa un pulso en un contador, incrementando en uno el valor almacenado. En (c) el primer eje de ruedas ha circulado sobre el tubo, que se encuentra entre los dos pares de ejes del vehículo. Al no haber ningún peso encima del tubo, la presión vuelve a normalizarse y la membrana se relaja. En (d) el vehículo termina de pasar por encima del tubo, por lo que el eje trasero del vehículo pasa por encima del tubo, aumentando nuevamente la presión, haciendo que la membrana vuelva a cerrar el circuito, lo cual activa otro pulso del contador. En (e) el vehículo ha terminado de circular por encima del tubo neumático. El tubo contabilizaría dos pulsaciones, referido a que dos ejes han circulado por encima del tubo.

Su precio es bastante económico debido a la simpleza de su mecanismo, lo que lo convierte en un sistema de monitorización bastante extendido.

Sin embargo, son muy frágiles dado que el tubo de goma se encuentra expuesto a la intemperie y además sufre múltiples cambios de presión con cada impacto de las ruedas de un vehículo. Por el hecho de medir variaciones en la presión, necesita calibraciones periódicas, pues los factores externos como la climatología pueden afectar a la precisión de instrumental ya que afectan a la presión base. De igual manera sufren deterioros al estar expuestos y son susceptibles de actos vandálicos.

Los tubos pueden ser también enterrados bajo el pavimento, pero están expuestos al mismo problema de la fragilidad, por lo que no suelen ser la opción predilecta para su implantación en carreteras, debido a que deben ser calibrados y mantenidos con frecuencia. El instalarlos debajo del firme, implica que cualquier tarea de mantenimiento requiera cortar la vía al tráfico.

En la Tabla 3.1 se recogen las principales fortalezas y debidades de estos sistemas de monitorización basados en tubos neumáticos.

Tabla 3.1
Tubos neumáticos: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Fácil implantación. Puede ser implantado sobre la calzada o por debajo de ella. • Bajo coste. • Sistema de monitorización muy extendido, con multitud de soluciones disponibles en el mercado.	• Contabiliza ejes de ruedas, lo cual afecta a la precisión del sistema. • Requiere mantenimiento, tanto por las tareas de calibrado como por su fragilidad aparente. • Su instalación y tareas de mantenimiento requiere cortar la vía al tráfico. • Vulnerable a estrés por el exceso de tráfico y factores climatológicos. • Imposible reconocer al mismo vehículo en visitas sucesivas.

3.4.2 *Bobinas de inducción magnética*

Las bobinas de inducción o espiras [161, 163, 212] se basan en el principio físico de la inducción magnética. En concreto se basa en la capacidad de los materiales ferromagnéticos de alterar el campo magnético de una bobina cuando se encuentra cerca de ella. Las bobinas de inducción o espiras magnéticas, constan de un malla de cable conductor enterrada bajo el pavimento o desplegada encima del mismo[2] como se representa en la Figura 3.9.

[161, 163, 212] Inductive loop sensor for traffic detection, and traffic monitoring apparatus and method using such a loop sensor, A review of magnetic sensors, The future of magnetic sensors

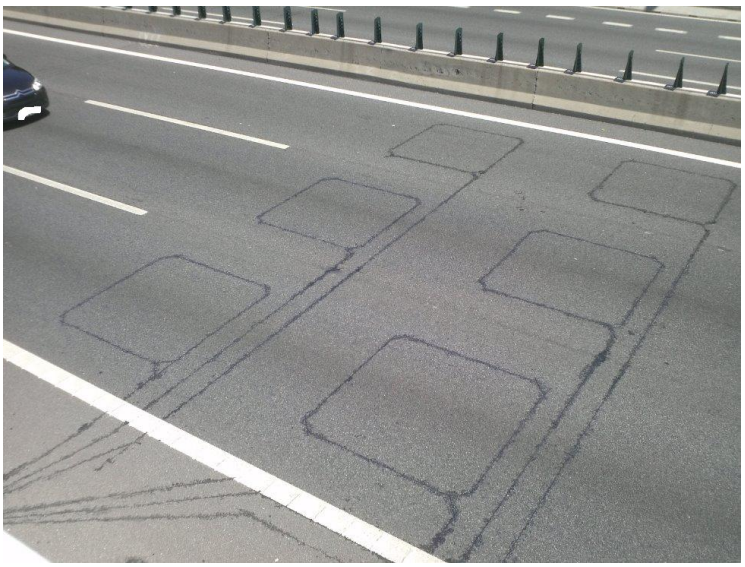


Figura 3.9
Fotografía de seis bobinas de inducción magnética o espieras instaladas bajo la calzada.
Fuente: NOSMAN, Clubastrah.net.

2 ↑Existen también alternativas más compactas y portables que son atornilladas sobre el firme para permitir estudios offline durante cortos periodos de tiempo, sin embargo no están tan extendidas.

Sobre dicha malla, circula una corriente eléctrica que es registrada constantemente por un sensor voltímetro. Cuando una masa metálica circula por encima de la malla, se produce una variación en la corriente eléctrica que es registrada por el sensor (Figura 3.10). En base a esta información, se infiere cuando un vehículo ha circulado por encima de la bobina magnética. Los sistemas habitualmente anotan el tiempo en el que se ha producido la variación y cuanto tiempo ha transcurrido hasta que la corriente eléctrica ha vuelto hasta su valor inicial. De esta forma, también se conoce el tiempo durante el cual el vehículo ha estado sobre la malla, por lo que se puede dar una aproximación de la velocidad (Ecuación 3.3)(Figura 3.2).

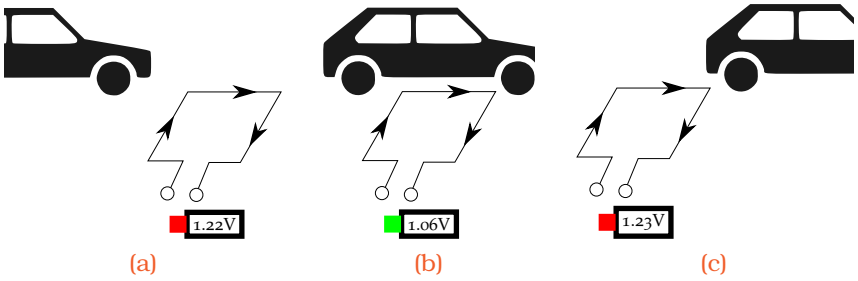


Figura 3.10
Funcionamiento de una bobina de inducción magnética o espira:
En (a) el vehículo aún no se ha situado sobre la espira, por lo que la corriente que circula sobre la misma permanece en el valor de calibración. En (b) al circular el vehículo por encima de la espira metálica, la corriente que circula por la malla se ve afectada por inducción magnética por la masa metálica encima de ella (el vehículo). Si la variación de corriente supera un umbral, se almacena el paso del vehículo. En (c) el vehículo ha dejado de circula por encima de la malla, por lo que la corriente que circula por ella vuelve a estabilizarse.
(Los valores de voltaje empleados en la figura son valores de ejemplo, no valores tomados de escenarios reales.)

Sin embargo, es habitual disponer de pares de mallas consecutivos, con el fin de permitir el cálculo más de preciso de la velocida (Figuras 3.9 y 3.3).

La principal desventaja de estos sistemas de monitorización, como con los tubos neumáticos, radica en que no pueden volver a identificar al mismo vehículo en visitas sucesivas. Esto implica que, en la práctica, sólo permiten cuantificar el número de vehículos y la velocidad. Limitando así la cantidad de información que se puede obtener, como matrices de entrada y salida veraces (no basadas en volumen) o reconstrucción de rutas. Haciendo imposible determinar si todo el tráfico ha sido generado por un único vehículo "dando vueltas" o más de uno..

Los costes asociados son bastante altos debido a que se implantan en la propia vía, que implica que en la gran mayoría de los casos, se requiere obra civil para su instalación, sus tareas de mantenimiento requieren cortar el tráfico en la vía y tienen costes altos asociados[180, 248].

[180, 248] Kilómetros a precio de Oro, "State-of-the-ArtReport on Non-Traditional Traffic Counting Methods

Sin embargo, son sistemas muy preciosos y robustos, por lo que son principalmente empleados en carreteras con grandes volúmenes de tráfico, como autovías y autopistas, siendo el principal y más extendido sistema para el conteo del tráfico empleado mundialmente [53, 55, 94, 161].

[53, 55, 94, 161] Traffic measurement and vehicle classification with single magnetic sensor, Using dual loop speed traps to identify detector errors, A vehicle classification based on inductive loop detectors, Inductive loop sensor for traffic detection, and traffic monitoring apparatus and method using such a loop sensor

En la Tabla 3.2 se recogen las principales fortalezas y debidades de estos sistemas de monitorización basados inducción magnética.

Tabla 3.2
Bobina de Inducción Magnética: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Diseño flexible para satisfacer una gran variedad de escenarios. • Tecnología muy madura y estudiada. • Provee información básica (volumen, presencia, ocupación, velocidad, avance y colas). • Tolerable a las condiciones meteorológicas como la lluvia, niebla o nieve. • Alta fidelidad en el conteo. • Estándar para la obtención de datos preciosos.	• Imposible reconocer al mismo vehículo en visitas sucesivas. • Su instalación requiere manipular (perforar) el pavimento. • Su uso reduce la esperanza de vida del pavimento. • Su instalación y tareas de mantenimiento requiere cortar la vía al tráfico. • Vulnerable a estrés por el exceso de tráfico y las temperaturas. • Se requieren múltiples detectores para un funcionamiento correcto. • La precisión de la detección se puede ver comprometida detectando vehículos largos de distintas clases.

3.4.3 *Sistemas de monitorización basado en reconocimiento de imágenes de vídeo*

Los sistemas de monitorización basados en el reconocimiento de imágenes de vídeo se han vuelto muy pupulares en las tareas de monitorización del tráfico, principalmente por su rápida respuesta y su capacidad para monitorizar grandes áreas o zonas (por ejemplo, todos los carriles de una autovía) con una única cámara[141].

[141] A survey of video processing techniques for traffic applications

Además, estos sistemas se han convertido en sistemas muy estudiados gracias a los avances en intestigación de vehículos de conducción autónoma [109] donde el reconocimiento de imágenes cobra una vital importancia para determinar donde se encuentra el vehículo autónomo y para el reconocimiento de otros vehículos en movimiento [56] u obstáculos que el vehículo autónomo debe evitar[125].

[109] How google's self-driving car works

[56] A real-time computer vision system for vehicle tracking and traffic surveillance

[125] Trajectory tracking and obstacle avoidance of car-like mobile robots in an intelligent space using mixed H 2/H infinite decentralized control

Se clasifican como sistemas de monitorización exahustivos, no intrusivos y de recolección indirecta. Esto se debe a que la imagen capturada debe ser procesada para aproximar de forma algorítmica el número de vehículos circulando. Este procesamiento se divide en dos partes: reconocimiento de terreno y detección de vehículos. Además, muchos sistemas necesitan realizar transformadas para convertir la perspectiva 3D obtenida por la cámara a un plano 2D más facilmente procesable [38] (Figura 3.11).

[38] Vision-based road detection in automotive systems: A real-time expectation-driven approach

En función del movimiento existente en las cámaras, se distinguen entre sistemas de monitorización basados en cámara fija o estacionaria y sistemas basados en cámaras móviles [141]. En los sistemas de cámara móviles,

[141] A survey of video processing techniques for traffic applications

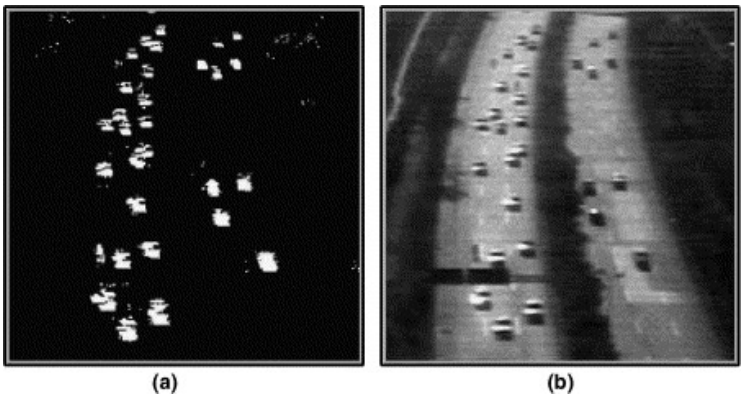


Figura 3.11
Procesamiento realizado por un sistema de monitorización basado en captación de imágenes de vídeo. En (b) se puede apreciar la imagen real capturada por la cámara, y en el (b) el resultado del procesamiento realizado para localizar los vehículos.
Fuente: Real time classification and tracking of multiple vehicles in highways [218] Figura 1.

mediante barridos de la cámara se compone una imagen mayor, pudiendo abarcar mayor superficie, pero añadiendo más complejidad al procesamiento.

Debido a que una cámara recoge una sucesión de imágenes a lo largo del tiempo, existen algoritmos que permiten determinar la ruta que está siguiendo un objeto a través del plano captado por la cámara [56, 189]. Esto permite el estudio de las decisiones tomadas en intersecciones [285] y medir la fluidez del movimiento de los vehículos [138] (Figura 3.12).

[56, 189] A real-time computer vision system for vehicle tracking and traffic surveillance, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences

[285] Video image vehicle detection system for signaled traffic intersection

[138] A feature-based vehicle tracking system in congested traffic video sequences

Existen sistemas que permiten registrar el movimiento de los vehículos entre distintos planos de cámara, haciendo uso del reconocimiento de la matrícula [9, 50] mediante técnicas de reconocimiento de caracteres (Optical character recognition u OCR)[187], sin embargo requieren imágenes de una calidad superior (mayor cantidad de píxeles) para poder distinguir la matrícula, por lo que son sistemas más caros, y por lo tanto menos extendidos. Además, debido a que se disponen de imágenes de los vehículos captados, se pueden emplear algoritmos de clasificación que reconozcan el tipo de vehículo que ha sido detectado [58].

[9, 50] License plate recognition from still images and video sequences: A survey, Automatic license plate recognition

[187] Optical character recognition

[58] A system for video surveillance and monitoring

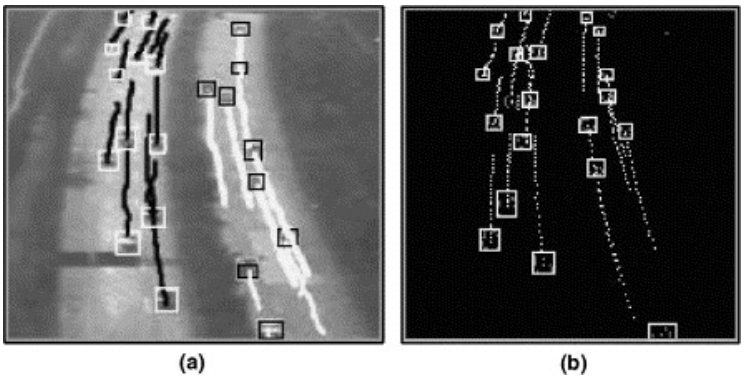


Figura 3.12
Seguimiento o tracking realizado por un sistema de monitorización basado en captación de imágenes de vídeo. Las líneas dibujadas indican la posición del vehículo registrado en momento anteriores del plano de vídeo.
Fuente: Real time classification and tracking of multiple vehicles in highways - Figura 7 [218] .

Tabla 3.3
Sistemas de reconocimiento de imágenes: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Capaz de detectar múltiples carriles y áreas con una sola cámara. • Facilidad para añadir zonas de notificación en caso de detección. • Genera gran cantidad de datos. • Permite unir varias cámaras para monitorizar grandes localizaciones. • Generalmente coste asumible para abordar múltiples zonas a monitorizar (carriles).	• Instalación y mantenimiento muy costoso. • El rendimiento se ve muy afectado por las condiciones meteorológicas. • La lente de la cámara es suceptible de sufrir imperfectos, suciedad u objetos que impidan la visibilidad. • Requiere una altura entre 15 y 20 metros para una detección de presencia y velocidad óptimos. • Se ve afectada por vibraciones debido al viento. • Coste computacional y de procesamiento de datos. • Sistema controvertido por la opinión pública, muy suceptible a actos vandálicos.

Sin embargo no todo son beneficios en los sitemas de monitorización de vídeo, ya se encuentran con algunas limitaciones que están frenando su expansión, sobre todo para la gestión de tráfico urbano. Sus costes de implantación y mantimiento son bastante elevados [180], así como sus costes computacionales [302] en comparación con sistemas más simples como los presentados anteriormente. Se ven muy afectados por las condiciones meteorológicas que afectan a la visibilidad o que pueden interferir en la pulcritud de las lentes. Además, la presencia de cámaras crea mucha suceptibilidad en la opinión publica, por lo que suelen ser reguladas de forma más estricta y restrictiva que otros sistemas de monitorización.

[180] *Kilómetros a precio de Oro*

[302] *Road obstacle detection and tracking by an active and intelligent sensing strategy*

Es por ello que su uso se encuentra limitado a la monitorización del tráfico en vías de alta capacidad, en la que un número elevado de carriles puede ser gestionado con una cámara emplazada. En la Tabla 3.3 se recogen las principales fortalezas y debilidades que presentan los sistema de reconocimiento de imágenes de vídeo para la monitorización de vehículos.

3.5 SISTEMAS DE MONITORIZACIÓN DE PERSONAS

La monitorización de personas ha cobrado una gran importancia en los últimos años debido al entorno de las Smartcities (Sección 2.2). El principal problema al que se enfrentan las ciudades del futuro está originado por el aumento de la población viviendo y moviéndose por las ciudades. Proveer de fuentes de datos a las ciudades del futuro sobre la movilidad asociada a las personas, permite obtener información sobre los desplazamientos de las mismas, y en base a esta información generar conocimiento sobre su comportamiento, que sirva para emprender estrategias más eficientes para su gestión.

Este tipo de conocimiento, la monitorización de personas ha estado más asociado a los entornos ecónomicos y de márketing, por ejemplo, obteniendo ratios de visitantes que realizan compras en una tienda [259]. Sin embargo los beneficios de realizar una monitorización de personas en una `Smartcity` van mucho más allá.

[259] Fundamentals of marketing

Obtener información sobre cuanta gente visita un lugar público, en que horarios preferentes, con que frecuencia, cuanto tiempo tienen que esperar o cuantos se van sin llegar a ser atentidos puede servir para optimizar y mejorar la eficiencia de la planicación de los servicios provistos. Por ejemplo en un museo, una oficina administrativa, un centro deportivo o clínico pueden obtener información real sobre como interaccionan las personas con dichos servicios. Lo cual proveería los medios para mejorar la eficiencia de dichos servicios.

Las tecnologías para el conteo y monitorización personas en sus orígenes han aparecido por evolución de los sistemas para monitorizar vehículos [240]. Sin embargo las personas disfrutan de un libre albedrío al contrario que los vehículos que se ven obligado a circular por unas vías determinadas y en un sentido concreto. Esto hace que la detección y monitorización de personas sea un problema mucho más complejo.

[240] Case study analysis of pedestrian and bicycle data collection in US communities

Sin embargo las métricas y medidas aplicadas a la monitorización de vehículos son fácilmente aplicables a la monitorización de personas. Cómo se ha señalado anteriormente, la fuente de datos resulta irrelevante si es capaz de proporcionar las magnitudes que se precisan para la obtención de métricas y aplicación de procedimiento. Y de igual manera, resulta irrelevante la naturaleza de los entes a ser monitorizados, ya sean vehículos, personas o dispositivos inteligentes [105].

[105] Traffic Control Systems Handbook

A continuación se recogen algunos de los sistemas empleados para la monitorización de personas.

3.5.1 *Sensores basados en haces de rayos infrarrojos*

Las primeras aproximaciones de sensores dedicados al conteo de personas se basan en rayos infrarrojos que atravesaban un punto de paso cerrado y unipersonal. Al cortase el rayo infrarrojo se contabiliza que una persona ha atravesado el punto de paso [117] (Figura 3.13).

[117] People-counting system using multisensing application

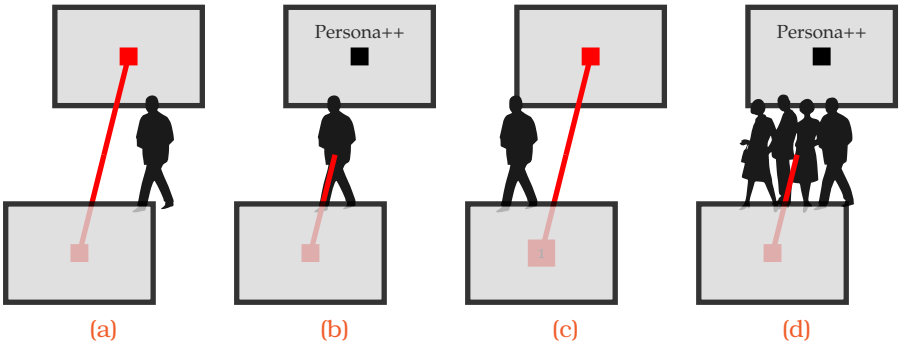


Figura 3.13
Funcionamiento de un haz infrarrojo para detección de personas en un punto de paso. En (a) la persona se aproxima al haz infrarrojo. En (b) en el instante justo que la persona se interpone en el haz infrarrojo, lo cual corta el haz infrarrojo y se registra como el paso de una persona. En (c) la persona que bloqueaba el haz ha pasado, por lo que este vuelve a cerrar el haz infrarrojo, cerrándose el circuito. En (d) un grupo de personas caminan juntas, por lo que el infrarrojo es interrumpido mientras los cuatro están pasando. Esta es una de las mayores limitaciones del empleo de haz infrarrojos para la detección de personas.

A pesar de sus ventajas como un precio económico, su principal impedimento de implantación masiva es que tiene que limitar el punto de paso a una única persona, pues si dos o más personas caminan juntas, el infrarrojo es interrumpido constantemente durante el tránsito de todas las personas, contabilizándose únicamente un paso. En la Tabla 3.4 se recogen las principales fortalezas y debilidades de este sistema de monitorización.

Tabla 3.4
Sistemas de rayos infrarrojos: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Bajo coste.	• Limitados a zonas de tránsito unipersonal. • La altura de su emplazamiento resulta conflictiva. • Dificultad para distinguir entre personas que caminan en grupos. • No es posibile realizar el seguimiento de los movimientos de las personas.

Además, su instalación tiene un factor de elección crítico, que es la altura a la que el sensor tiene que colocarse. De emplazarse muy bajo, se detectarán las piernas durante el paso, pudiendo detectar dos o más zancadas de un único caminante como varias personas distintas. Colocado a una altura superior, los brazos pueden interferir en el haz infrarrojo, activando nuevamente el

contendo de una persona cuando lo que se ha detectado es un brazo, y el tronco de la persona volverá a ser detectado y considerado como una persona distinta. Situado a una altura para enfocar los hombros o la cabeza, supone de descartar personas con distintas estaturas, o enfrentarse con personas muy altas al mismo problema anterior con el conteo de los brazos.

3.5.2 *Sensores basados mediciones de calor*

Los sensores termales deben sus orígenes a aplicaciones militares. Se basan en cámaras termales que detectan la cantidad de emisiones térmicas reflejadas por los objetos del escenario enfocado [68]. Dado que las cualidades termales de las personas son completamente distintas a las radiacciones del escenario (o más bajas o más altas), el procesamiento de la imagen térmica captada permite identificar el número de personas que se encuentran enfocadas [116] (Figura 3.14).

[68] Robust Background-Subtraction for Person Detection in Thermal Imagery.

[116] People count system using multi-sensing application

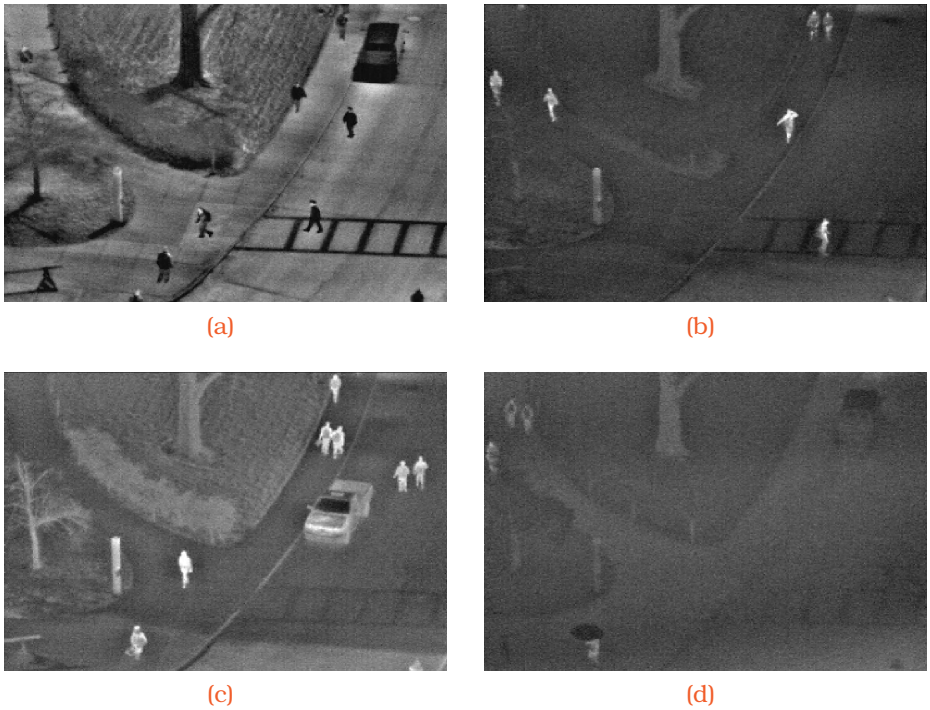


Figura 3.14
Imágenes captadas por una cámara térmica en distintos instante del día, con distinta condiciones térmicas del escenario. El blanco representa emisión térmica que el color negro.
Fuente: James W. Davis et al. [67]

Debido a que la temperatura corporal es estable ($36,6° \pm 0.5°$), resulta distinguible en la mayoría de escenarios. Sin embargo, existen muchos factores que afectan a la emisión térmica del escenario, como la temperatura y el momento del día y la estación, la pigmentación de los elementos, la exposición a fuentes de calor o refrigeración, la incidencia de sombras, etc. Por ejemplo en la Figura 3.14(a) las personas emiten menos calor que el escenario, por lo que identificarlas supone localizar las áreas con menos emisiones termales. Sin

embargo, en el resto de escenarios, las personas se encuentran más calientes que el entorno, por lo que su búsqueda se centra en las áreas con mayor emisión térmica. Si consideramos que a lo largo del tiempo ambas mediciones son continuas (no se producen saltos abruptos) el teorema de Bolzano nos indica que existen periodos de tiempo en el que el escenario y las personas se encuentran a la misma temperatura. Hecho que se puede observar en la Figura 3.14(d). Esto provoca que haya instantes de tiempo en los que el sistema de monitorización es ciego, es decir, es incapaz de distinguir entre el entorno y las personas.

De igual forma estos sistemas no están exentos de fallos (Figura 3.15). Detectando un vehículo como una persona (Figura 3.15(b)), o a dos personas caminando juntas como una única persona (Figura 3.15(c)). Las imágenes termales, al tener que basarse en el contraste con un escenario, no resultan viables para grandes concentraciones de personas.

(a) (b) (c)

Figura 3.15
Imágenes procesadas por un sistema de monitorización térmico en distintos instante del día, con distinta condiciones térmicas del escenario. El blanco representa emisión térmica que el color negro
Fuente: James W. Davis et al. [67]

Otros problemas de los sistemas termales es la existencia de escenarios con puntos ciegos, la dificultad para distinguir entre personas y otros objetos con similares emisiones térmicas y su incapacidad operativa para funcionar con grandes masas de personas. Además, resulta imposible reconocer a la misma persona en base a su emisión térmica, lo que imposibilita la monitorización entre cámaras de una misma persona. La Tabla 3.5 recoge las principales fortaleza y debilidades de este sistema.

Tabla 3.5
Sistemas basados en mediciones termales: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Pueden distinguir personas independientemente de las condiciones lúmínicas. • Pueden trazar movimientos en el plano de la cámara. • Puede distinguir si hay una persona en zonas de acceso restringido.	• Existencias de periodos ciegos. • Las condiciones externas que influyen en la temperatura influyen en el sistema. • Imposibilidad de seguimiento entre distintas cámaras. • Dificultades para el seguimiento de grupo de personas.

Debido a las limitaciones que presenta, los sistemas termales han quedado excluidos de su implantación en las ciudades para el control masivo de personas, sin embargo, resultan muy empleados en las zonas de control restringido al tránsito, porque en estos escenarios se les puede extraer todos sus beneficios. Es decir, estos sistemas operan como sistemas de reconcimiento de presencia en zonas restringidas, como pueden ser edificios cerrados o zonas no abiertas al público de edificios concretos. Además su empleo no está exento de polémica, al considerar que vulneran la privacidad de las personas [195].

[195] The use of Thermal Imaging and the Fourth Amendment

3.5.3 *Sistemas de monitorización basados en cámaras de vigilancia*

De igual manera que con la captación de vehículos, se han empleado cámaras de vídeo para la monitorización de personas [48, 241]. Sin embargo la libertad de movimiento de las personas vuelve a perjudicar a la eficacia y aplicabilidad del sistema.

[48, 241] Towards a Robust Solution to People Counting, A system for counting people in video images using neural networks to identify the background scene

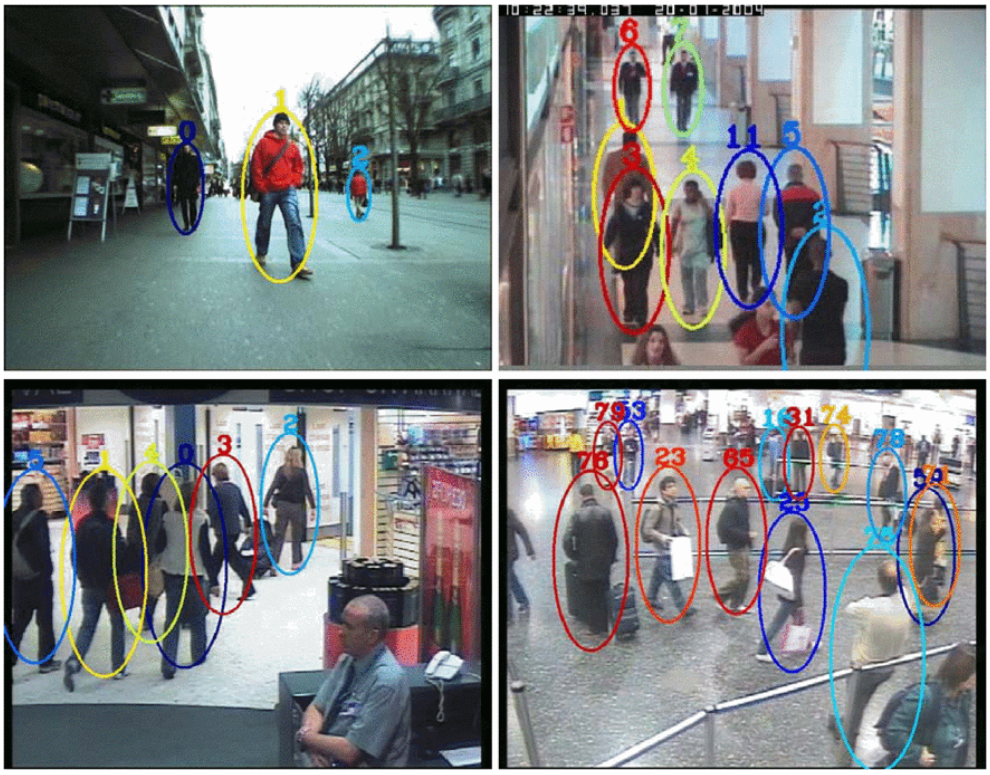


Figura 3.16
Monitorización de personas por medio de imágenes de vídeo. Se utilizan planos bajos y picados para limitar el número máximo de personas enfocadas.
Fuente: Cheng-Hao Kuo et al. [155]

Al contrario que los vehículos que circulan por calzadas definidas y encauzadas en vías, las personas se distribuyen por toda la superficie, presentan movimientos en apariencia erráticos pero periódicos, conservando distancia entre ellos salvo en situaciones de paradas [103]. Esto hace computacionalmente más difícil monitorizar personas que monitorizar vehículos. La primera

[103] Understanding individual human mobility patterns

consecuencia de esto es que al contrario de la cámaras de monitorización de vehículos que son emplazadas en puntos elevados y con planos abiertos, las cámaras de monitorización de personas se sitúan a solo un par de metros del suelo, con planos picados o cerrados, en los que se limita el tamaño del plano captado, limitando el número de personas que pueden ser detectadas como máximo al mismo tiempo (Figura 3.16).

La posibilidad de control de movimiento o `tracking` se encuentra limitada debido a que las personas no portan una matrícula que sea fácilmente reconocible. Sin embargo, sí existen aproximaciones que permiten seguir el movimiento de una persona en base a su aspecto actual. Esto permite por ejemplo seguir a una persona a lo largo de un edificio, siempre y cuando no altere su aspecto físico (Figura 3.17).Sin embargo, este problema de clasificación es computacionalmente muy complejo para sistemas en los que se detecten miles o millones de personas [155].

[155] How does person identity recognition help multi-person tracking?

Figura 3.17
Seguimiento o tracking de una persona por medio de imágenes de vídeo, buscando similitudes en el aspecto físico de las personas localizadas.
Fuente: Cheng-Hao Kuo et al. [155]

Si bien el aspecto y vestimenta de las personas puede variar a lo largo de los días, lo cual dificulta su seguimiento a lo largo del tiempo, la cara permanece inalterable a corto y medio plazo. De igual manera que para detectar las matrículas de los vehículos se hace necesario disponer de cámaras que ofrezcan imágenes de alta calidad, las cámaras de monitorización de personas pueden usar esta mayor resolución para intentar reconocer el aspecto de las caras. El reconocimiento facial está en auge, si bien el reconocimiento de grandes masas de personas está aún dentro de sus limitaciones [314].

[314] Face recognition: A literature survey

Es por ello que la monitorización de personas por medio de imágenes de vídeo, ha quedado relegada a la vigilancia de zonas concretas y acotadas y la generación de imágenes históricas para ser consultadas en caso de existir contratiempos o alteraciones que requieran su verificación. El reconocimiento facial, de igual manera, ha quedado relegado a los casos de investigaciones

criminales donde se busca un sujeto concreto, más que a la identificación de multitudes. Estos sistemas, al igual que los termales, resultan muy útiles para la vigilancia de zonas cortadas al paso público o en zonas de vigilancia por ser susceptibles de sufrir atentados a su integridad, como pueden ser los cajeros automáticos, los elementos arquitectónicos públicos (fuentes, jardines, estatuas) o de zonas con gran afluencia de delitos. Además, su implantación debido a la aparente vulneración de la libertad de la persona está fuertemente regulada [107] lo que limita su implantación masiva.

[107] The legal regulation of CCTV in Europe

Tabla 3.6
Sistemas en captación de imágenes de personas: Fortalezas y Debilidades

FORTALEZAS	DEBILIDADES
• Reconocimiento de personas en planos. • Pueden trazar movimientos en el plano de la cámara y entre cámaras en periodos de tiempo cercanos. • Sistema muy eficiente en zonas de paso único, como puntos de desembarco o entradas de edificios. • Genera gran cantidad de datos.	• Instalación y mantenimiento muy costoso. • Alto coste computacional en el procesamiento de los datos. • Sistema muy controvertido por la opinión pública. • Dificultad para la gestión de multitudes de personas. • Las condiciones externas que influyen en la visibilidad o la pulcritud de la lente influyen negativamente en el sistema. • Imposibilidad de seguimiento entre distintas cámaras. • Dificultades para el siguiemiento de grupo de personas. • El reconocimiento facil de multitudes de personas no es asumible actualmente.

3.6 SISTEMAS DE MONITORIZACIÓN BASADOS EN CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS

Cómo se ha presentado anteriormente, los sistemas de detección de vehículos tienen altos costes asociados, debido a que se implantan sobre la vía. Las alteraciones en la vía no son deseables, ya que para nuevas tareas de implantación y mantenimiento, requieren cortar el acceso a ellas. Además, se ha presentado como en las ciudades europeas las modificaciones en las infraestructuras son difícilmente abordables. Por último, estos sistemas resultan limitados para el seguimiento masivo de vehículos, debido a que los sistemas que lo permiten resultan excesivamente caros y complejos computacionalmente por la alta calidad de imagen que se requeriría.

Por otro lado, los sistemas presentados para la monitorización de personas resultan insuficientes para la monitorización de las masas que se mueven por las ciudades actuales. Todos ellos funcionan muy bien en la detección de personas en entornos cerrados como edificios o zonas concretas, pero se encuentran limitados debido a la naturaleza de los movimientos de la personas en zonas amplias o ante multitudes. Además, no permiten un seguimiento de los movimientos de las personas, por lo que no resultan útiles para este desempeño.

De igual manera, se ha presentado que las ciudades futuro necesitan de un fuente de datos que les provea de información sobre los movimientos que realizan las personas y vehículos para poder gestionar sus servicios e infraestructuras de forma más eficiente. Se hace necesario proveer de una fuente de datos que satisfaga estos requerimientos.

Partiendo del funcionamiento de los sistemas de monitorización de personas, se puede observar como todos ellos se basan en principios muy similares. La detección y estudio de las alteraciones en el espectro electromagnético. Ya sea interrupciones en haces infrarrojos, sus emisiones térmicas o su refracciones del espectro luminosos. Sin embargo estás tres aproximaciones han resultado ser insuficientes.

Tal y como se ha presentado en la Sección 2.1, los dispositivos inteligentes o `smartdevices` se han popularizado y extendido entre prácticamente toda la población. Si bien no podemos detectar a las propias personas, si se puede monitorizar los dispositivos inteligentes que portan. Una de las características que se han presentado de los dispositivos inteligentes es que se comunican con dispositivos de igual o diferente naturaleza. Y estas comunicaciones se realizan de forma inalámbrica, empleando el espectro radioelectrico como canal de comunicaciones. Y debido a que son dispositivos influenciados por el Internet de las Cosas, realizan estas comunicaciones sin inmediación directa de sus portadores.

Debido a que estas comuniciones se producen de forma inadvertida para sus dueños, se pueden medir las alteraciones en el espacio radioelectrico que provocan dichas comunicaciones. E incluso, se pueden realizar pequeños

diálogos con estos dispositivos con el fin de conocer más sobre ellos. Si bien un sistema de monitorización que se base en monitorizar los dispositivos inteligentes no estaría muestreando personas directamente, se sabe que 83 % de las personas usa un smartphone. De esta manera se trata un sistema de monitorización de personas no exahustivo[3] que si bien tiene un ratio de conversión bastante directo y además, presenta un tendencia a ser absoluto en el futuro según la tendencia de adopción de los jóvenes (Figura 2.5). Además, dado el carácter personal e intransferible que tienen estos de dispositivos, se puede extrapolar a la detección de la persona en concreto que lo porta en todos los casos que el dispositivo sea detectado.

Por último, estos dispositivos inteligentes suelen disponer de medios con lo que conocer la ubicación en la que se encuentra el dispositivo. Si bien puede parecer el mismo problema, conocer la ubicación de un dispositivo y saber cuantos dispositivos se encuentran en dicha ubicación son problemas totalmente distintos.

3.6.1 *El posicionamiento por medio de comunicaciones inalámbricas*

Las comunicaciones que realizan los dispositivos suelen ser empleadas para que el dispositivo obtenga información, más que para proporcionarla. Así el dispositivo suele emplear sus comunicaciones para ver si ha recibido un correo electrónico, algún mensaje instantáneo, el tiempo atmosférico o su ubicación. Sin embargo este hecho en sí no resulta útil para la monitorización, a no ser, que se interfiera en la lógica interna del propio dispositivo[4].

Independientemente de la tecnología de comunicación inalámbrica que emplee el dispotivio para conocer su ubicación, todas ellas se basan en la recepción de información, no en la emisión. Esto es, en recibir una serie de señales determinadas a una intensidad concreta, realizar una serie de opeaciones con dicha información e inferir como resultado dónde se encuentra el dispositivo.

En la Tabla 3.7 se presentan las diferentes tecnologías de comunicación que emplean los dispositivos inteligentes para su posicionamiento. En dicha tabla figuran las siguientes columnas:

DATOS Indica la fuente de datos resultante después de que el dispositivo realice los cálculos necesarios. Dicho dato puede ser consultado en una base de datos tanto local como online para conseguir una posición más concreta.

REFERENCIA Indica si es una localización absoluta (p.e. unas coordenadas geográficas exactas) o relativas a una localización (p.e. a una distancia concreta de un punto de refencia)

3 ↑Según la clasificación presentada al principio de esta sección.

4 ↑Cosa que si pueden hacer Google, Microsoft o cualquier empresa que consiga que el usuario instale una App en su dispositivo sin decirle que esta App le está monitorizando.

Tabla 3.7
Comparativa de las tecnologías de comunicación inalámbrica que emplean los dispositivos inteligentes para conocer su posición.

TECNOLOGÍA	DATOS	REFERENCIA	EXPRESIÓN	PRECISIÓN	COBERTURA
GPS [178]	Coordenadas Geográficas	Absoluta	Física	1-5 metros (95 %-99 %)	Exteriores
WiFi [253]	ID Punto de Acceso + Intensidad de señal	Relativo	Simbólico	1-20 metros	< 100 metos desde el punto de acceso
	Con Triangulación: Coordenadas Aproximadas	Absolutas	Físico		
Torre telefonía	ID Torre + Intensidad de señal	Relativa	Simbólica	50-200 metros en ciudad	Cobertura telefónica. 5-30km desde la torre
	Con Triangulación: Coordenadas Aproximadas	Absoluta	Física		
Bluetooth [255]	Identificador de Dispositivo	Relativa	Simbólica	Rango del sensor Bluetooth	5-10 metros para Clase 2. 20-30 metros para clase 1.
RFID [192]	ID del Lector	Relativa / Absoluta	Simbólico / Físico	Rango del sensor RFID	<1 metro en RFID pasivo. <100 metros en RFID activo

EXPRESIÓN Indica si el dato obtenido por el sistema de posicionamiento proporciona una posición geográfica absoluta o una etiqueta simbólica[5].

PRECISIÓN Indica la precisión o margen de error de la posición determinada por el sistema de ubicación.

COBERTURA Indica el rango de cobertura teoríco máximo donde la comunicación entre el emisor y el dispositivo inteligente puede ser realizada.

Haciendo uso de las emisiones que recibe el dispositivo inteligente, este es capaz de determinar donde se encuentra, lo que resuelve el problema del posicionamiento. El problema de la monitorización, es similar, salvo que se intercambian los papeles de emisor y receptor. Es decir, no se desea responder donde está un dispositivo, sino cuantos dispositivos hay en las inmediaciones de un punto determinado.

3.6.2 *La captación de comunicaciones inalámbricas como medio de monitorización*

Cómo se ha presentado anteriormente, los dispositivos emplean sus comunicaciones inalámbricas para comunicarse y obtener la información. Las comunicaciones que suelen usar los dispositivos inteligentes más frecuentemente son las presentadas en la Tabla 3.7: GPS, WiFI, Telefonía Móvil[6], Bluetooth (o BT), NFC y RFID.

Sin entrar en detalles que serán descritos en el capítulo 4, la monitorización por medio de las captación de comunicaciones inalámbricas consiste en indentificar un dispositivo en las inmediaciones por medio de las comuniciones que emite.

Existen autores que ya han empleado la misma filosofía para resolver el problema de la monitorización [295] de personas y vehículos, aunque centrándolo más en la detección dentro de edificios y de forma aislada.

[295] Bluetooth based collaborative crowd density estimation with mobile phones

5 ↑Como por ejemplo «Dentro Edificio Baxter».
6 ↑Más adelante se concretará en la defición de Telefonía Móvil en el Capítulo 4.6

Así por ejemplo la idea de usar el BT de los dispositivos personales para la monitorización no es nueva, y ha sido estudiada por multitud de investigadores a lo largo de los últimos años. Tom Nicolai et al.[193] usan detectores estáticos BT para estudiar el movimiento de la gente; Morrison et al.[188] usa el mismo principio para el seguimiento de la gente en la asistencia de actividades en un estadio; Kostakos et al. [153] para el estudio y seguimiento de las personas en líneas de trasportes públicos de autobuses. Usando detectores portables de comunicaciones BT, Weppner et al.[295] estiman la densidad de personas en un estadio de futbol durante un campeonato. El uso de este tipo de sistemas de captación es una tendencia que está en alza, y han empezado surgir los primeros artículos basados en el fundamento más que en la aplicación práctica. Por ejemplo Nicolai et al.[193] han estudidado la relación entre el número de personas en un entorno urbano y él número de dispositivos BT en estado de descubrimiento.

[193] About the relationship between people and discoverable Bluetooth devices in urban environments

[188] Visualisation of spectator activity at stadium events

[153] Using Bluetooth to capture passenger trips on public transport buses

[295] Bluetooth based collaborative crowd density estimation with mobile phones

[193] About the relationship between people and discoverable Bluetooth devices in urban environments

La de idea de usar las comunicaciones WiFi también ha sido explotada recientemente. Durante el desarrollo de esta tesis, Wei Xi et al.[300] ha usado las comunicaciones Wifi para demostrar que pueden ser empleadas para contar el número de personas con gran precisión. O por ejemplo con Gabe Fierro et al.[84] que han empleado las comunicaciones WiFI para estimar el nivel de ocupación de un edificio sin necesitar infraestructura adicional. Esta información puede ser util para la evacuación de grandes edificios [7] o para la estimación de la asistencia a una sala concreta [258].

[300] Electronic frog eye: Counting crowd using WiFi

[84] Zone-level Occupancy Counting with Existing Infrastructure

[7] SmartEvacTrak: A people counting and coarse-level localization solution for efficient evacuation of large buildings

[258] Estimating the number of people in a particular area using WiFi

Otros autores han empleado vías de comunicación inalámbrica no directamente empleadas por los dispositivos, como por ejemplo la emisión de ultrasonidos en anuncios de televisión y radio detecables por los dispositivos inteligentes [12]. O por el ruido de las antenas en las bandas de frecuencias no empleadas [301].

[12] Privacy Threats through Ultrasonic Side Channels on Mobile Devices

[301] Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances

Sin embargo, a pesar de que hay autores que ya han empleado la captación de las comunicaciones inalámbricas para la resolución de problemas concretos, el estudio de la aplicación de esta fuente de datos para servir a la constitución de una `Smartcity` no ha sido estudiado aún en profundidad. De igual manera su aplicación se ha centrado en aspectos del marketing más que la estudio de la movilidad de personas.

3.6.3 *Alternativas comerciales*

Debido a que las `Smartcities` están siendo fuertemente impulsadas por las empresas del sector de las TICS, han surgido empresas que ofrecen soluciones para la monitorización de personas y vehículos haciendo uso de la captación de comunicaciones inalámbricas.

Sin embargo estos sistemas resultan opacos debido al celo empresarial a la difusión de sus avances y principios fundamentales, por lo que resulta imposible conocer su funcionamiento, así como a que estudios y valoraciones han sido sometidos antes de ser lanzado como producto. Además, debido a que son productos comerciales, existe una falta de actitud crítica ante

las limitaciones del sistema, ya que normalmente se enmascaran para no perjudicar a la venta del producto.

Por último, en la mayoría de ocasiones se ofertan dentro de un pack de `Smartcity` que ofrece una infraestructura centralizada y opaca para las administraciones, así como un portal propio de la empresa ofertante donde acceder a los datos.

A continuación se recogen alguna de estas soluciones que han sido desarrolladas:

BIT CARRIER [87, 282]: Ofrece un control del tráfico basado en conteo de dispositivos BT y Wifi, contando personas y rutas. Según su web, actualmente disponen en Cataluña de más de 150 sensores desplegados, capaces de monitorizar 200000 personas cada día.

[87, 282] System and method for monitoring people and/or vehicles in urban environments, BitCarrier. Go With the Flow

TRAFFICNOW [267]: Ofrece controles del tráfico centrados en pequeñas zonas urbanas mediante BT y WiFi. Actualmente consta con unos 500 sensores desplegados en las carreteras de París, Estambul, Wisconsin y Vigo.

[267] Trafficnow

TRAFFAX INC [45]: Ofrece una solución para el cálculo de tiempos de rutas entre instalaciones basado en la implantación de sensores BT en la misma y tokens emisores en los vehículos a ser monitorazos.

[45] TraffaxInc

SAVARI NETWORKS [238]: Ofrece una solución para `Smartcities` que incluye el producto StreetWAVE para la montorización del tráfico en tiempo real.

[238] SavariNetworks

TRAFFICCAST [268]: Han desarrollado modelos predictivos del tráfico en diferentes ciudades basándose en datos obtenidos por diferentes tecnologías como imágenes de vídeo, BT y etiquetas RFID emplazadas en los vehículos de la administración.

[268] TrafficCast

3.7 RESUMEN

En esta sección se han presentado los principios, magnitudes y medidas de efectividad ampliamente empleados en la gestión del tráfico de vehículos, que han sido presentados debido a su aplicabilidad a la fuente de datos propuesta. La mayoría de los sistemas y algoritmos existentes para la gestión del tráfico hace uso de estas magnitudes, por lo que la naturaleza de la fuente de datos es irrelevante si es capaz de proporcionar las mismas métricas que han sido proporcionadas por otros sistemas. Debido a esta independencia, dichas magnitudes y métricas pueden ser aplicadas a los estudios de movilidad independientemente de la naturaleza de los datos.

A su vez, se han presentado y discutido los tres sistemas de monitorización más frecuentemente empleados para la gestión del tráfico, que serán de vital importancia para la validación de la fuente de datos propuesta, pues serán comparados cara a cara en algunos escenarios.

Se han presentado las tecnologías que han surgido para la monitorización de personas y se han analizando las tres más importantes. Para estos sistemas se han resumido los problemas que presentan para la monitorización de masas de personas, requisito necesario para la implantación masiva del sistema para una `Smartcity`.

A continuación se han presentado los sistemas de monitorización de personas emergentes, basados en la captura de la comunicaciones inalámbricas. Se han presentado brevemente las tecnologías que se utilizan en las comunicaciones entre dispostivos[7], así como los estudios que han surgido en los últimos años que emplean la captación de dichas comunicaciones para la resolución de problemas concretos asociados con la monitorización de personas. Sobre todo problemas asociados a entornos de marketing o asistencia a edificios culturales.

Por último, se han presentado las soluciones comerciales existentes en paquetes de `Smartcities`, que se basan en los principios de la captación inalámbrica para ofrecer fuentes de datos a las ciudades y sus administraciones. Sin embargo, dichas soluciones comerciales son opacas para los investigadores y forman parte de packs ofertados a las instituciones como soluciones completas, por lo que su estudio y contrastación resultan inviables para la mayoría de los investigadores.

Esta falta de transparencia en los sistemas emergentes abre las puertas de la elaboración de esta tesis doctoral centrada en el estudio de la viabilidad de estos sistemas para el entorno de las ciudades inteligentes.

7 ↑Que será nuevamente abordadas en profundidad en el capítulo X

Parte II

Marco teórico y Metodología

4

FUNDAMENTOS TEÓRICOS Y TECNOLÓGICOS DE LA CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS

Todo es mejor con Bluetooth.

— Ph.D Sheldon Lee Cooper.

Este capítulo presenta los fundamentos tecnológicos que sustentan las comunicaciones inalámbricas, cuya captación nutre la fuente de datos para la monitorización de personas y vehículos propuesta en esta tesis. Estos fundamentos se centran en las capacidades que presentan dichas tecnologías para la detección de dispositivos en las inmediaciones y en la información que puede extraerse o conocerse de dichos dispositivos detectados. Es deseable que los dispositivos sean capaces de ser identificados de forma única, con el fin de poder trazar sus movimientos en base a detecciones sucesivas en distintos puntos geográficos.

Se presentarán cinco tecnologías principales, aunque solamente dos de ellas han sido empleadas en la propuesta de esta tesis: Bluetooth BR/EPR y WiFi ya que han resultado ser las más adecuadas. Sin embargo, se presentarán brevemente las tecnologías empleadas en las comunicaciones telefónicas y en las comunicaciones NFC y RFID indicando los aspectos[1] que imposibilitan su empleo para la monitorización de dispositivos propuesta en esta tesis.

Índice del capítulo

1 ↑Tanto tecnológicos como legales.

4.1 INTRODUCCIÓN A LAS COMUNICACIONES INALÁMBRICAS

Las comunicaciones inalámbricas se caracterizan por emplear la modulación de ondas electromagnéticas a través del espacio para transmitir información entre un emisor y un receptor, sin necesidad de emplear un medio de propagación físico [100] acotado. En general, las comunicaciones inalámbricas emplean ondas de radiofrecuencia de baja potencia acotadas a una banda de frecuencia específica (Figura 4.1). Esta banda de frecuencia, puede ser de uso libre o privado. La emisión, e incluso recepción, en bandas de frecuencia privadas puede estar limitada a autoridades con licencia de uso, incurriéndose en un delito en caso de emitir (o recibir) en dichas frecuencias sin la correspondiente licencia.

[100] Wireless communications

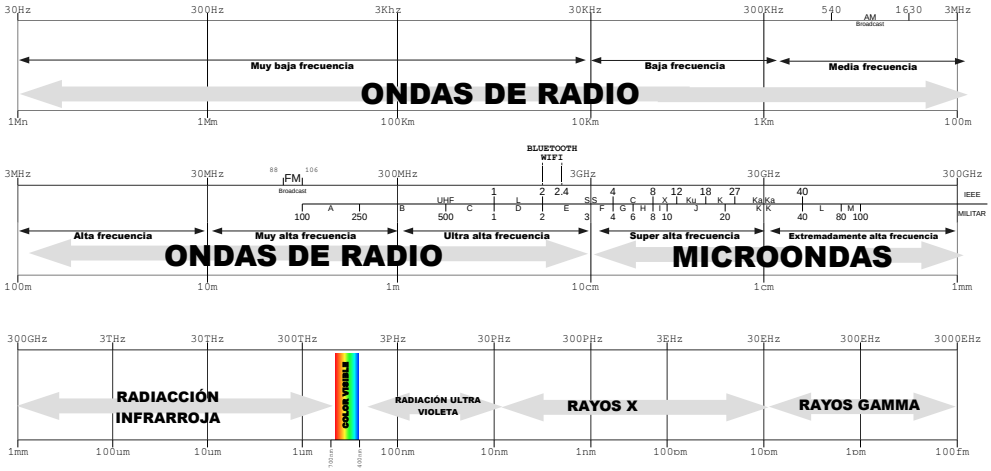


Figura 4.1
Bandas de frecuencias en las que se divide el espectro electromagnético.

[254] 521-2002 - IEEE Standard Letter Designations for Radar-Frequency Bands

El IEEE en su estándar IEEE 521 [254] divide las frecuencias del espectro radioeléctrico en diversas bandas. Estas bandas son empleadas para diferenciar entre diferentes servicios que emplean radiofrecuencia para sus comunicaciones, como la televisión, la radio, el radar, la telefonía móvil o la transmisión de datos. La misma banda de frecuencia puede ser empleada por múltiples protocolos de comunicación coexistiendo en el mismo espacio radioeléctrico. Merece especial atención la banda ISM[2] definida para uso público y no comercial, que ha acogido las comunicaciones WLAN y WPAN que se han popularizado en los últimos años.

Los dispositivos inteligentes presentados en el Capítulo 2 emplean la banda de frecuencia ISM para sus comunicaciones WWAN y WPAN, haciendo uso de bandas comerciales o licenciadas para las comunicaciones con antenas de telefonía para transmisión de voz. Sin tener en cuenta las tecnologías empleadas para las comunicaciones telefónicas y de datos (que se presentarán en la Sección 4.6), han proliferado las comunicaciones Bluetooth (Sección 4.2), WiFi (Sección 4.3), NFC (Sección 4.4) y RFID (Sección 4.5).

2 ↑Industrial, Scientifical and Medical

4.2 BLUETOOTH

Bluetooth es una tecnología inalámbrica empleada en comunicaciones de corto alcance WPAN (o alcance personal) que busca reemplazar los cables que unen los dispositivos electrónicos personales y que se encuentra definida dentro del estándar IEEE-802.15.1-2002 [255]. Las principales características de este protocolo son su robustez, su bajo consumo energético y económico. Además, la gran mayoría de los componentes del protocolo son opcionales, lo que permite una gran diversidad y flexibilidad de uso. Esto ha originado que sea empleado en una gran variedad de dispositivos.

[255] 802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems

Actualmente, aunque se encuentran recogidos dentro del mismo protocolo, existen dos sistemas de transmisión inalámbrica Bluetooth: `Basic Rate (BR)` y `Low Energy (LE)`. Ambos sistemas incluyen mecanismos para el descubrimiento de dispositivos, el establecimiento de conexiones y la realización de comunicaciones.

Los sistemas `BR` pueden incluir opcionalmente las extensiones `Enhanced Data Rate (EDR)` y `Alternate Media Access Control (MAC)/Physical (PHY)` o `Alternate MAC/PHY (AMP)`. `BR` ofrece comunicación síncrona y asíncrona a $721.2kb/s$, aumentada a $2.1Mb/s$ con `(EdR)` y a $54Mb/s$ con `(AMP)`. Estas mejoras le permite operar en escenarios donde la tasa de envío es elevada.

Los sistemas `LE` incluyen características de diseño para reducir el consumo energético, así como menor complejidad y coste económico. Estas características limitan al sistemas de comunicación por lo que son empleadas en entornos donde el volumen de datos de transmisión es menor o existen ciclos de comunicación cortos.

A pesar de que Bluetooth `BR` y `LE` son dos sistemas de transmisión distintos, los dispositivos pueden implementar ambos sistemas de transmisión dentro del mismo dispositivo Bluetooth. Esto se debe a que el núcleo de la tecnología Bluetooth está compuesto por un `Host` y uno o más `Controllers`. El `Host` es una entidad lógica definida como todas las capas de protocolo debajo de los perfiles no principales y por encima del `Host Controller Interface (HCI)`. El `Controller` se define como todas las capas de protocolo que hay debajo del `HCI` (Figura 4.2).

Una implementación Bluetooth tiene únicamente un `Primary Controller`, y se puede presentar en diferentes configuraciones:

- Un `controller BR/EDR` incluyendo `Radio`, `Baseband`, `Link Manager` y opcionalmente `HCI`
- Un `controller LE` incluyendo `LE PHY`, `Link Layer` y opcionalmente `HCI`.
- Una combinación de `controller BR/EDR` y `controller LE` en un único controlador.

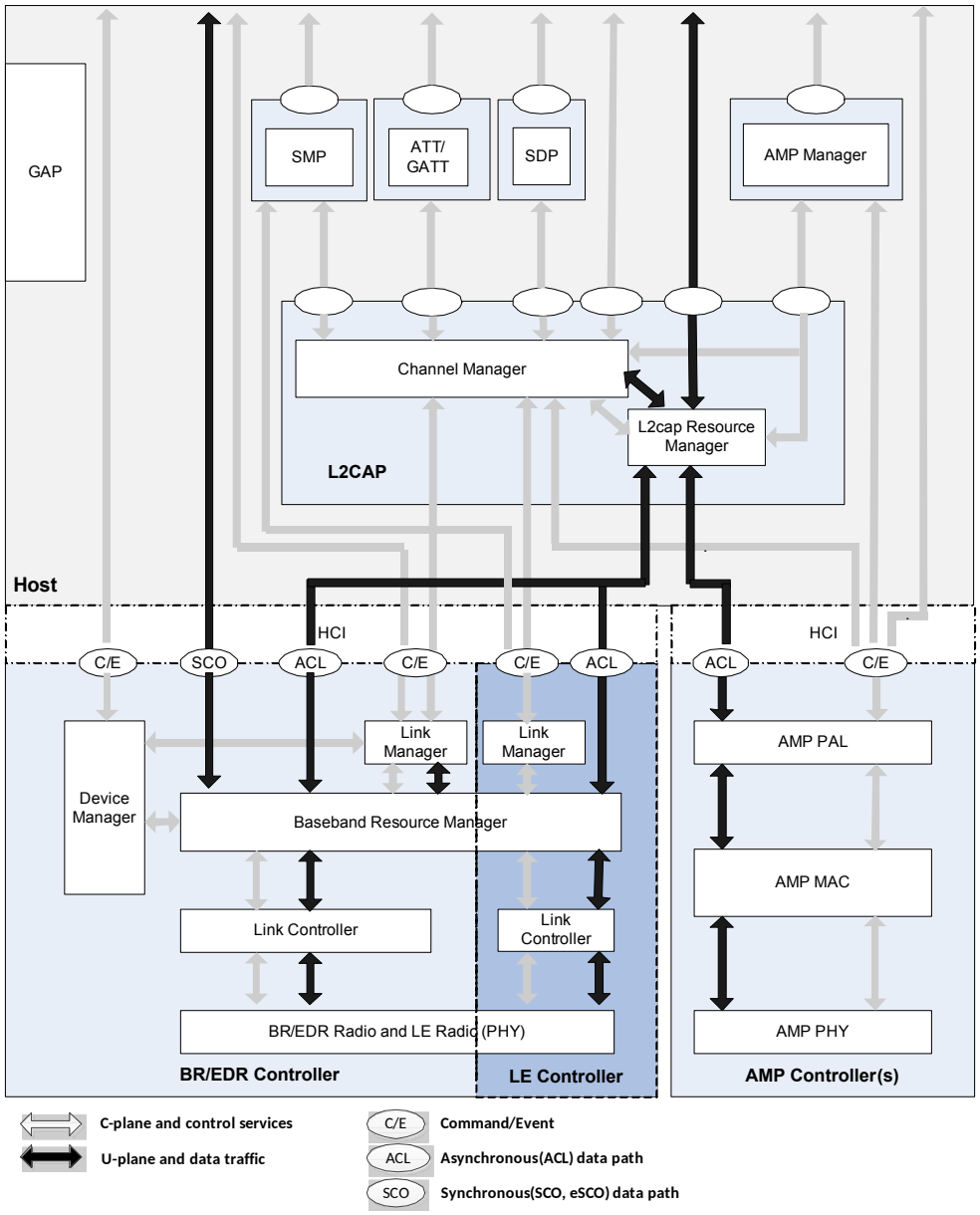


Figura 4.2
Arquitectura Bluetooth
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 180

Adicionalmente, un sistema Bluetooth puede tener uno o más `Secondary Controllers` que implementen otro método de transmisión. Cada `controller` ofrece una funcionalidad y características determinadas en sus capas más bajas de la arquitectura y permite que la capas más altas se abstraigan del tipo de transmisión que se emplee.

Dado que cada `controller` ofrece unos principios en la transmisión de datos distinta, se hace necesario conocer el funcionamiento esencial por separado de cada uno de ellos para el propósito de esta tesis.

4.2.1 *Controllers Bluetooth*

4.2.1.1 *BR/EDR*

La capa física (PHY) Bluetooth BR/EDR opera en la banda de libre disposición ISM del espectro electromagnético, concretamente entre las bandas $2.000GHz$ y $2.4835GHz$ (Figura 4.1). Esta banda se divide en 79 canales espaciados $1MHz$. El sistema emplea un mecanismo de salto de frecuencia (FHSS) para evitar las interferencias y permitir muchos portadores FHSS. Se emplea una modulación binaria en frecuencia (Binary FSK) para minimizar la complejidad de la transmisión. El ratio de transmisión de símbolos es de 1 megasímbolo por segundo, por tanto el ratio de bits es de $1Mb/s$. Si dispone de la extensión EDR, el ratio aumenta a 2 o $3Mb/s$ dependiendo de la versión implementada.

Durante una operación de comunicación, un canal físico de radio es compartido por un grupo de dispositivos, que están sincronizados y comparten el patrón para los saltos de frecuencia. El dispositivo que provee la referencia es denominado Master. El resto de dispositvos, que se sincronizan al reloj del Master son denominados Slaves. Un grupo de dispositivos sincronizados siguiendo esta disposición es denominado piconet. La distribución en piconet es uno de los fundamentos principales de las comunicaciones Bluetooth BR/EDR.

Los dispositivos en una piconet usan un patrón de salto de frecuencias determinado, que se determina algorítmicamente empleando entre otros el reloj del master. Este patrón indica el orden pséudo-aleatorio de los 79 canales en los que se realizará la transmisión. Este patrón puede ser adaptado excluyendo canales en los que se estén detectando interferencias. Este mecanismo de salto de frecuencia adaptativo, permite que Bluetooth coexista sin contratiempos con otros sistemas de transmisión inalámbrica que operen en la misma banda de frecuencia. Además, permite que sólo los dispositivos dentro de una piconet, con el mismo patrón de salto, sean capaces de comunicarse entre ellos[3].

Los dispositivos Bluetooth BR/ED se clasifican en tres clases basándose en sus capacidades de potencia de salida más altas (Tabla 4.1).

Tabla 4.1
Clasificación dispositivos Bluetooth en función de su potencia

CLASE	POTENCIA MÁXIMA DE SALIDA	POTENCIA NOMINAL DE SALIDA	POTENCIA MÍNIMA DE SALIDA	ALCANCE APROXIMADO
1	$100mW$ $(20dBm)$	N/A	$1mW$ $(0dBm)$	$\sim 100metros$
2	$2.5mW$ $(4dBm)$	$1mW$ $(0dBm)$	$0.25mW$ $(-6dBm)$	$\sim 5-10metros$
3	$1mW$ $(20dBm)$	N/A	N/A	$\sim 1metro$

3 ↑Aunque, como se presentará más adelante, los dispositivos slaves solo pueden comunicarse con el master.

El canal físico se subdivide en slots de tiempo. Los datos se transmiten entre dispositivos Bluetooth en paquetes posicionados en dichos slots de tiempo. Si las circustancias lo permiten, un número de slots consecutivos se pueden dedicar a la transmisión de un único paquete. El salto de frecuencias tiene lugar entre las operaciones (transmisión o recepción) de paquetes. Bluetooth BR/EDR ofrece un canal de comunicaciones full duplex mediante el empleo un esquema Time-Division Duplex (TDD).

En la pila de protocolos, por encima del canal físico, se encuentran la capa de enlace (link) y canales (channels) y los protocolos de control asociados. La jerarquía de links y channels desde el canal físico hacia arriba se compone por: canal físico (Physical channel), enlace físico (Physical link), transporte lógico (Logical Transport), enlace lógico (Logical Link) y canal L2CAP.

Generalmente dentro de un canal físico, se forma un enlace físico entre el dispositivo master y los slaves. Sin embargo existen excepciones en los que no se generan enlaces físicos, como por ejemplo en los procesos de escaneo (Inquiry scan y Page scan). El enlace físico provee de un medio de transporte de paquetes bidireccional en la mayoría de escenarios, salvo los modos de retransmisión (Connectionless Slave Broadcast) en cuyo caso el enlace físico provee de un canal unidireccional del master hacia los slaves. En ambos casos, no existen enlaces físicos entre los slaves de una piconet.

El enlace físico se emplea como transporte para uno o más enlaces lógicos que soportan el tráfico hacia un único destinatario. Este transporte puede ser síncrono, asíncrono e isócrono, así como tráfico de difusión. El tráfico de los enlaces lógicos se mutiplexa en el enlace físico, ocupando tantos slots como marque un gestor de recursos.

Existe un protocolo de control para las capas físicas de transporte sobre los enlaces lógicos, además de los datos de usuario. Este es el protocolo de gestor de enlace o Link Manager Protocol (LMP). Los dispositivos que están activos en una piconet tienen predeterminado un transporte lógico asíncrono orientado a conexión, empleando para el transporte la señalización marcada por el protocolo LMP, denominado ACL.

El gestor de enlace usa el LMP para controlar las operaciones de los dispositivos en la piconet así como proveer servicios para gestionar las capas inferiores de la arquitectura. El LMP se transporta usando el ACL primario.

Finalmente la capa L2CAP provee una abstracción de los canales a las aplicaciones y servicios. Esta capa se encarga de la segmentación y re-ensamblado de los datos de aplicación, así como de la multiplexación a canales a través de un enlace lógico compartido. L2CAP tiene un canal de control de protocolo que se transporte sobre el ACL definido por defecto. Los datos de aplicación enviados al protocolo L2CAP se pueden llevar a cabo en cualquier enlace lógico que soporte dicho protocolo.

4.2.1.2 *Bluetooth Low Energy LE*

De igual manera que BR/EDR, LE opera en la banda de frecuencia ISM sin licenciar a 2.4GHz, empleando salto de frecuencia FHSS para reducir las interferencias y transmitiendo en modulación binaria en frecuencia (Binary FSK) para minimizar la complejidad de la transmisión.

Sin embargo, LE emplea una terminología diferente para describir las PHYs con diferencias en la modulación, la codificación y la tasa de datos resultante. La tasa de símbolos es de un 1 megasímbolo por segundo, donde 1 símbolo representa un bit y por tanto soporta una velocidad de 1Mb/s. La transmisión LE puede también codificar en el símbolo la correción de errores, que se conoce como `LE 1M PHY`. Esto puede usar dos esquemas de codificación distintos:

$S = 2$ donde 2 símbolos representan 1 bit, y por tanto soporta una velocidad de de 500kb/s.

$S = 8$ donde 8 símbolos representan 1 bit, y por tanto soporta una velocidad de 125kb/s.

El empleo de estos recursos en la redundancia de datos, hace que baje la velocidad de Bluetooth LE drásticamente en comparación con Bluetooth BR/EDR.

LE emplea dos esquemas de acceso múltiple: dividiendo en frecuencia (`Frequency division multiple access o FDMA`) o en tiempo (`Time division multiple access o TDMA`).

FDME La banda de frecuencia se divide en 40 canales físicos separados por 2MHz. Tres de dichos canales se usan como primarios (o anunciación o `Advertising`) y los 37 restantes se emplean como canales secundarios para el envío de datos (Figura 4.3).

TDMA Donde un dispositivo transmite un paquete en un instante determinado de tiempo a otro dispositivo, que emitirá su respuesta después de un intervalo de tiempo predeterminado.

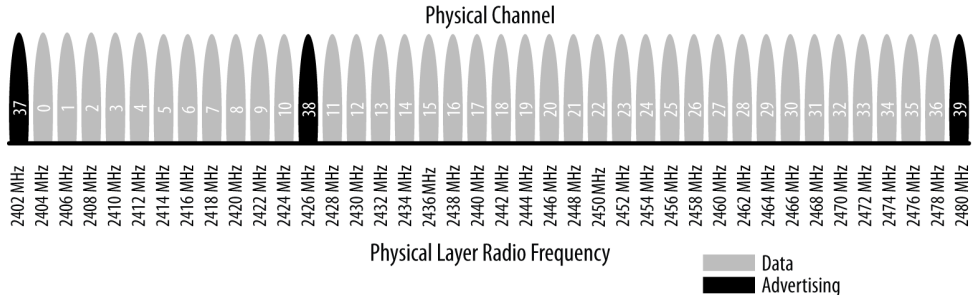


Figura 4.3
Canales empleados en las comunicaciones Bluetooth LE.
Fuente: Getting Started with Bluetooth Low Energy Tools and Techniques for Low-Power Networking [147] - Pag. 170

El canal físico se subdivide en unidades de tiempo denominadas eventos o events. Los datos se transmiten entre dispositivos LE en paquetes posicionados en dichos eventos. Existen cuatro tipos de eventos: anunciación (advertising), anunciación extendido (extended advertising), anunciación periódico (periodic advertising) y conexión (connection).

Los dispositivos que transmiten mensajes de anunciación por el canal PHY son denominados anunciadores o advertisers. Los dispositivos que reciben paquetes de los advertisers sin intención de establecer conexión son denominados scanners.

La transmisión de anuncios por el canal PHY tiene lugar en los eventos de anunciación. Al comienzo de cada evento de anunciación, el advertiser envía un paquete anunciando el tipo de evento anunciado del que dispone. Dependiendo del tipo de evento anunciado y del interés que genere en el scanner, este puede emitir una respuesta empleando el mismo canal PHY en el que se emitió para solicitar la información.

El advertiser, haciendo uso de ese mismo canal PHY, proporciona los datos relativos al evento. En el siguiente evento del mismo tipo, el advertiser cambiará el canal PHY empleado para el anuncio (Figura 4.4).

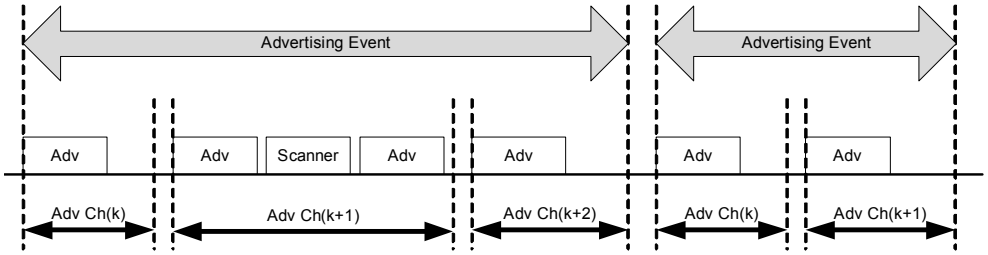


Figura 4.4
Eventos de anunciación empleados en las comunicaciones Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 171

Haciendo uso del mecanismo de eventos, los dispositivos LE pueden comunicarse entre ellos de forma unidireccional sin llegar a establecer una conexión. Es decir, en escenarios donde únicamente se desee mandar información de un dispositivo a otro, sin necesidad de interacción, como por ejemplo un sensor propagando información o un periférico como un ratón indicando los movimientos que se están realizando.

A los dispositivos LE que necesitan establecer una comunicación bidireccional con otro dispositivo LE se le denomina iniciadores o initiators.

El dispositivo initiator espera a que el advertiser anuncie un evento denominado connectable advertising event, al que el initiator responderá empleando el mismo canal PHY. Cuando el evento termina, comienza el proceso de conexión entre los dos dispositivos. Una vez que se establece la conexión, el initiator se convierte en el master de la piconet, y el advertiser en un slave.

Los eventos de conexión o Connection events del advertiser se usan para el envío de datos entre el master y el slave. En los connection events

los saltos de canal se realizan al comienzo del evento, siendo el master el que realiza la primera transmisión al slave. Dicho canal PHY será empleado alternamente por el master y el slave durante la duración del connection event (Figura 4.6).

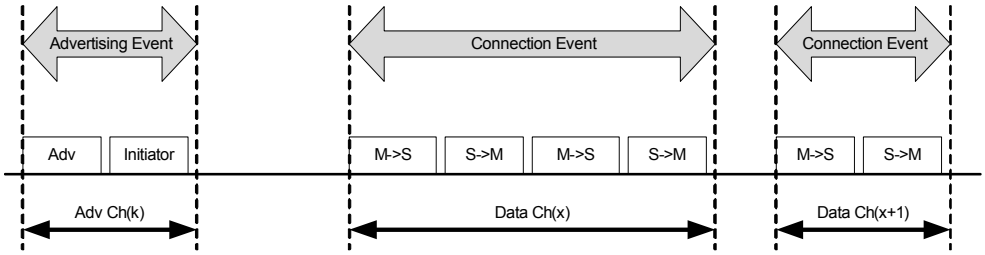


Figura 4.5
Evento de conexión empleado en las comunicaciones Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Figura 1.4

De igual manera que en BR/EDR, los dispositivos en una piconet siguen el mismo patrón de salto de frecuencias, sin embargo en LE las frecuencias empleadas son las pertenecientes a los 37 canales secundarios (Figura 4.3).

Las capas superiores de la arquitectura (Figura 4.2) siguen la misma distribución que en BR/EDR: canal físico (Physical channel), enlace físico (Physical link), transporte lógico (Logical Transport), enlace lógico (Logical Link) y canal L2CAP. Sin embargo presentan algunas diferencias.

A pesar de no existir un canal físico (physical channel), existe un enlace físico (physical link) entre los dispositivos. Este enlace provee el transporte de paquetes bidireccionales entre el master y el slave. Debido a que una piconet puede incluir múltiples slaves, existen restricciones entre los dispositivos que pueden formar un enlace físico. Existe un enlace físico entre cada slave y el master. Los slaves pueden establecer más de un enlace físico con más un master al mismo tiempo para estar conectado a varias piconets simultáneamente, sin embargo en una piconet los slaves no pueden tener enlaces entre ellos. Los roles de master y slave no pueden intercambiarse. El número de scanners e initiators que puede tener un advertiser es potencialmente ilimitado, existiendo entre estos un canal unidireccional.

El enlace físico (physical link) se usa para transportar uno o más enlaces lógicos (logical link) soportando tráfico asíncrono. El tráfico de los logical links se multiplexa en el phisical link llamando a la función de asignación del resource manager.

Un protocolo de control para las capas físicas y de enlace se implementa sobre los enlace lógicos, denominado link layer protocol (LL). Los dispositivos activos en una piconet disponen por defecto de una conexión lógica asíncrona (LE asynchronous connection logical transport o ACL LE) que es usada para portar el protocolo LL.

La capa de enlace usa el protocolo LL para controlar las operaciones entre los dispositivos de la piconet y proveer los servicios para gestionar las capas inferiores de la arquitectura (PHY) y LL).

De igual manera que en Bluetooth BR/EDR, sobre la capa de enlace se encuentra la capa L2CAP que provee una abstracción sobre los canales a las aplicaciones y servicios, encargándose de la segmentación y re-ensamblado de los datos de aplicación, así como de la multiplexación a canales a través de un enlace lógico compartido. L2CAP tiene un canal de control de protocolo que se transporte sobre el ACL definido por defecto. Los datos de aplicación enviados al protocolo L2CAP se pueden llevar a cabo en cualquier enlace lógico que soporte dicho protocolo.

Sin embargo LE provee dos capas por encima del protocolo L2CAP:

SMP o Security Manager protocol) emplea un canal L2CAP determinado para implementar una capa de seguridad entre dispositivos.

ATT o Attribute Protocol provee un método de comunicar pocos datos sobre un canal L2CAP determinado, como por ejemplo, para determinar los servicios y capacidades de otros dispositivos.

Estas dos capas, proveen mecanismos de seguridad y de identificación de servicios necesarios en las transmisiones, ya que dado que no existe conexión no existen mecanismos de seguiridad como en BR/EDR[4].

4.2.1.3 *Bluetooth AMP*

El núcleo Bluetooth prove un mecanismo denominado MAC/PHYs (AMP) que permite el empleo de controladores de radio secundarios. La radio primaria BR/EDR se emplea para el descubrimiento, asociación, establecimiento y mantenimiento de la conexión. Una vez que la conexión L2CAP ha sido establecida entre dos dispositivos sobre BR/EDR, el AMP Manager del master puede descubrir que AMPs están disponibles en el slave. A partir de ese momentoel envío de los datos se transfiere del controlador BR/EDR al controlador AMP.

Cada AMP contiene una capa denominada Protocol Adaptation Layer (PAL) sobre las capas MAC y PHY. La capa PAL mapea los protocolos Bluetooth y el comportamiento especificado por el HCI a los protocolos MAC y PHY en el emisor, y del mapeo inverso en el receptor. Los canales L2CAP se pueden crear o mover a un AMP y de igual manera, pueden ser devuelto a la radio BR/EDR, que en todo momento es la portadora de las comunicaciones de gestión. Finalmente, los AMPs pueden ser activados o desactivados para minimizar el consumo energético de los dispositivos.

4 ↑Donde los dispositivos tienen que comunicarse empleando un patrón de salto determinado, además de otros mecanismos de seguridad que escapan al ámbito de esta tesis.

4.2.2 *Identificación dispositivos Bluetooth*

Cada dispositivo Bluetooth puede ser identificado de forma única mediante un código de 48bits denominado Bluetooth Devices Address (BD_ADDR). Este identificador se corresponde con EUI-48 según se especifica en el estándar IEEE 802-2014 [256] en el apartado de la generación de direcciones únicas (universal addresses). La creación de un EUI-48 requiere los siguientes MAC Address Block que tienen que ser obtenidos de una autoridad registradora del IEEE[56].

[256] 802-2014 - IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture

- MAC Address Block Large (MA-L) o LAP
- MAC Address Block Medium (MA-M) o UAP
- MAC Address Block Small (MA-S) o NAP

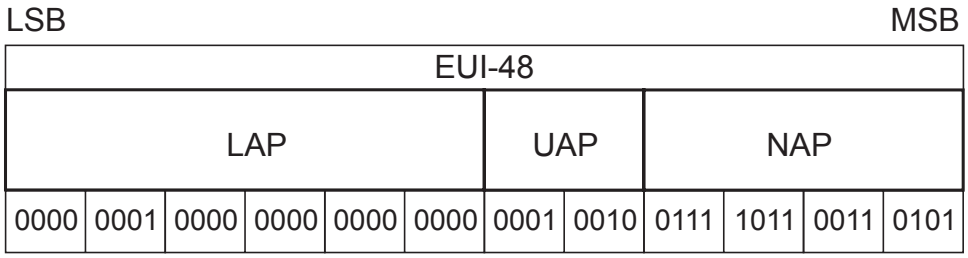


Figura 4.6
Dirección EUI-48 o MAC de un dispositivo Bluetooth
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 357

El campo BD_ADDR puede tomar cualquier valor, excepto un bloque continuo de 64 LAPs[7] empleado para operaciones de investigación y mantenimiento. Uno de estos LAPs reservados es común a todos los dispositivos Bluetooth y los 63 restantes están reservados para caracterizar tipos de dispositivos. Estos LAPs no puede ser parte de ningún BD_ADDR válido.

El uso de este identificador permite reconocer a los dispositivos que ya se han vinculado con anterioridad para poder volver a establecer la conexión. Este mecanismo de identificación único[8] sirve para recordar con que dispositivos ha autorizado el usuario que se puede establecer conexión. Sin embargo, en Bluetooh LE, al no estar orientado a conexión, se utilizan diferentes alternativas de identificación.

5 ↑En http://standards.ieee.org/develop/regauth/index.html se puede obtener información sobre como obtener una dirección MAC.
6 ↑En https://standards.ieee.org/develop/regauth/tut/eui48.pdf se puede encontrar un tutorial sobre como crear direcciones EUI-48
7 ↑De $0x9E8B00$ a $0x9E8B3F$.
8 ↑Y las claves compartidas entre dispositivos, que son mecanismos de seguridad que no vienen al caso.

4.2.2.1 *Identificación de Dispositivos LE*

Los dispositivos que implementan el controlador Bluetooth LE al no requerir una conexión para los estados de anunciación (advertising) o escaneo (scanning) pueden preferir no difundir su BD_ADDR real. En caso de que si decidan hacerlo, se denomina que está propagando su Public Device Address. Si decide no compartir su BD_ADDR, entonces se denomina Random Device Address.

Existen tres métodos de generar una Random Device Address para un dispositivo Bluetooth LE.

Static Device Address

Se genera un dirección de 48 bits generada aleatoriamente (Figura 4.7), pero cumpliendo los siguientes requerimientos:

- Los dos bits más significativos de la dirección se establecerán a 1.
- Al menos uno de los bits aleatorios tiene que ser establecido a 0.
- Al menos uno de los bits aleatorios tiene que ser establecido a 1.

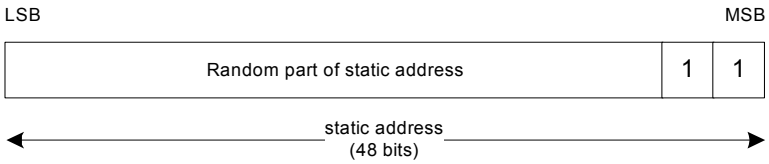


Figura 4.7
Dirección Estática Aleatoria o Static Device Address de un dispositivo Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2557

El dispositivo elige una nueva dirección cada vez que que se produce un nuevo ciclo de carga. El dispositivo no puede cambiar su Static Device Address de ninguna otra manera. Si el dispositivo se encuentra en estado cargando de forma continua, se empleará la Static Device Address generada en el inicio de ese dispositivo hasta que se complete el ciclo de carga.

Non-resolvable Private Device Address

Se emplea un dirección de 48 bits generada aleatoriamente (Figura 4.8), pero cumpliendo los siguientes requerimientos:

- Los dos bits más significativos de la dirección se establecerán a 0.
- Al menos uno de los bits aleatorios tiene que ser establecido a 0.
- Al menos uno de los bits aleatorios tiene que ser establecido a 1.
- La dirección resultante no tiene que ser igual a la dirección BD_ADDR.

En este caso, por tanto, se genera una nueva dirección BD_ADDR totalmente aleatoria. Sin embargo, puede darse el caso de que la dirección generada ya hubiese sido otorgada a otro dispositivo, lo que originaría que dos dispositi-

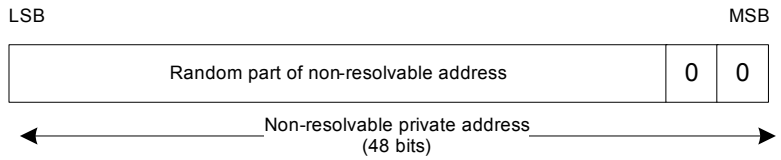


Figura 4.8
Dirección Privada no resoluble o `Non-resolvable Private Device Address` de un dispositivo Bluetooth `LE`
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2558

vos físicos distintos compartiesen la misma clave. Es por ello que su uso está desaconsejado en el protocolo.

`Resolvable Private Device Address`

Para generar esta dirección el dispositivo el dispositivo debe de disponer de una Clave de Identidad local (`Local Identity Resolving Key o IRK`) o de una clave de resolución de identidad de pares (`Peer Identity Resolving Key IRK`). Esta la dirección se generará usando el `IRK` seguido de 24bits generados aleatoriamente. Esta parte aleatoria se denomina `prand` y debe cumplir los siguientes requisitos:

- Los dos bits más significativos de la dirección se establecerán a `0` y `1` respectivamente (Figura 4.9).
- Al menos uno de los bits aleatorios tiene que ser establecido a `0`.
- Al menos uno de los bits aleatorios tiene que ser establecido a `1`.
- La dirección resultante no tiene que ser igual a la dirección `BD_ADDR`.

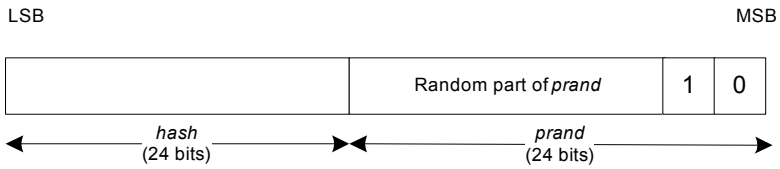


Figura 4.9
Dirección Privada resoluble o `NResolvable Private Device Address` de un dispositivo Bluetooth `LE`
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2558

Para generar los 24 bits faltantes para generar una dirección `EUI-48` válida, se realiza un hash empleando la función `ah` [29] que emplea como parámetros el `IRK` del dispositivo y la semilla de entrada empleada para generar `prand`.

[29] Specification of the Bluetooth System, version 5.0

Debido a que el `IRK` es otorgado por una autoridad verificadora, se tiene un control sobre en que rangos puede generar un dispositivo concreto sus direcciones `BR_ADDR` a difundir cuando usa una `Random Device Address`.

4.2.2.2 *Limitaciones de la identificación Bluetooth LE*

Los tres mecanismos de indentificación presentados anteriormente suponen un impedimento para la identificación unívoca de los dispositivos Bluetooth LE, ya que pueden cambiar de una vez para otra, y además, puede darse el caso de que en dos zonas distintas, dos dispositivos hayan generado la misma clave Random Device Address.

Sin embargo, existen maneras de identificar si el dispositivo Bluetooth detectado ha proporcionado su Public Device Address, o ha proporcionado una Static o una Resolvable Device Address. Los dos bits más significativos de ambos métodos se establen a valores distintos de 00. Las direcciones BD_ADDR se otorgan por el IEEE, de forma que otorga a cada fabricante rangos consecutivos de direcciones acogidas mediante una misma NAP, es decir, mediante los 2 bytes más significativos de la misma.

Esta información es pública, por lo que en base a la parte NAP de una dirección BD_ADDR se puede conocer al fabricante o manufacturador del dispositivo. Debido a que el IEEE ha reservado los dos bits más significativos a Random Device Address, a ningún manufacturador se le ha otorgado un rango NAP cuyos dos bits más significativos sean distintos de 00.

Esta convención en el protocolo de asignación de BD_ADDR, permite por un lado, conocer si la el Device Address difundido por un dispositivo es un Static o un Resolvable Device Address y por tanto no se corresponde con un identificador unívoco. O por el contrario, si es una Public Device Address y determina un dispositivo unívoco, al que además, se le puede conocer su manufacturador consultando la información publicada por el IEEE.

Sin embargo, existe el caso de que un dispositivo Bluetooth LE decida proclamar una Non-Resolvable Device Address, en cuyo caso, no existe manera de discernir si se trata de su dirección BD_ADDR real o si ha sido generada aleatoriamente.

4.2.3 *Búsqueda de dispositivos Bluetooth BR/EDR*

Los dispositivos Bluetooth BR/EDR están fuertemente orientados a conexión (al contrario que los Bluetooth LE), sin embargo proveen algunos servicios que operan sin necesidad de establecer una conexión.Estos servicios se encargan de buscar dispositivos Bluetooth en las inmediaciones y obtener la información necesaria para poder iniciar el proceso de conexión.

4.2.3.1 *Servicios sin conexión*

Los servicios sin conexión que oferta un dispositivo Bluetooth BR/EDR son los siguientes:

`General inquiry`

(Opcional). El propósito de este servicio es ofertar al `initiator` los parámetros requeridos para la conexión de los dispositivos descubribles, como por ejemplo, los dispositivos en el radio de alcance con respecto al `initiator` y que están configurados para escanear los mensajes de descubrimiento. Incluso los dispositivos en el modo limitado de descubrimiento[9] responderán a esta petición. Este servicio se oferta por dispositivos que necesitan descubrir los dispositivos cercanos constantemente o no bajo una condición concreta.

`Limited inquiry`

(Opcional) El propósito de este servicio es el de ofertar al `initiator` los parámetros requeridos para la conexión de los dispositivos descubribles de dispositivos con descubrimiento limitado. Los dispositivos que se encuentren en rango con el `iniciator` y están configurados para responder ante peticiones con el código de acceso limitado, además de aquellos que estén configurados para responder ante las peticiones generales. Este servicio se utiliza en dispositivos en los que el descubrimiento está limitado a momentos determinados, como periodos de tiempo, durante cierta actividad o ante eventos específicos. Dado que no se garantiza que el dispositivo a descubrir responda las peticiones de acceso limitado, el `initiator` puede elegir cualquier procedimiento de consulta (`General inquiry` o `Limited inquiry`). Incluso aunque se espere que el dispositivo remoto a ser descubierto responda a peticiones limitadas, debe hacerse en secuencia con una `General inquiry`, de tal manera que ambas operaciones sean completadas en un marco de tiempo determinado.

`Name discovery`

El propósito de este servicio es proveer al `iniciator` los nombres de los dispositivos Bluetooth conectables. Para utilizar este servicio, el `iniciator`

9 ↑Los modos de descubrimiento se presentarán en la siguiente SubSección 4.2.3.2

debe conocer el BD_ADDR de los dispositivos de los que quiere conocer el nombre (Figura 4.10).

Device discovery

Este servicio sólo está ofertado en los dispositivos que implementan Bluetooth BR/EDR y Bluetooth LE. Su propósito es proveer al `iniciator` la información requerida para la conexión con los dispositivos que se encuentren en el rango con el `iniciator`. Durante el proceso del descubrimiento de dispositivos, primero se realiza un `General inquiry` o `Limited inquiry`, y posteriormente un `Name discovery`.

Bonding

El propósito de este servicio es crear un enlace entre dos dispositivos Bluetooth generando una clave común que es creada por el `iniciator` durante este procedimiento y se intercambia y se almacena por ambos dispositivos Bluetooth. Esta clave se utiliza para la autenticación futura de ambos dispositivos. El protocolo Bluetooth implementa dos variantes de este servicio `General Bonding` y `Dedicated Bounding`, cuya diferencia radica en si la conexión se realiza para acceder a un servicio determinado (`General Bonding`) o sin tener acceso a un servicio concreto `Dedicated Bounding`.

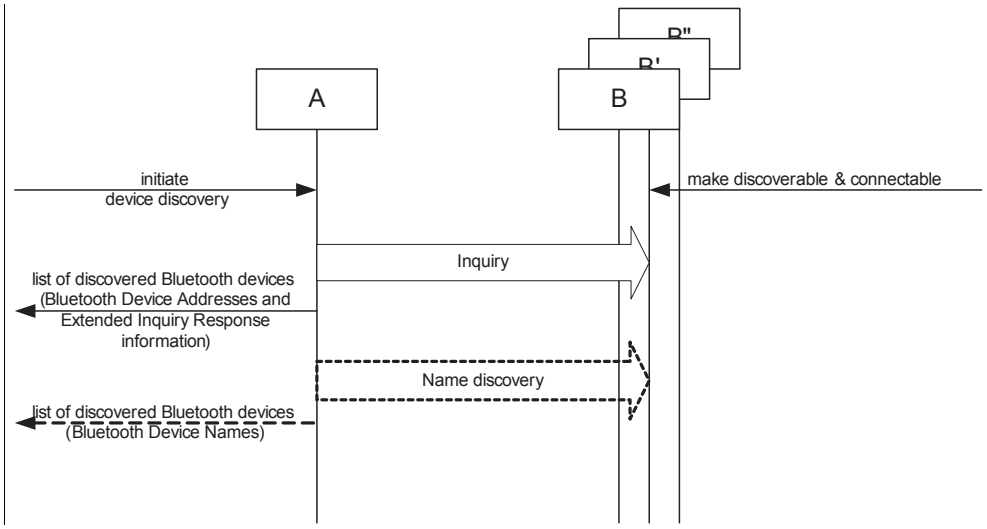


Figura 4.10
Búsqueda de dispositivos Bluetooth
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2027

La existencia de varios modos de descubrimiento de dispositivos Bluetooth, obedece a la existencia de diversos modos en los que un dispositivo Bluetooth se puede encontrar en base al estado y comportamiento que desea ofrecer en caso de que intente ser descubierto.

4.2.3.2 *Modos de descubrimiento*

En la sección anterior se han descrito como funciona la búsqueda de dispositivos Bluetooth y se ha comentado que los dispositivos pueden decidir responder o no a estas búsquedas. Este comportamiento se modela mediante el establecimiento de un modo de descubrimiento o `Discoverability mode`.

Los dispositivos Bluetooth `BR/EDR` pueden implementar cuatro modos de descubrimiento no simultáneos en función de si desean estar disponibles para una conexión o no. Estos modos de descubrimientos son los siguientes:

`Non-discoverable mode`

o modo no-descubrible, el dispositivo no entrará nunca en el estado `INQUIRY_SCAN`, por lo que no responderá a las peticiones de `Device discovery` de ningún `initiator`. Para el usuario, figurará que el dispositivo no se encuentra en las inmediaciones.

`Limited discoverable mode`

o modo descubrible limitado se emplea en los dispositivos que necesitan estar visibles por un tiempo limitado, durante condiciones temporales o eventos específicos. El propósito es el de responder a búsquedas `Limited inquiry`.

`General discoverable mode`

o modo descubrible general, se emplea por los dispositivos que se desea sean detectables de forma continuada en el tiempo o no supeditados a ninguna condición concreta. El propósito es el de responder a búsqueda `General inquiry`.

`Discoverable mode`

o modo descubrible, es la generalización de los estados `Limited` o `General discoverable`.

De todos ellos, el único modo obligatorio del protocolo es el modo `General discoverable`, resultando el resto de modos opcionales. Esto es debido a que es el modo imprescindible para poder realizar una conexión. Los otros modos `Limited` y `Non-discoverable` suelen requerir de algún tipo de interfaz en el propio dispositivo que le permita conmutar entre los demás modos.

4.2.3.3 *Descubrimiento de dispositivos Bluetooth BR/EDR*

Debido a que Bluetooth BR/EDR está orientado a conexión, la búsqueda de dispositivos cercanas es importante en el protocolo. En los apartados anterior se han definido tanto los servicios que oferta el protocolo Bluetooth que no requieren conexión y que sirven para establecerla, así como los modos de descubrimiento que implementa el protocolo.

Con los servicios y modos, el protocolo Bluetooth construye dos mecanismos para la búsqueda de dispositivos en las inmediaciones. El empleo de uno u otro se marca en función de si se desea hacer la búsqueda una única vez (o modo ONE-TIME INQUIRY - Figura 4.11) o si la búsqueda tiene que hacerse de forma ininterrumpida, periódica y continuada en el tiempo (o modo PERIODIC INQUIRY - Figura 4.12).

El protocolo de búsqueda de un dispositivo Bluetooth por parte de un iniciator comienza con la emisión de un comando HCI_Inquiry (Figura 4.11(a)) al LM para realizar una ONE-TIME INQUIRY o de un comando HCI_Periodic_Inquiry (Figura 4.12(a)) para realizar una búsqueda periódica.

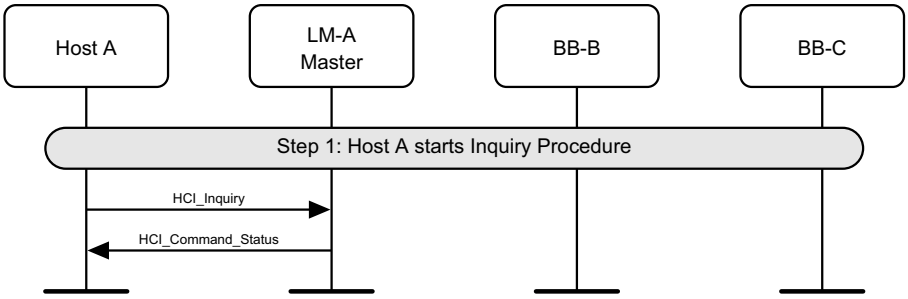
El controlador comenzará la consulta de la banda base esperando las respuestas de los dispositivos cercanos. Los dispositivos que escuchan el HCI_Inquiry y están en modo Discoverable envían un paquete FHS (Figuras 4.11(b) y 4.12(b)) que el LM del initiator recibe y propaga a capas superiores del protocolo.

La búsqueda ONE-TIME INQUIRY finaliza cuando el initiator envía un comando HCI_Inquiry_Cancel (Figura 4.11(c)). O bien, en la petición ONE-TIME INQUIRY se puede indicar al LM un criterio de parada[10] (Figura 4.11(d)). Una vez la búsqueda ha terminado, se envía la lista de paquetes FHS recibidos desde dispositivos cercanos.

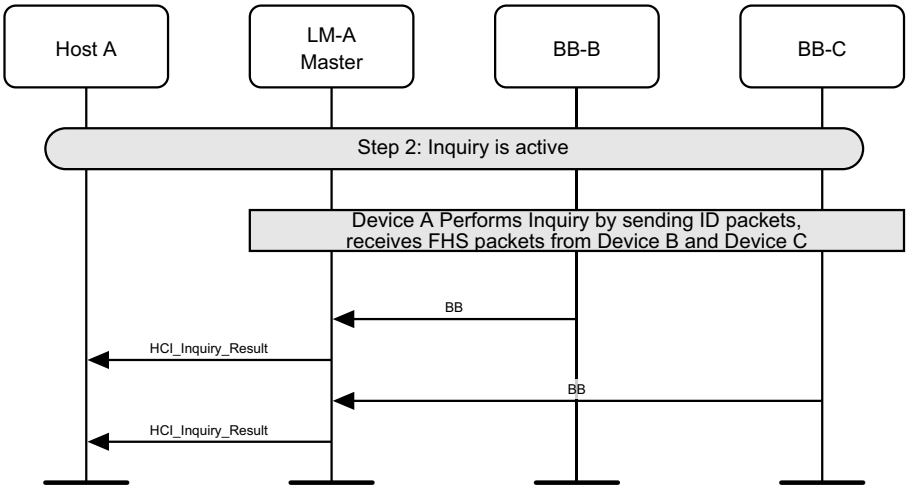
La búsqueda PERIODIC INQUIRY sólo puede ser detenida mediante el envío de un comando HCI_EXIT_Periodic_Inquiry (Figura 4.12(d)). Sin embargo, requiere también un criterio de parada de igual naturaleza que el empleado por ONE-TIME INQUIRY. Hasta que no se cumpla dicho criterio de parada, el LM no enviará la lista de FHS al initiator (Figura 4.12(c)). Dicho criterio de parada no interrumpe el PERIODIC INQUIRY, que seguirá propagando y recibiendo un paquete FHS hasta que reciba un HCI_EXIT_Periodic_Inquiry.

La búsqueda de dispositivos Bluetooth BR/EDR consiste por tanto en la emisión de un determinado paquete, y la obtención de una respuesta por parte de los dispositivos cercanos. Esta respuesta, se envía formateada en un paquete denominado FHS.

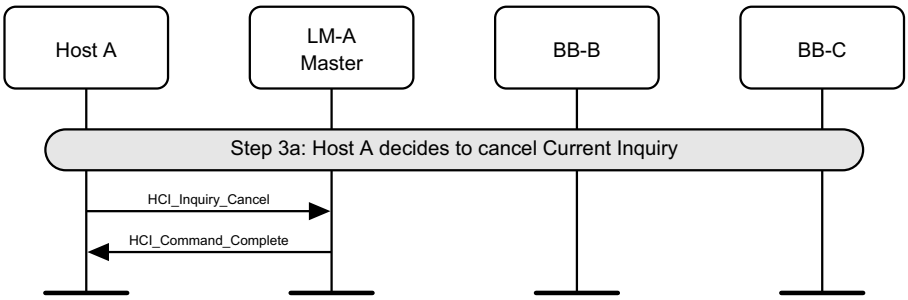
10 ↑El criterio de para puede estar basado en la búsqueda durante un tiempo determinado o en la detección de una cierta cantidad determinada de dispositivos



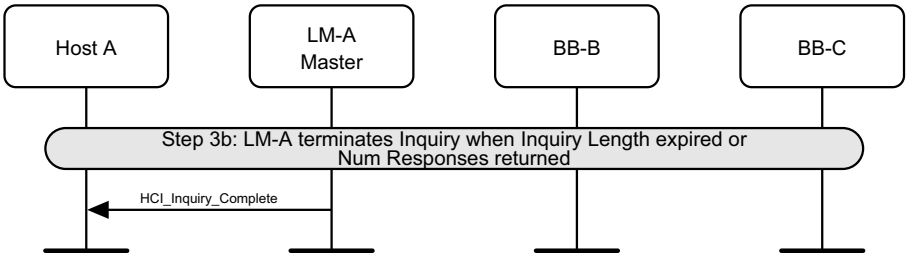
(a) Paso 1



(b) Paso 2

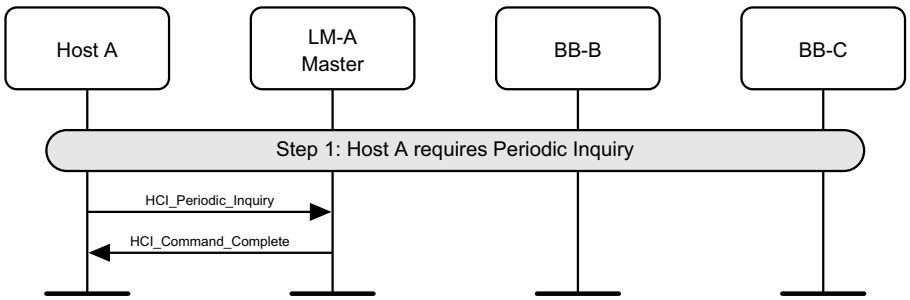


(c) Paso 3

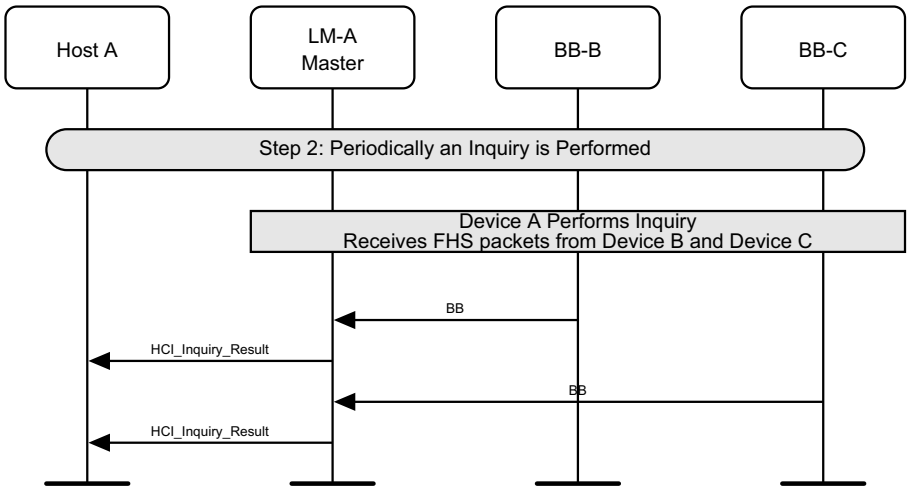


(d) Paso 4

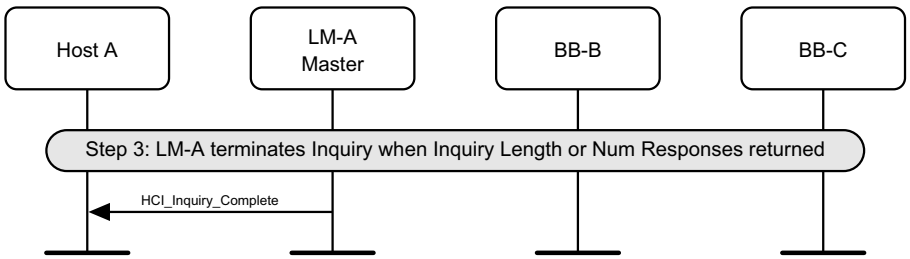
Figura 4.11
Pasos del protocolo de búsqueda única de dispositivos Bluetooth.
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 1416



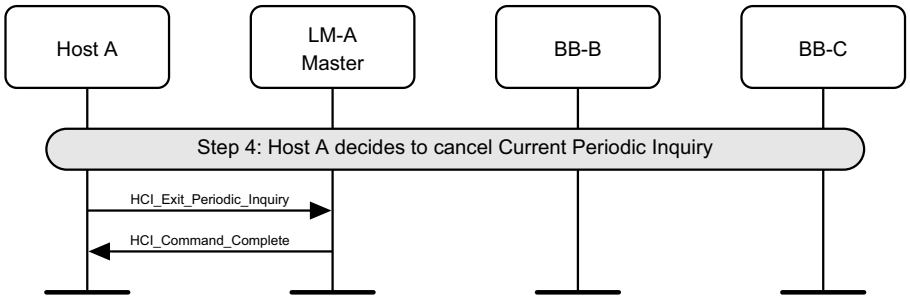
(a) Paso 1



(b) Paso 2



(c) Paso 3



(d) Paso 4

Figura 4.12
Pasos del protocolo de búsqueda periódica de dispositivos Bluetooth.
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 1418

4.2.3.4 *Paquete FHS*

El paquete FHS contiene, entre otras cosas, la dirección del dispositivo Bluetooh y el reloj del remitente (Figura 4.13). Se emite en respuesta a los peticiones de escaneo por los dispositivos cercanos que se encuentran en estado Discoverable. El paquete FHS no está cifrado, ni dispone de ningún tipo de mecanismo de seguridad que encripte la información que porta.

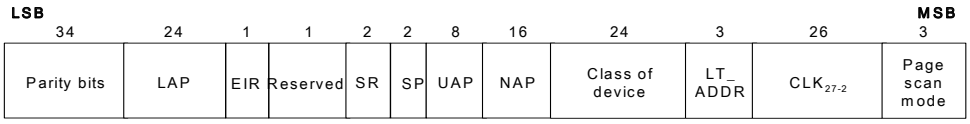


Figura 4.13
Paquete FHS usado en las comunicaciones Bluetooth BR/EDR
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 413

Su estructura se encuentra definida con los siguientes campos:

PARITY BITS (34 bits) Contiene los bits de paridad que forman la primera parte de la palabra de sincronización del código de acceso del dispositivo que envía el paquete.

LAP (24 bits) Contiene parte de la dirección física o BD_ADDR[11] del dispositivo.

EIR (1) Indica que una respuesta extendida tiene que llegar al dispositivo.

-RESERVED- (1) Reservado para uso futuro.

SR (2) Indica el intervalo entre dos ventanas consecutivas de exploración de página.

SP (2) Establecidos a 10 en la versión actual del protocolo. Resto de valores reservados par uso futuro.

UAP (8) Contiene parte de la dirección física o BD_ADDR[12] del dispositivo.

NAP (16) Contiene parte de la dirección física o BD_ADDR[13] del dispositivo.s

CLASS OF DEVICE (24) Indica la naturaleza del dispositivo según se codifica en las tablas del Anexo A.1.

LT_ADDR (3) Dirección lógica de transporte que se emplea en el establecimiento de conexión, en un cambio de rol o en tareas de re-envío dentro de una piconet.

CLK_27-2 (26 bits) Contiene el valor del reloj nativo del dispositivo que envía el FHS tomado en el instante de realizar la transmisión. Este reloj dispone de una precisión de 1.25ms (dos slots de tiempo).

PAGE SCAN MODE (3) Este campo indica el modo de escaneo por defecto por el emisor del paquete FHS. Actualmente sólo se emplea la combinación 000, estando reservadas las otras para usos futuros.

11 ↑ tal y como se ha descrito en la sección 4.2.2
12 ↑ tal y como se ha descrito en la sección 4.2.2
13 ↑ tal y como se ha descrito en la sección 4.2.2

Del paquete FHS resultan relevantes los campos relacionado con la dirección BD_ADDR definidos en la Sección 4.2.2 y el campo Class of device. Este campo, codifica la clase o naturaleza del dispositivo que ha emitido el paquete FHS.

4.2.3.5 *Clases de dispositivos Bluetooth*

En el paquete FHS se reservan 24 bits para codificar el tipo de dispositivo Bluetooth que ha emitido el paquete (Figura 4.14). De estos 24 bits se emplean 5 bits para codificar el tipo principal del dispositivo o Mayor device class (Tabla 4.2), 6 bits para codificar el subtipo del dispositivo (Tablas A.1-A.11), 11 bits para codificar los servicios (Service Class) y 2 bits reservados para uso futuro.

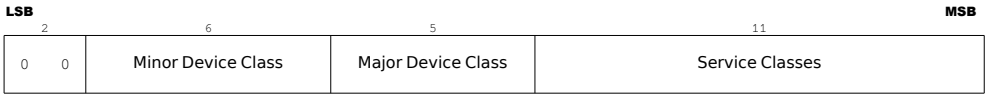


Figura 4.14
Codificación de la clase de dispositivo en el Paquete FHS usado en Bluetooth BR/EDR
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 414

Tabla 4.2
Tipo de dispositivos Bluetooth según su Major Device Class
Fuente: Bluetooth Baseband [252].

12	11	10	9	8	MAJOR DEVICE CLASS
0	0	0	0	0	Misc
0	0	0	0	1	Computer
0	0	0	1	0	Phone
0	0	0	1	1	Network
0	0	1	0	0	Audio/Video
0	0	1	0	1	Peripheral
0	0	1	1	0	Imaging
0	0	1	1	1	Wearable
0	1	0	0	0	Toy
0	1	0	0	1	Health
1	1	1	1	1	Uncategorized
X	X	X	X	X	–Reserved–

En base a estos bits que se puede obtener información sobre la naturaleza del dispositivo, lo que permitirá al dispositivo iniciator conocer que servicios puede ofrecerle el dispositivo cercano [14]. Sin embargo, la existencia de estos servicios resulta irrelevante para la temática de esta tesis, que se aprovechará únicamente en la naturaleza del dispositivo detectado.

14 ↑Así por ejemplo, un dispositivo de clase Computer puede ofrecer servicios el intercambio de ficheros, mientras que un dispositivo Peripheral puede ofrecer información empleada como interfaz de entrada.

4.2.4 *Búsqueda de dispositivos Bluetooth LE*

Al contrario que Bluetooth BR/EDR, Bluetooth LE no está orientado a conexión, por lo que ofrece una gran cantidad de servicios que no requieren la existencia de un enlace entre los dispositivos. Al contrario que Bluetooth BR/EDR, estos servicios no se centran en el establecimiento de conexiones, sino que están centrado en la emisión direccional de información del `master` a los `slaves` de la `piconet`.

Esta peculiaridad hace que los dispositivos puedan estar en más modos que los dispositivos Bluetooth BR/EDR, que haciendo un ejercicio de abstracción, sólo pueden estar en dos modos, `No-Discoverable` si no quieren conexión con nadie y `Discoverable` en caso contrario.

4.2.4.1 *Estados de un enlace entre dispositivos Bluetooth LE*

El estado del enlace entre un dispositivos Bluetooth LE en una `piconet` puede ser descrito en términos de la máquina de estados que se representa en la Figura 4.15.

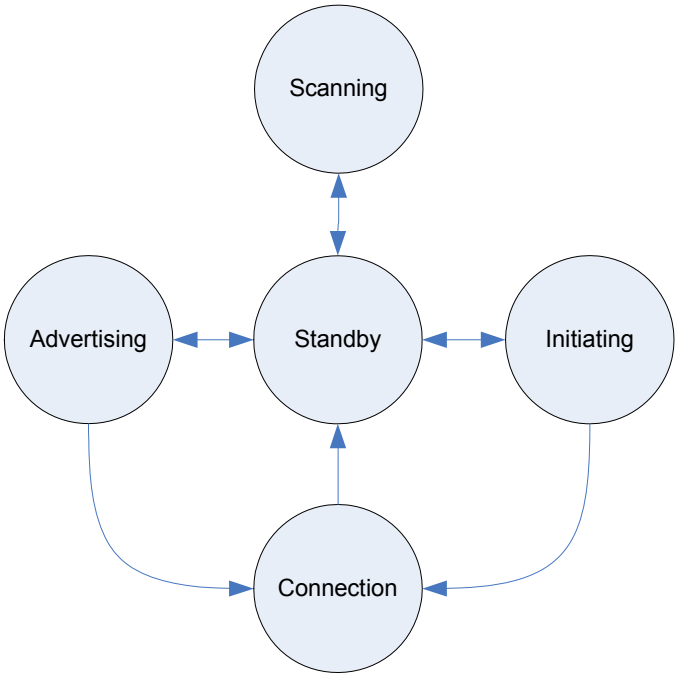


Figura 4.15
Diagrama de estados de enlace de un dispositivo Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2554

La capa de Enlace permite que solo un estado esté activo a la vez para una piconet concreta. Los estados son los siguientes:

Standby State En este estado no se transmite ni recibe ningún paquete. Este estado puede ser activado desde cualquier otro estado.

Advertising State En este estado, se estarán retrasmitiendo paquetes de anunciación (advertising) y respondiendo a las solicitudes de respuesta activadas por dichos canales. Un dispositivo en este estado se conoce como anunciante o advertiser. A este estado, sólo se puede llegar desde el estado de Standby State.

Scanning State En este estado el dispositivo estará escuchando paquetes de canales de anunciación y emitiendo solicitudes de respuesta para los advertising que le interesen. Un dispositivo en este estado se conoce como escaner o scanner. A este estado, sólo se puede llegar desde el estado de Standby.

Initiating State En este estado se escuchan los canales de anunciación de un dispositivo específico y respondiendo para iniciar una conexión con ese dispositivo. El dispositivo en este estado se denomina iniciador o initiator. Este estado sólo puede ser ingresado desde el estado de Standby.

Connection State En este estado se ha producido una conexión entre los dispositivos de la piconet. Dentro de dicha conexión se definen dos roles: master y slave. A este estado se puede llegar bien desde el Scanning State o desde Initiating State.

Estos modos, por tanto, están más centrado en la emisión o recepción de información que en la visibilidad de los propios dispositivos.

4.2.4.2 *Descubrimiento de dispositivo Bluetooth LE*

Debido a que Bluetooth LE no se encuentra centrado en la conexión, se centra más en la búsqueda de servicios ofertados por los dispositivos Bluetooth en las inmediaciones que en la propia búsqueda de dispositivos Bluetooth con los que conectarse. La búsqueda de estos servicios corre a cargo de los scanners que se encargan de buscar los paquetes de anunciación de servicio que están emitiendo los advertiser. Aunque Bluetooth LE implementa numerosos servicios de escaneo para los scanners según los criterios privacidad y la periodicidad deseada, se pueden dividir en dos tipos distintos:

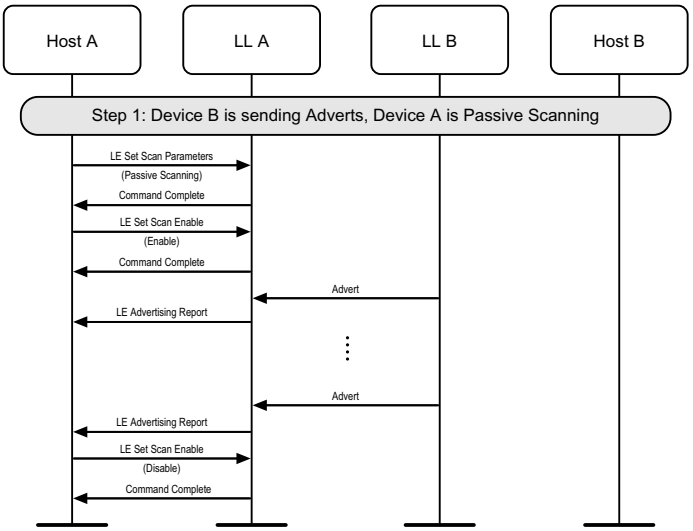
Pasive Scanning

El escaneo pasivo, empleado para conocer que dispositivos cercanos están en modo advertising. El controlador capturará los paquetes advert[15] emitidos y lo notificará al dispositivo scanner (Figura 4.16(a)).

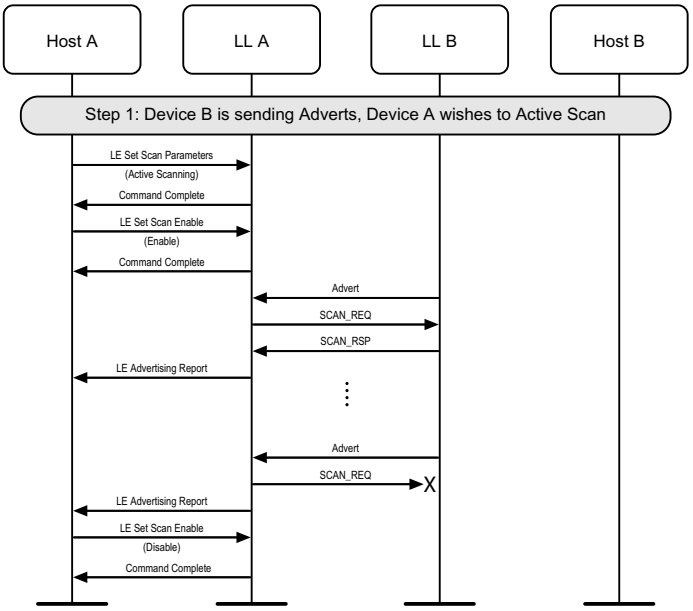
15 ↑Se definirá este paquete en la sección 4.2.4.3, donde se definen el paquete estándar empleado en las comunicaciones Bluetooth LE.

Active Scanning

El escaneo activo se emplea para obtener aún más información que en el `pasive scanning`. A parte del mensaje `Advert`, implica emitir por parte del `scanner` una petición `SCAN_REQ` a cada servicio detectado, esperando una respuesta `SCAN_RSP`. (Figura 4.16(b)).



(a) Escaneo pasivo



(b) Escaneo activo

Figura 4.16
Escaneo pasivo y activo de anunciaciones en Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2724-2425

Estos paquetes contiene toda la información recogida por el `scanner` de los servicios anunciados por los dispositivos en las inmediaciones.

4.2.4.3 *Paquetes `Advert` y `SCAN_RSP`*

Los procesos de escaneo de los dispositivos Blueooth `LE` obtienen como respuesta de los dispositivos en las inmediaciones dos tipos de paquetes distintos en función de si se desea una búsqueda pasiva o activa: los paquetes `Advert` y los paquetes `SCAN_RSP`.

4.2.4.3.1 Paquete `Advert`

Se emplea tanto para los canales de `advertising` como para los de datos. Cada paquete (Figura 4.17) consta de cuatro campos obligatorios:

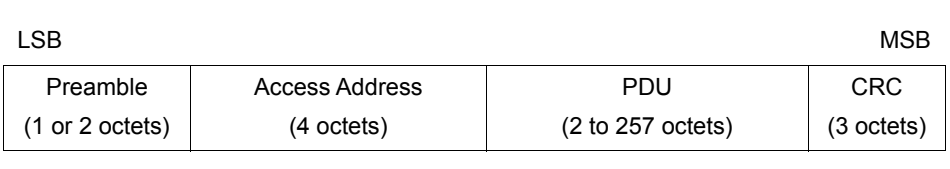


Figura 4.17
Composición del paquete `Advert` devuelto en los escaneos de dispositivos Bluetooth `LE`
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2562

`PREAMBLE` (1 o 2 octetos). Se emplea para establecer la frecuencia de sincronización, mostrando una secuencia alterna de ceros y unos. Los paquetes transmitidos con `LE 1M PHY` tienen una longitud de 8 bits, mientras que los transmitidos con `LE 2M PHY` constan de 16 bits.

`ACCESS ADDRESS` (4 octetos). Se genera por la capa de enlace para cada PDU de `advertising` que se envíe. La capa de enlace en el estado `advertising` generará una nueva dirección de acceso cada vez que permita el envío de un conjunto de anunciación distinto.

`PDU` (De 2 a 257 octetos). En un paquete advertising enviado por el canal primario, contiene la descripción de los datos que se está anunciando. En los canales secudarios, contiene los propios datos que se han solicitado mediante la respuesta a un mensaje de `advertising` por parte de un `scanner`.

`CRC` (3 octetos). Es un código de Verificación por redundancia cíclica[16] realizado sobre el PDU.

En base a este paquete, se puede obtener información sobre el servicio que está siendo anunciado, pero ninguna información relativa sobre el dispositivo que lo anuncia. El campo `PDU` contiene la identificación la información de los datos.

Al realizar un escaneo pasivo, el `scanner` no recibe ninguna información adicional. Sin embargo, en un escaneo activo (Figura 4.16(b)), el `scanner` mandará una `SCAN_REQ` que será respondida por un `SCAN_RESP` por parte el `advertiser`. En ese caso, el campo `PDU` contendrá cierta información sobre el `advertiser`.

16 ↑Descrito en profundidad en [29], página 2601)

4.2.4.3.2 Paquete PDU en respuesta SCAN_RESP

Se ha comentado que el paquete advert devuelve un PDU que codifica la información del servicio. En términos generales, un PDU en Bluetooth LE se compone de una sucesión de AD Structures, que contienen la longitud de los datos, los bits que componen el dato y el tipo de dato que se adjunta que marcan el formato de interpretación o parseo. (Figura 4.18).

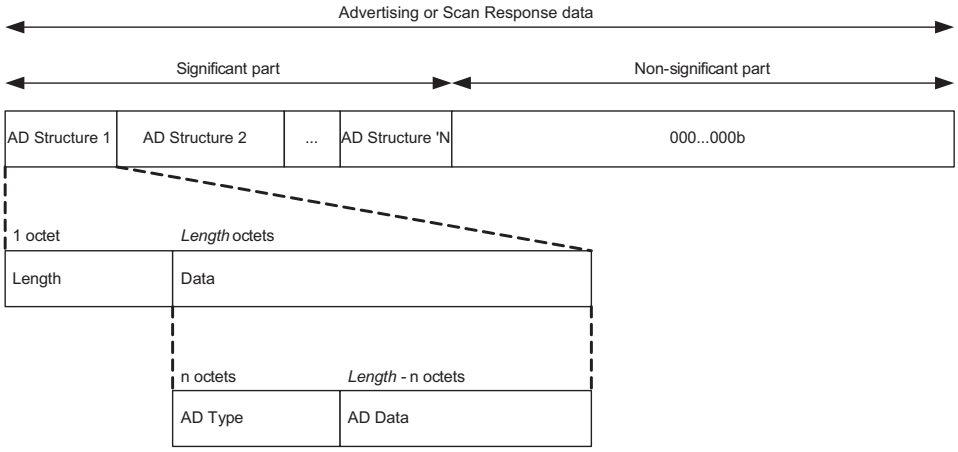


Figura 4.18
Composición del PDU empleado en las comunicaciones Bluetooth LE
Fuente: Specification of the Bluetooth System 5.0 [29] - Pag. 2562

Cuando se realiza una petición SCAN_REQ, el campo PDU de la respuesta SCAN_RESP se emplea para proporcionar cierta información sobre el dispositivo. La información a proporcionar está estandarizada, pero ningún campo es de carácter obligatorio, pudiendo elegir el advertiser la información que difundirá en la respuesta.

Sin entrar a definir todas los campos o AD Structures que pueden ser proporcionados en una SCAN_REQ[17], enunciaremos los que usualmente son empleados: ADV_IND, ADV_NONCONN_IND, AUX_ADV_IND, AUX_CHAIN_IND y AUX_SYNC_IND.

Las estructuras que empiezan por ADV, contienen el campo AdvA de 48 bits que se corresponde con la dirección Public Device Address o Random Device Address, según haya estimado el advertiser. Además, contienen el campo AdvData que contiene la información de anunciación del advertiser.

Sin embargo, según se ha visto en la sección 13, los dispositivos Bluetooth LE no siempre divulgan su BD_ADDR real o Public Device Address. Ya que puede decidir difundir su Random Device Address, o incluso si se desea, no enviar ninguna. Lo cual supone un problema para la identificación del dispositivo.

17 ↑Pues hacerlo escapa del ámbito de este documento. Se puede encontrar en Specification of the Bluetooth System 5.0 [29] - Pag. 2562 a 2600

4.2.5 *Adecuación del protocolo Bluetooth para la monitorización*

Presentado el protocolo Bluetooth, se presenta la discusión sobre su adecuación para la monitorización de dispositivos y su posterior aplicación a la monitorización de personas y/o vehículos. Si bien, como ambas versiones de transmisión del protocolo ofrecen comportamientos distintos, serán abordadas por separadas.

4.2.5.1 *Bluetooth BR/EDR*

En el caso de Bluetooth BR/EDR, al estar fuertemente orientado a conexión entre dispositivos, dispone de mecanismos o servicios empleados de para la detección de dispositivos en las inmediaciones. Además, esta detección provee de numerosa información que incluye la dirección unívoca del dispositivo.

Esta identificación se entrega por una identidad certificadora, el IEEE, otorgando bloques consecutivos a un mismo fabricante, de forma que puede ser reconocido el fabricante del dispositivo que se ha sido detectado.

Del mismo modo, el protocolo provee una taxonomía exhaustiva de las diversas naturalezas de dispositivos que emplean comunicaciones Bluetooth. Esta naturaleza se otorga junto a la dirección del dispositivo, y por extensión, su fabricante. Estas características ofrecen un escenario alentador para justificación de la Hipótesis I y por tanto, para la monitorización de los dispositivos Bluetooth BR/EDR.

De esta forma, es viable de forma nativa al protocolo detectar los dispositivos en las inmediaciones y obtener información sobre ellos. Sin embargo, los dispositivos pueden implementar el modo `No-discoverable`, que hace que permanezcan ocultos.

4.2.5.2 *Bluetooth LE*

En el caso de Bluetooth LE, el protocolo está más orientado a la detección de servicios anunciados que de dispositivos concretos. Este hecho dificulta la detección de dispositivos cercanos, debido a que un mismo dispositivo puede anunciar un número indeterminado de servicios. La falta de un estándar en la identificación del dispositivo, así como la posibilidad de que el dispositivo no comparta su identificador real, dificultan la viabilidad de una identificación unívoca.

Dado que el protocolo limita la velocidad de transmisión de datos por debajo de $1 Mb/s$, su uso puede resultar insuficiente en escenarios con un ratio de bits superior. Además, Bluetooth LE está desarrollado con una fuerte atención en el ahorro de energía, y con un carácter de transferencia de información direccional[18]. Su uso se ha popularizado en dispositivos sensores, werables, periféricos o iBeacons [191], donde la información solo se trans-

18 ↑Al contrario que Bluetooth BR/EDR que ofrece un canal bidireccional.

mite del advertiser al scanner, con poca o nula comunicación en sentido contrario.

Estas peculiaridades en el diseño, hacen que la detección de dispositivos Bluetooth LE no ofrezca un escenario que resulte idóneo para la monitorización de dispositivos extendida a la monitorización de personas y vehículos.

Sin embargo, aunque este hecho pueda parecer tomar un matiz negativo, supone una ventaja ya que acota perfectamente la naturaleza de los dispositivos Bluetooth que pueden ser detectados y monitorizados.

4.2.5.3 *Bluetooth para la detección de vehículos*

Aunque en el Estudio 6.1.3, se estudiarán en profundidad los dispositivos Bluetooth que han sido detectados en el desarrollo de esta tesis así como su naturaleza y fabricantes, anticipamos en base a la definición del protocolo la siguiente sentencia, cuyas pruebas y evidencias empíricas serán presentadas más adelante:

La detección de dispositivos Bluetooth resulta idónea para la identificación de vehículos.

Los dispositivos Bluetooth que pueden ser detectados mediante la monitorización de sus comunicaciones, emplean en su mayoría la forma de transmisión BR/EDR. Dado que Bluetooth LE se ha centrado en la mejora de eficiencia energética, es comprensible que los dispositivos que hagan uso de esta trasmisión empleen baterías con una capacidad energética limitada, y por tanto, se prioriza la eficiencia energética

Los vehículos en movimiento, disponen de un motor de combustión alimentando una batería de alta capacidad[19], habitualmente proporcionando $12V$ y con una capacidad nominal de entre $60Ah$ y $100Ah$ en los turismos convencionales. Dado que el consumo nominal de los dispositivos Bluetooth BR/EDR (Tabla 4.1) suele rondar los $30mAh$ [158], el consumo de un dispositivo Bluetooth BR/EDR no supone un grave impacto para la batería de vehículos, al contrario que en otros escenarios, como un pequeño sensor, un periférico, un teléfono móvil o un wearable.

[158] A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi

Además, los dispositivos que van a ser emplazados y alimentados por un vehículo suelen prescindir de implementar el modo de no descubrimiento, opcional en el protocolo, ya que no supone ningún ahorro energético significativo, y requeriría habilitar una interfaz para la conmutación de modos, en función de si el usuario quiere o no emplear comunicaciones Bluetooth. Además, este modo es fácilmente simulado cuando el vehículo se detiene y se apaga, momento en el cual los dispositivos internos no prioritarios interrumpen su alimentación. De esta forma, el dispositivo Bluetooth solamente estaría en funcionamiento cuando el vehículo está en marcha, y por tanto, existe una persona conduciéndolo que puede emplear la comunicación Bluetooth.

19 ↑O en el caso de coches eléctricos, una única batería de gran capacidad que alimenta todo el vehículo

Por otro lado, dentro de los dos modos de descubrimiento disponibles en el protocolo suelen hacer uso del `General discoverable mode`, de forma que el descubrimiento del dispositivo no requiere la pulsación de ningún elemento físico, solamente los procesos requeridos para la vinculación[20]. De esta forma, aunque el dispositivo disponga de un botón, este se emplea para la vinculación del dispositivo, y no para habilitar el descubrimiento del mismo.

Finalmente, y como se presentará en el capítulo TO DO, en la práctica la mayoría de dispositivos detectables obedecen a dispositivos con unas naturalezas o clases concretas (Tabla 4.2. Estas clases, pertenecen a dispositivos que con una alta probabilidad se encuentran dentro de un vehículo, ya sea de forma nativa o acoplada. Esta clase complementada con la información del fabricante, habilita el establecimiento de una serie de criterios que permiten discernir cuando un dispositivo Bluetooth detectado pertenece o no a un vehículo. Y de esta forma filtrar de forma exhaustiva los dispositivos en los que existen dudas.

Debido a todas estas consideraciones, y sin presentar todavía evidencias fruto de los experimentos de campo[21] se puede afirmar que la captación de dispositivos bluetooth puede ser útil para la monitorización de vehículos y en menos medida para la monitorización de personas.

4.2.6 *Legalidad de la captación Bluetooth*

Debido a que la búsqueda de dispositivos está soportada de forma nativa por el protocolo, la búsqueda de dispositivos Bluetooth en las inmediaciones no supone la vulneración de ninguna ley ni reglamento.

Las tramas capturadas, se emiten por los dispositivos Bluetooth en respuesta a una petición de información (`HCI_INQUIRY`), enviando un paquete (`FHS`) que es enviado sin mecanismos de seguridad que tengan que ser vulnerados.

La captura de otras tramas Bluetooth, por ejemplo las pertenecientes a las comunicaciones de dos dispositivos que forman una `piconet` entre ellos, resulta inviable sin realizar ataques o aprovechar vulnerabilidades del protocolo [111, 154], ya que implicaría conocer al menos las claves del dispositivo, el reloj del dispositivo `master` y el patrón de salto de frecuencias. Este tipo de ataques, siempre que existan mecanismos de seguridad a ser atacados, supone un delito y están amparados tanto por la declaración de Derechos Humanos [17], como por normativas y reglamentos locales, por ejemplo en España, por la Constitución[22] o por el Real Decreto de 14 de septiembre de 1882 [220].

[111, 154] Man-in-the-middle attacks on bluetooth: a comparative analysis, a novel attack, and countermeasures, "Man in the Middle" Attacks on Bluetooth

[17] Universal declaration of human rights

[220] Real Decreto de 14 de Septiembre de 1882

20 ↑Que no han sido descritos en profundidad en esta tesis, pero que en este ámbito sólo se precisa saber que implica la generación o compartición de un código concreto entre ambos dispositivos.

21 ↑Que serán presentadas en el capítulo 6.

22 ↑Artículo 18.3

4.3 WIFI

El nombre WiFi proviene de la marca de la Alianza WiFi, que es la organización comercial que adopta, prueba y certifica el cumplimiento de los estándares IEEE 802.11 [253] relacionados con las redes inalámbricas de área local. Este estándar define el uso de los niveles inferiores de la arquitectura[23] de las redes inalámbricas que se puede ver en la Figura 4.19: la capa física (`physical layer o PHY`) y la capa de enlace de datos (`media acces control` o MAC).

[253] 802.11-1999 - Standard for Information Technology

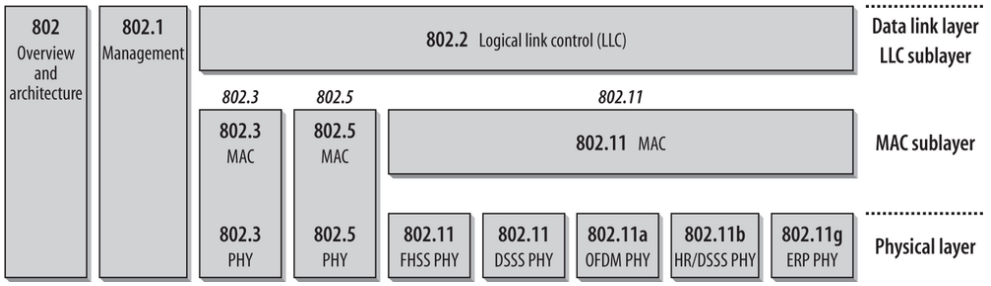


Figura 4.19
Relación entre la familia IEEE 802.11 y el modelo OSI
Fuente: 802.11 Wireless Networks: the definitive guide [176] - Pag. 35

Al igual que Bluetooth, el estándar 802.11 engloba varias versiones (o protocolos) que ofrecen técnicas de modulación `half duplex` por medio del aire. Las versiones más extendidas son la 802.11b y la 802.11g que operan en la banda de frecuencias de los 2.4GHz con 11 canales superpuesto a 4 canales de distancia, un ancho de banda de señal de 22MHz, con separaciones de 5MHz. Debido a la saturación de la banda de frecuencia de los 2.4GHz, la versión 802.11a propone el empleo de la banda U-NII de 5GHz, ofreciendo al menos 23 canales no superpuestos. La versión 802.11n puede emplear la banda de 2.4GHz o la de 5GHz a conveniencia, mientras que la versión 802.11ac emplea sólo la banda de los 5GHz.

Si bien la banda de los 5GHz está menos saturada que la banda de 2.4GHz, su alcance es menor y presenta mayor dificultad ante obstáculos, debido a que cuanto mayor es la frecuencia menor es el rango efectivo cubierto [176]. Por ello, la mayoría de los smartphones y dispositivos inteligentes operan casi exclusivamente en la banda de los 2.4GHz. Sin embargo, algunos smartphones de gama alta están empezando a incorporar WiFi 802.11ac y WiFi 802.11n, aunque sin prescindir nunca de la compatibilidad con WiFi 802.11b y 802.11g.

[176] 802.11 Wireless Networks: The Definitive Guide (O'Reilly Networking)

23 ↑O modelo OSI.

4.3.1 *Componentes de las redes 802.11*

Las redes 802.11 implican a cuatro componentes o elementos físicos en el proceso de transmisión de datos inalámbricos representado en la Figura 4.20.

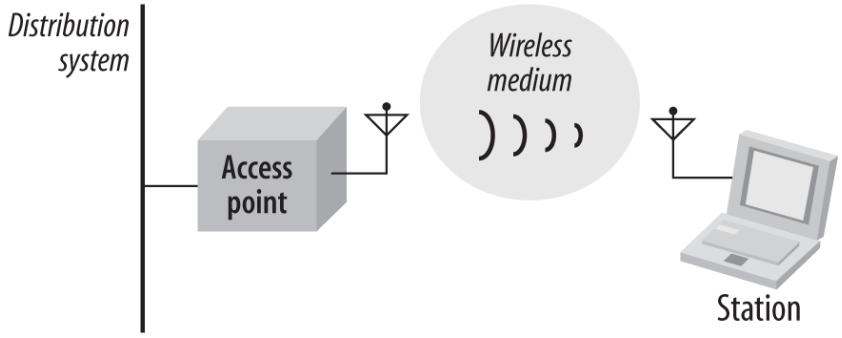


Figura 4.20
Componentes de una red de comunicaciones inalámbrica 802.11
Fuente: 802.11 Wireless Networks: the definitive guide [176]

ESTACIONES (STATIONS) son dispositivos con capacidad de cómputo e interfaces de red inalámbricas entre las que se transmiten datos de forma inalámbrica.

PUNTO DE ACCESO (ACCESS POINT O AP) dispositivo puente encargado de la conversión entre tramas inalámbricas a tramas cableadas y viceversa, así como otras tareas de encaminamiento y gestión de la red.

MEDIO INALÁMBRICO (WIRELESS MEDIUM) es el medio físico entre el que se realiza el intercambio de tramas entre estaciones y puntos de acceso.

SISTEMA DE DISTRIBUCIÓN (DISTRIBUTION SYSTEM) encargado de la interconexión mediante un medio cableado de los distintos AP.

Las comunicaciones inalámbricas 802.11 se sustentan en la existencia de un área[24] de cobertura denominada BSS o Basic service set que ofrece un conjunto de servicios de comunicaciones básico. Cuando una estación se encuentra se encuentra en un BSS puede comunicarse con otros miembros pertenecientes al BSS. Si el BSS consta de un punto de acceso, se habla de un BSS de Infraestructura o Infraestructure BSS. En caso contrario, se denomina BSS Independiente o Independent BSS. Estas dos topologías se muestran en la Figura 4.21.

24 ↑Muchos autores, con los que coincido, señalan que el término correcto debería ser volumen ya que las ondas se propagan en espacio tridimensionalmente. Sin embargo, el término área se ha impuesto en la nomenclatura.

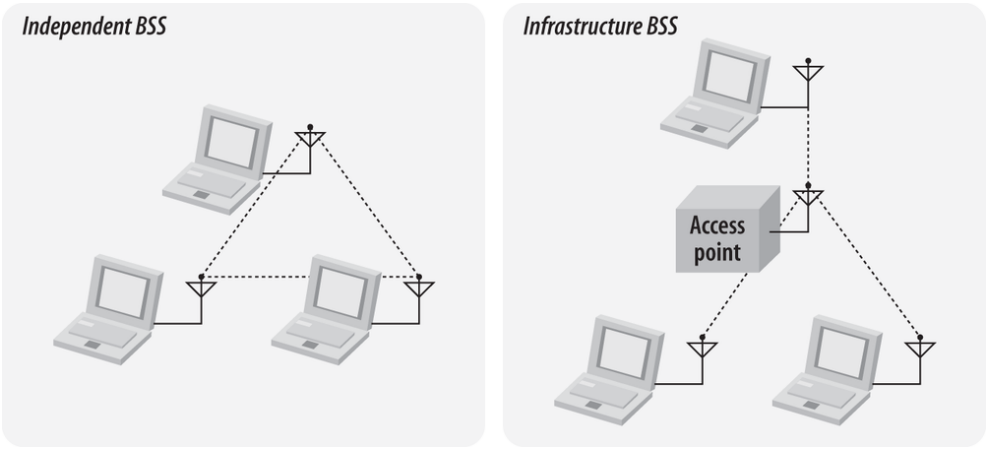


Figura 4.21
BSS independiente y de infraestructura en una red 802.11
Fuente: 802.11 Wireless Networks: the definitive guide [176]

Sin embargo, la mayoría de despliegues de redes que se realizan empleando el protocolo 802.11 son BSS de Infraestructuras, donde generalmente el AP provee de una salida a Internet mediante el sistema de distribución para permitir su acceso a todas las estaciones conectadas al BSS.

4.3.2 *Servicios de red en 802.11*

El protoloco 802.11 está basado en el protocolo Ethernet [251], por lo que está diseñado para funcionar como si fuese una capa de enlace situada justo por encima de dicho protocolo. De igual manera a Ethernet (o Bluetooth presentado anteriormente), las estaciones y puntos de acceso se identifican mediante una dirección IEEE 802 MAC de 48bits [256] otorgada por un organismo regulador.

[251] 802.3 Ethernet Working Group

[256] 802-2014 - IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture

Se ofrecen nueve servicios, de los que tres se emplean para el movimiento de datos y seis para operaciones de gestión y administración de red. Estos servicios se recogen en la tabla 4.3.

Tabla 4.3
Servicios de red existentes en el protocolo 802.11

SERVICIO		ELEMENTO	DESCRIPCIÓN
Distribución	(Distribution)	Distribución	Envía tramas para determinar la dirección de destino en redes BSS de infraestructura.
Integración	(Integration)	Distribución	Envía de tramas a una red IEEE 802 LAN externa a la red inalámbrica
Asociación	(Association)	Distribución	Establece el AP que sirve como puente para una estación particular.
Reasociación	(Reassociation)	Distribución	Cambia el AP que sirve como puente para una estación particular.
Desasiciación	(Disassociation)	Distribución	Elimina la estación de la red.
Autenticación	(Authentication)	Estación	Establece la identidad de la estación antes de la asociación.
Desautenticación	(DeAuthentication)	Estación	Termina la autenticación de la estación, y por tanto, también la asociación.
Confiabilización	(Confidentiality)	Estación	Provee protección contra la intromisión en la red (Aevesdropping).
Entrega MSDU	(MSDU delivery)	Estación	MAC service data unit (MSDU) provee de datos del receptor.
TPC	(Transmit Power Control)	Estación	Reduce interferencias mediante la minimización de la potencia de transmisión.
DFS	(Dynamic Frequency Selection)	Estación	Evita interferencias con operaciones de muestreo en la banda de los 5GHz.

De estos servicios se extraen las características que definen la naturaleza de las redes 802.11. Los servicios de asociación indican que las estaciones deben de asociarse para poder establecer comunicación ya sea con otras estaciones en el caso de BSS Independientes o con puntos de acceso en el de BSS de Infraestructura. La comunicación entre dos estaciones que no estén asociadas con una BSS común no es posible salvo que cada estación esté asociada a un BSS unida mediante un sistema de distribución cableado. Dos estaciones que no pertenezcan a una BSS no podrán establecer comunicación entre ellas aunque se encuentren en el mismo rango de alcance, salvo que se constituyan como una BSS Independiente.

Debido a que el medio que se emplea es común a todas las estaciones independientemente de que pertenezcan o no una BSS determinado, el protocolo impone que los AP requieran verificación de su identidad para poder unirse al BSS, añadiendo una capa de seguridad a la autenticación de las estaciones.

4.3.3 *Escaneo de BSS*

Dado que las estaciones no se encuentran en un medio cableado, su localización no resulta inmediata. Como se ha expuesto anteriormente, en las redes inalámbricas las estaciones deben de incorporarse previamente a una BSS antes de establecer comunicación con un AP u otra estación. Por tanto, el protocolo debe de proveer mecanismos que permitan a una estación localizar las BSSs cercanos, con el fin de poder vincularse a ellas. El proceso de identificación de estas BSS por parte de una estación se denomina escaneo o `scanning`.

El protocolo determina varios parámetros implicados en el proceso de escaneo, en función de las necesidades y requerimientos de la estación. Los más relevantes para el ámbito de monitorización son:

`BSSType` (`independent`,`infraestructure`, `or both`) Se emplea para indicar el tipo de BSS que se desea encontrar.

`BSSID` (`individual or broadcast`) Indicando si se desea escanear una BDSS específica (`individual`) o a cualquier red que permita unirse a estaciones (`broadcast`) obteniendo una lista con todas las BSS cercanas.

`SSID` (`‘‘network name’’`) Es un conjunto de caractéres asignado a una o más BSS para facilitar la identificación. El SSID no es un identificador único ni unívoco, pudiendo ser compartido por varias BSS.

`ScanType` (`active or pasive`) Indica el tipo de escaner que debe realizarse.

`ChannelList` Representa la lista de canales en las que se realizará el escaneo, pudiendo ser una selección o todos los disponibles.

De estos parámetros destacar el parámetro `ScanType` que indica que en el protocolo IEEE 802.11 se definen dos tipos de escaneo de BSSs, que se describen a continuación.

Escaneo Pasivo

El escaneo pasivo supone un menor coste energético porque no requiere el envío de ninguna trama por parte de la estación. En este escaneo, la estación itera por los distintos canales esperando la escucha de unas tramas denominadas `beacon` emitidas por los `AP` u otras estaciones en un `BSS` Independiente. En base a las tramas `beacon` recibidas, se proporciona información sobre los `BSS`s cercanos. Este tipo de escaneo se representa en la Figura 4.22.

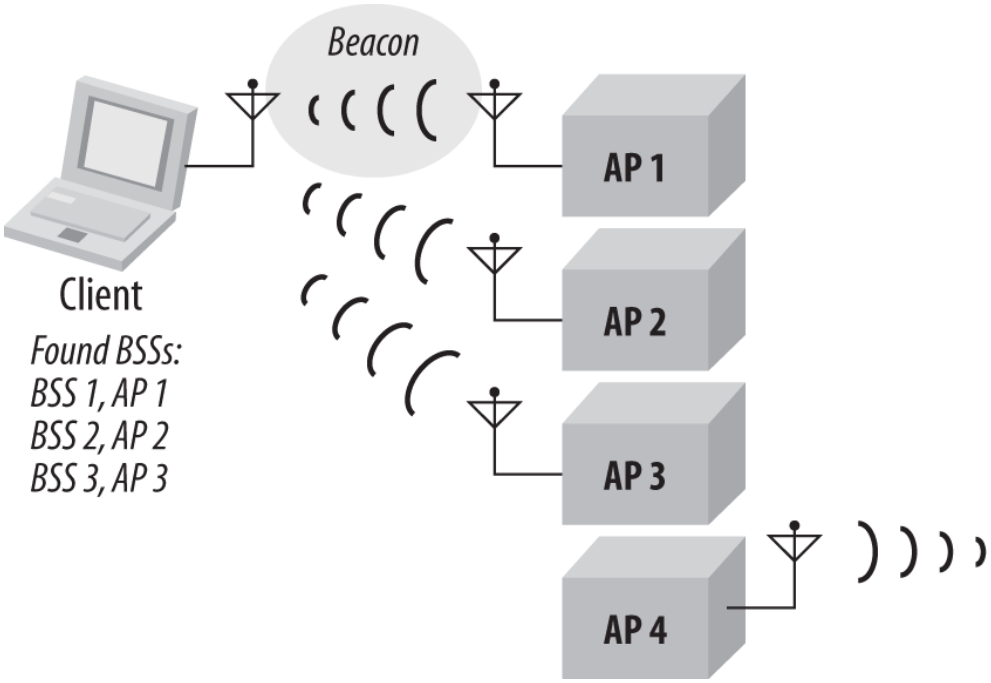


Figura 4.22
Esquema de escaneo pasivo de una estación de red 802.11, obteniendo como resultado la lista de `BSS` y sus respectivos `AP`.
Fuente: 802.11 Wireless Networks: the definitive guide [176]

El escaneo pasivo está diseñado para obtener toda la información de la trama `beacon` necesaria para la asociación de la estación a la `BSS`. Sin embargo, presenta algunas limitaciones ya que requiere que los `AP` estén propagando tramas `beacon` anunciándose.

Sin embargo, por motivos de seguridad y privacidad, los `AP` puede decidir permanecer "silenciosos" sin proclamar que hay una `BSS`. Este caso, por ejemplo es en el que se encuentran las redes configuradas para no anunciar su `SSID`.

Escaneo Activo

En este escaneo, las estaciones envían para cada canal una trama de solicitud de sondeo o `Probe Request` a una red determinada mediante un `SSID` concreto. Por tanto, para poder realizar este escaneo, la estación tiene que conocer el `SSID` de la `BSS` a la que conectarse, ya sea porque se ha conectado con anterioridad a dicha `BSS` (y conserva almacenado el `SSID`) o bien porque se le proporciona por alguna vía, como por ejemplo, mediante interacción con el usuario o la detección mediante un escaneo pasivo.

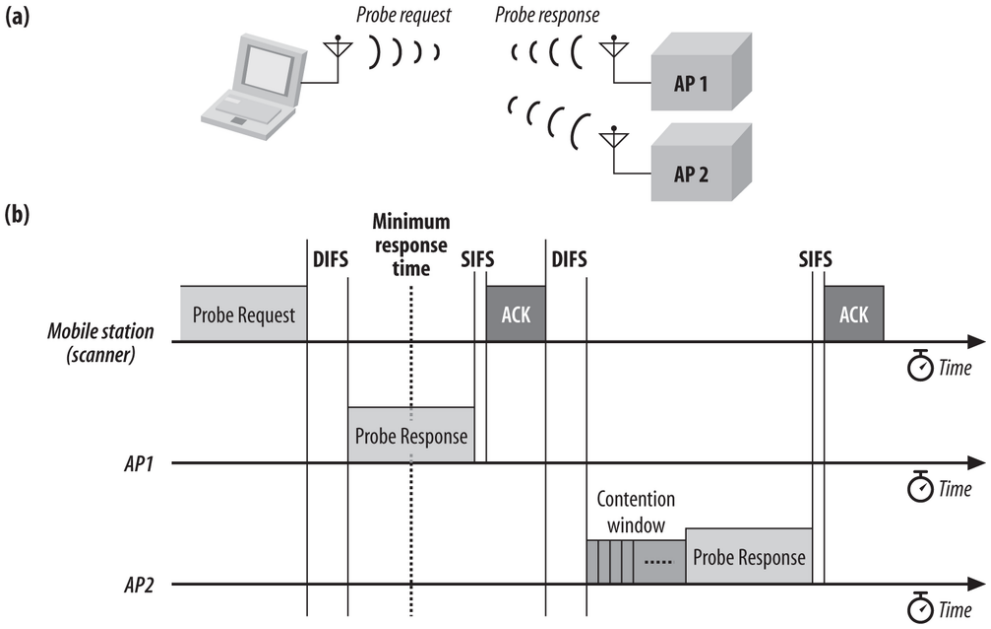


Figura 4.23
Esquema del escaneo activo de una estación de red 802.11, con los diagramas de tiempo de las comunicaciones implicadas.
Fuente: 802.11 Wireless Networks: the definitive guide [176]

En respuesta al `Probe Request` los `AP` que se corresponden con la `BSS` determinada por la `SSID` emiten una trama de respuesta (`Probe Response`).

Este tipo de búsquedas son habituales en entornos donde los `AP` no propagan la información de la `BSS` o donde su `SSID` permanece oculta. Es por ello que las estaciones suelen estar programadas para enviar periódicamente peticiones `Probe Request` a las redes 802.11 con las que han establecido una conexión previa.

Resultados del escaneo

Tanto el escaneo activo como el escaneo pasivo, producen un `Scan Report` al término del proceso donde se listan las `BSS` descubiertas y los parámetros necesarios para la conexión a cada una de ellas.

Por tanto, en contraposición a Bluetooth que si provee mecanismos nativos para la detección de dispositivos, las redes WiFi 802.11 no proveen de mecanismos para la detección de dispositivos cercanos, sino que disponen de

mecanismos de detección de BSSs a las que poder conectarse. Sólamente podría detectarse si una estación está configurada como una BSS Independiente podría ser detectado, sin embargo, la información que se estaría adquiriendo en el Scan Report sería relativa a la BSS y no a la propia estación física.

4.3.4 *Modos de funcionamiento de las interfaces 802.11*

Si bien el protocolo no provee de servicios nativos para la búsqueda de dispositivos WiFi, si que dispone de mecanismos adicionales, generalmente empleados para la auditoría y control de las redes, que se emplean para este fin.

La distinción entre las estaciones y los AP (Figura 4.20) viene dada según el modo de funcionamiento en el que la interfaz de red inalámbrica está establecida. El protocolo WiFi 802.11 provee siete modos de funcionamiento en los que una interfaz de red puede funcionar:

MAESTRO (Master) Modo de funcionamiento de los AP que proveen de una BSS de Infraestructura.

ESTACIÓN (Managed or station) Modo de funcionamiento de las estaciones que son conectadas a una BSS por medio de un AP.

AD HOC (Ad hoc) Modo de funcionamiento para la constitución de una BSS Independiente de una estación.

MALLA (Mesh) Modo de funcionamiento que permite la conexión directa entre estaciones, prescindiendo de APs y sin que ninguna de ellas tenga que operar como modo Ad hoc.

REPETIDOR (Repeater) Modo de funcionamiento cuyo cometido es el de ampliar la señal WiFi recibida de uno o varios emisores dentro de una BSS y emitirla amplificada con el fin de ampliar el rango de la señal inicial.

PROMISCUO (Promiscuous) Modo empleado para capturar todo el tráfico de una BSS concreta, a la cual la interfaz de red se encuentra asociada.

MONITOR (Monitor) Modo empleado para capturar todo el tráfico recibido por la interfaz de red, trafico que es generado por todas las BSSs cercanas, sin necesidad de estar asociado a ellas.

El modo Monitor, permite obtener todo el tráfico de red que se esté generando entre todas las BSSs cercanas, así como todo el tráfico asociado a la búsqueda de BSS activa y/o pasiva. Si un dispositivo está emitiendo comunicaciones WiFi, una tarjeta de red WiFi en modo monitor y en el alcance de dichas comunicaciones, podrá capturar todo tráfico de red generado en su entorno, aunque el tráfico no vaya dirigido a ella ni sea una estación vinculada a las BSS en las que se están generando las comunicaciones capturadas.

4.3.5 *Trama WiFi*

Todas las comunicaciones empleadas en WiFi emplean el mismo formato de trama de red. Ya que en el modo monitor, se capturan todas las tramas de red generadas por las comunicaciones WiFi en las cercanías, se hace necesario estudiar el formato de dichas tramas para conocer que información puede extraerse que sea útil para la monitorización de dichos dispositivos.

Cada trama de red WiFi contiene los siguientes tres componentes básicos (Figura 4.24):

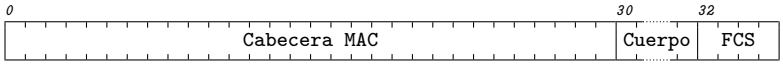


Figura 4.24
Trama MAC empleada en las comunicaciones WiFi. Cada bloque representa 1 byte.
Fuente: 802.11-1999 - Standard for Information Technology [253]

CABECERA MAC Contiene principalmente información para el control de la trama, duración y las direcciones MAC implicadas en la comunicación. Opcionalmente dispone de información sobre el control de secuencias de las tramas, sobre criterios de calidad de servicio (o QoS)[25] o campos de control HT[26].

CUERPO DE LA TRAMA Contiene la información transmitida. El tamaño del cuerpo, así como el tipo (y subtipo) de información que se está transmitiendo viene especificado en la cabecera MAC.

FCS que contiene un CRC de 32bits siguiendo el estándar del IEEE.

De estos componentes, tanto el cuerpo de la trama como el FCS no aportan información sobre el emisor, por lo que para la monitorización de dispositivos solo resulta relevante la cabecera MAC. La cabecera MAC se encuentra formateada con los siguientes campos(Figura 4.25):

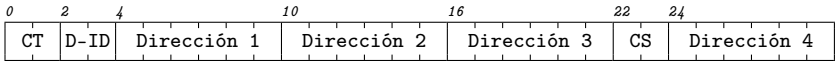


Figura 4.25
Cabecera MAC empleada en las comunicaciones WiFi. Cada bloque representa 1 byte.
Fuente: 802.11-1999 - Standard for Information Technology [253]

CONTROL DE TRAMA (CT) Contiene 2bytes con información sobre la propia trama[27]

DURACIÓN/ID (D-ID) Dispone de diversos usos según se emplee para tramas NAV, CFP o PS-Poll. Su información no resulta relevante para la monitorización.

DIRECCIONES 1,2,3,4 Direcciones de red siguiendo el estándar IEEE MAC[256] de las estaciones implicadas en la comunicaciones. La naturaleza de cada dirección varía según el tipo de trama.

[256] 802-2014 - IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture

25 ↑Solo tramas que hagan uso de QoS
26 ↑Solo tramas +HTC
27 ↑Se presentará al detalle a continuación.

CONTROL DE SECUENCIA (CS) Campo de 16 bits que se emplea tanto para la desfragmentación de las tramas como para la gestión de tramas duplicadas descartadas.

Cómo se aprecia en la cabecera MAC (Figura 4.25), una trama WiFi puede implicar a hasta cuatro tarjetas de red identificadas por sus direcciones MAC. Una de dichas direcciones es la de la tarjeta de red del dispositivo que ha realizado la emisión original de la trama. Otra de las direcciones es la que identifique a la tarjeta de red dentro de la BSS que tiene que recibir la trama. Las otras direcciones posibles se emplean en los re-direccionamientos y encaminamientos de la trama a través de distintos AP.

Gracias al procesamiento de la trama, se identifica mediante su dirección MAC al dispositivo que ha originado y emitido la trama. El empleo de una posición de dirección dentro de la cabecera de la trama, viene determinado en función del tipo de trama.

Este tipo de trama viene determinado dentro de la cabecera MAC, en los dos bytes que son empleados para el campo de control de trama (Figura 4.26):

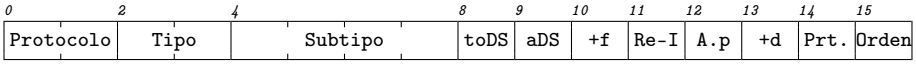


Figura 4.26
Cabecera de Control de Trama MAC empleada en las comunicaciones WiFi. Cada bloque representa 1 bit.
Fuente: 802.11-1999 - Standard for Information Technology [253]

PROTOCOLO indicando la versión del protocolo. Actualmente únicamente se emplea la versión 802.11 codificada con los bits 00, pues hasta el momento ninguna de las revisiones del protocolo ha necesitado una nueva especificación del formato de trama.

TIPO Y SUBTIPO indican el tipo de trama propagada. En el Anexo A.2 se presentan tablas que codifican los tipos y subtipos de las tramas.

TODS Y ADS indican si la trama está destinada para un sistema de distribución.

+F bit que indica que se ha fragmentado el paquete.

RE-I bit que indica que la trama se ha reintentando transmitir.

A.P bit que indica si el emisor está en modo ahorra de potencia.

+D bit que indica que se encuentra disponible al menos una trama más.

PRT bit que indica que la trama está protegida.

ORDEN bit que indica que la trama se envía en orden estricto.

El tipo y subtipo de la trama indica la naturaleza de la trama emitida, según la codificación que aparece en el Anexo A.2. En este punto, solo es necesario determinar que actualmente en el protocolo existen tres tipos de tramas distintas, que son presentadas a continuación:

ADMINISTRACIÓN son las tramas relacionadas con los servicios de asociación a BSSs.

CONTROL son las tramas relacionadas con los mecanismos de control y gestión de errores de la comunicación de red.

DATOS son las tramas que portan datos de las comunicaciones de red.

Una vez conocido el tipo de trama, es factible discernir cual de los campos de dirección de la trama se corresponde con la dirección física o MAC de la tarjeta de red del dispositivo que ha emitido originalmente dicha trama. Este punto resulta crítico para poder afirmar que el dispositivo que ha emitido la trama se encuentra en el alcance de la interfaz de red en modo monitor que la ha capturado. Ya que si la trama ha sido por ejemplo reenviada por uno o varios AP es posible el dispositivo que lo emitió originariamente no esté cerca.

4.3.6 *Adecuación de WiFi 802.11 para la monitorización*

Como se ha presentado, el protocolo 802.11 ofrece mecanismos de búsqueda y detección de BSS, careciendo de servicios nativos para el descubrimiento de dispositivos WiFi en la inmediaciones.

Empleando una tarjeta de red inalámbrica configurada en modo monitor, es posible detectar todo el tráfico de red generado por los dispositivos WiFi en las inmediaciones. Este tráfico puede ser generado tanto mediante las comunicaciones y transmisiones de datos que se estén produciendo entre las estaciones conectadas al AP de una BSS, como por las propias estaciones en la búsqueda de BSS con los que autenticarse y conectarse.

Los dispositivos que emplean WiFi pueden hacer uso dos de tipos de búsqueda de BSS una búsqueda pasiva que no implica la emisión de ninguna trama por parte del dispositivo; y la búsqueda activa, que implica la emisión de tramas por cada red identificada por su SSID de la que dispone el dispositivo, normalmente, porque se ha conectado con anterioridad.

Esto implica que los dispositivos que emplean WiFi están normalmente buscando los BSS con los que establecido comunicación con anterioridad, y están emitiendo tramas sondeando si dicha BSS está en las cercanías.

Estas tramas,se pueden capturar y procesar para extraer la dirección MAC del dispositivo que ha originado la trama. Además, es posible discernir si el dispositivo original que ha emitido la trama se encuentra en las inmediaciones de la interfaz de red en modo monitor mediante en estudio del tipo y subtipo de trama capturada. Esto permite afirmar que un dispositivo con una interfaz de red identificada por su dirección MAC unívoca y única, estaba en un determinado marco temporal cerca del punto geógrafico donde se encuentra emplazada la tarjeta de red en modo monitor que ha capturado dicha trama.

Todas estas consideraciones proporcionan un escenario válido para la monitorización de los dispositivos WiFi en las inmediaciones.

4.3.6.1 *WiFi para la detección de personas*

Aunque en la Sección 6.1 se estudia en profundidad los dispositivos WiFi que han sido detectados en el desarrollo de esta tesis, así como sus fabricantes, se anticipa en base a la definición del protocolo la siguiente sentencia, cuyas evidencias empíricas serán presentadas más adelante:

La detección de dispositivos WiFi resulta idónea para la identificación de personas y sus movimientos.

Debido a que las comunicaciones WiFi, se han convertido en el estándar de facto en las comunicaciones de red inalámbrica WLAN la práctica totalidad de dispositivos portables. De toda la gama de dispositivos que emplean WiFi, el `smartphone` es aquel que es empleado en mayor medida mientras se realizan desplazamientos. De esta forma, dispositivos como tablets o portátiles no son normalmente usado en desplazamientos.

Además, a los smartphones se les impone que estén constantemente actualizando su información, por lo que son dispositivos que en todo momento están a la espera de conexión a redes inalámbricas. El `smartphone` emplea a su vez, la detección de redes inalámbricas en las cercanías para otro tipo de tareas ajenas a la conexión de red, como por ejemplo, para su propio posicionamiento.

Por último, los smartphones buscan constantemente a las redes conocidas de las que disponen su `SSID` y credenciales de conexión almacenadas, lo que se denomina redes conocidas. Debido a que estas redes pueden no propagar su `SSID`, los `smartphones` están preparados para sondearlas constantemente.

Además, según los estudios presentados en la Sección 2.1.3.1 la implantación de los `smartphones` en la población es prácticamente total, de forma que es asimilable identificar a cada persona por medio de su `smartphone` en un desplazamiento acotado en el tiempo.

Fruto de todas estas consideraciones, y sin presentar todavía las evidencias de la parte experimental de esta tesis, se puede adelantar que la captación de dispositivos WiFi puede ser útil para la monitorización de personas en movimiento.

4.3.7 *Legalidad de la captación WiFi*

La monitoración de dispositivos por medio de sus comunicaciones WiFi requiere la captura de todas las tramas emitidas en una zona concreta. Esta captura se realiza gracias a una interfaz de red WiFi en modo monitor, lo que le permite capturar todas las tramas en el espacio radioeléctico independientemente de la `BSS` a la pertenezcan el par emisor/receptor de la trama y sin necesidad de que el monitor se haya autenticado o vinculado a dicha `BSS`.

De igual manera que con Bluetooth, al estar las comunicaciones realizándose en una banda de frecuencia de uso público la captura de las tramas

generadas en dichas comunicaciones no supone ninguna vulneración de ninguna ley ni reglamento.

Sin embargo, en caso de que dichas comunicaciones se encuentren protegidas mediante mecanismos de seguridad adicionales que impidan el acceso inmediato al mensaje, su captura incurrirá en un delito siempre y cuando se busque vulnerar o sortear dichos mecanismos de seguridad con la finalidad de extraer el contenido del mensaje. Este derecho, por ejemplo en España, está amparado en primera instancia por el artículo 18 de la Constitución Española y es regulado mediante las Leyes 11/1998 [164][28] y la Ley Orgánica 15/1999 [165] [29]

[164] Ley 11/1998 General de Telecomunicaciones

[165] Ley Orgánica de 15/1999 de Protección de Datos de Carácter Personal (LOPD)

En el protocolo 802.11 se establecen varios mecanismos de seguridad basados en la encriptación `WEP` y `WPA`. Sin entrar en detalles sobre las especificaciones del protocolo de estos mecanismos, ambos presentan funcionamientos similares basados en la encriptación del cuerpo del mensaje de la trama mediante claves conocidas unicamente por la pareja emisora/receptora.

Por tanto, si bien la captura de una trama encriptada no supone la vulneración de ninguna normativa, intentar desencriptar el contenido del cuerpo de una trama WiFi supone un delito pues se está quebrantando el mecanismo de seguridad por el que se ha protegido el mensaje.

Sin embargo estos mecanismos de seguridad solo son aplicados al cuerpo de la trama. La cabecera MAC se emite sin ningún tipo de mecanismo de securización aunque el cuerpo del mensaje haya sido encriptado debido a que la `BSS` haga uso de `WEP` o `WPA`. Dado que para la monitorización de dispositivos WiFi propuesta en esta tesis la parte de la trama WiFi que resulta relevante es la cabecera MAC, no se comprometen los mecanismos de seguridad que amparan la privacidad del cuerpo del mensaje. Además la fuente de datos propuesta no requiere que el cuerpo del mensaje de las tramas capturadas sea almacenado ni procesado para extraer el contenido del mismo.

De esta forma, aunque se capturen tramas pertenecientes a `BSSs` con mecanismos de seguridad, el proceso de monitorización no supone una vulnerabilidad a dichos mecanismos ya que no se busca acceder al contenido del mensaje, dado que lo relevante para la monitorización es la existencia de ese mensaje y su emisor y en ningún caso el contenido transmitido por el mismo.

Esto ampara legalmente la captura de tramas WiFi con fines de monitorización de dispositivos.

28 ↑Más conocida como Ley General de Telecomunicaciones

29 ↑Más conocida como Ley Orgánica de Protección de Datos de Carácter Persol o LOPD

4.4 NFC

NFC son las siglas de Near Field Communication que da nombre a la tecnología de comunicación inalámbrica de corto alcance y alta frecuencia basada en los estándares acogidos en ISO/IEC 14443 [61] e ISO/IEC 18092-RFID [60].

[61] ISO/IEC 14443: Identification cards — Contactless integrated circuit cards — Proximity cards

[60] ISO/IEC 18092: Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)

La comunicación inalámbrica se produce mediante el principio de inducción magnética [222] que afecta a dos antenas en forma de espira situadas dentro de sus respectivos campos magnéticos. NFC trabaja en la banda de frecuencia de los 13.56MHz perteneciente a la banda ISM, por lo que su uso no está licenciado.

[222] Foundations of electromagnetic theory

En función del número de campos magnéticos implicados en la transmisión de los datos se clasifican dos tipos de comunicación NFC:

NFC Activo donde ambas antenas generan su propio campo magnético para transmitir datos.

NFC Pasivo donde una sola antena genera su propio campo magnético (antena activa o iniciadora) y la otra emplea el principio de la modulación de carga para poder transmitir los datos.

NFC se ha implantado como estándar de facto en los smartphones para las comunicaciones de muy corto alcance o "contacto" empleadas en las tareas de identificación personal y pago electrónico [86]. El rango de alcance de las comunicaciones NFC es en el marco teórico inferior a los 20cm y en la práctica de menos de 5cm [85]. Esto provoca que los mensajes transmitidos por NFC sean difíciles de intervenir por agentes ajenos a la comunicación, pues deben de situarse entre el par emisor/receptor que se encuentra a escasos centímetros.

[86] NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]

[85] RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication

Además dado que NFC se emplea principalmente para la identificación en entornos críticos, como transacciones económicas, los smartphones complementan el hardware necesario para las comunicaciones NFC con hardware específico para el almacenamiento seguro de credenciales, como se puede ver en la Figura 4.27.

Esto implica que los identificadores no son transmitidos en plano (sin encriptar) en la comunicación, sino que se basan en pruebas criptográficas de autenticación con clave encriptada de las credenciales almacenadas. Este hecho impide la identificación unívoca del dispositivo, pues un mismo usuario puede almacenar sus credenciales en un tantos dispositivos como desee.

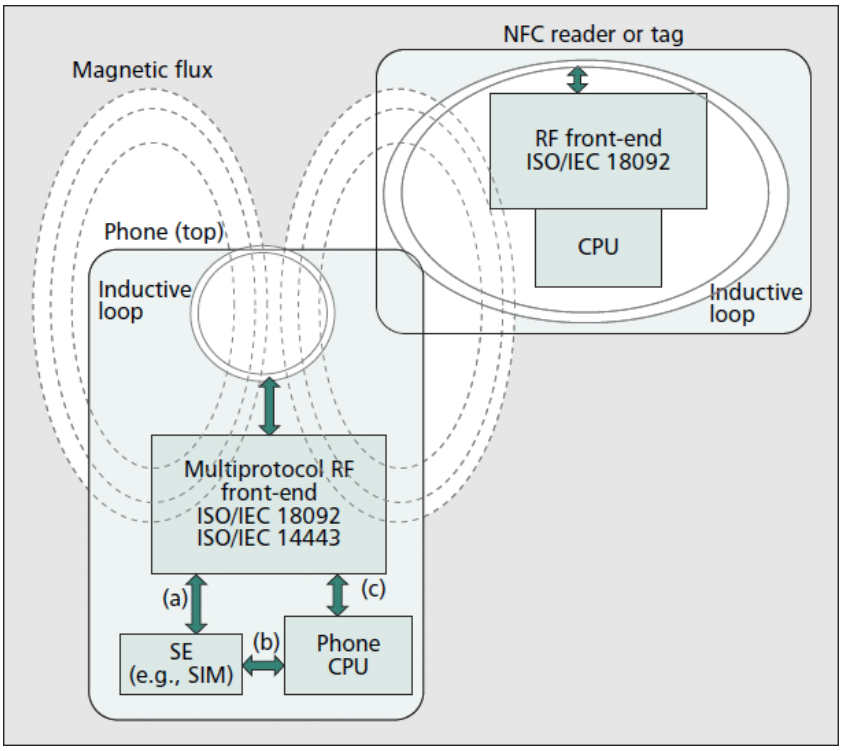


Figura 4.27
Esquema de funcionamiento de NFC en un smartphone que soporta ambas versiones del protocolo. Las líneas punteadas representan el flujo magnético que se genera entre ambas antenas de inducción. Para las transacciones seguras, el smartphone emplea un SE o Secure Element que es un hardware adicional securidazo donde se almacenan las credenciales encriptadas asociadas a identificadores, como por ejemplo los de las tarjetas bancarias.
Fuente: NFC in cell phones - Figure 1 [86]

Sin embargo, muchos dispositivos que carecen de almacenamiento seguro están incorporando NFC por su bajo coste y por su facilidad de uso para la vinculación a redes inalámbricas que requieren de autenticación como Bluetooth y WiFi, empleando la aproximación de ambos dispositivos y un intercambio de credenciales por NFC para dicho establecimiento de conexión. Esto permite por ejemplo la vinculación de un dispostivo Bluetooth a otro haciendo que se toquen para ponerlos al alcance del NFC o la autenticación al BSS de un AP WiFi siguiendo el mismo principio de "contacto".

4.4.1 *Adecuación de NFC para la monitorización de dispositivos*

Debido a la implantación creciente de sistemas de pago con los smartphones cada vez más fabricantes están incorporando NFC a sus dispositivos. Sin embargo, dado que el uso principal que se le está otorgando a esta tecnología es la de vía para transacciones económicas, los propios fabricantes están limitando el alcance para que requiera "contacto", de forma que sea imposible la captación de la comunicación NFC entre el par emisor/receptor sin interponer hardware adicional directamente entre ambos.

Esto limita la posibilidad de emplear la comunicación NFC para la monitorización de dispositivos, dado que el propio protocolo intenta imposibilitar la captación de la comunicación NFC para otros receptores.

Además, dado que NFC negocia con credenciales almacenadas en un entorno seguro, el dispositivo no se identifica de forma únivoca, si no que lo que se identifica es un token de usuario, como por ejemplo un certificado personal o asociado a una tarjeta bancaria.

Estos dos hechos imposibilitan que la captación de comunicaciones NFC pueda ser empleada para la monitorización de los dipositivos que emplean dicha comunicaciones. Si bien NFC se puede emplear para la identificación en entornos acotados, como el acceso a un edificio que requiera emplear el dispositivo como token identificador de acceso, su aplicación a la monitorización general resulta inviable en práctica debido al corto alcance de las comunicaciones y a la no identificación unívoca del dispositivo.

Es por ello que se descarta el empleo de la captación de comunicaciones NFC en el ámbito de esta tesis, pues dicha tecnología no resulta viable para tal fin.

4.5 RFID

RFID son las siglas de Radio Frequency Identification y da nombre al sistema basado en la emisión de identidades mediante ondas de radio. Se encuentra definido en numerosos estándares ISOs[30] debido a que sus origenes y atribuciones son un tanto inciertos ya que hay autores que situan los orígenes de esta tecnología en el MIT en el año 1920, otros lo atribuyen al ejército británico en 1939 o al soviético en 1945 [135]. Además, el sistema no sigue un único protocolo, sino que existen numerosas soluciones comerciales que emplean sus propias variantes de RFID para sus productos, como por ejemplo NFC que se basa en el mismo principio que de las comunicaciones RFID.

[135] Shrouds of Time: The history of RFID

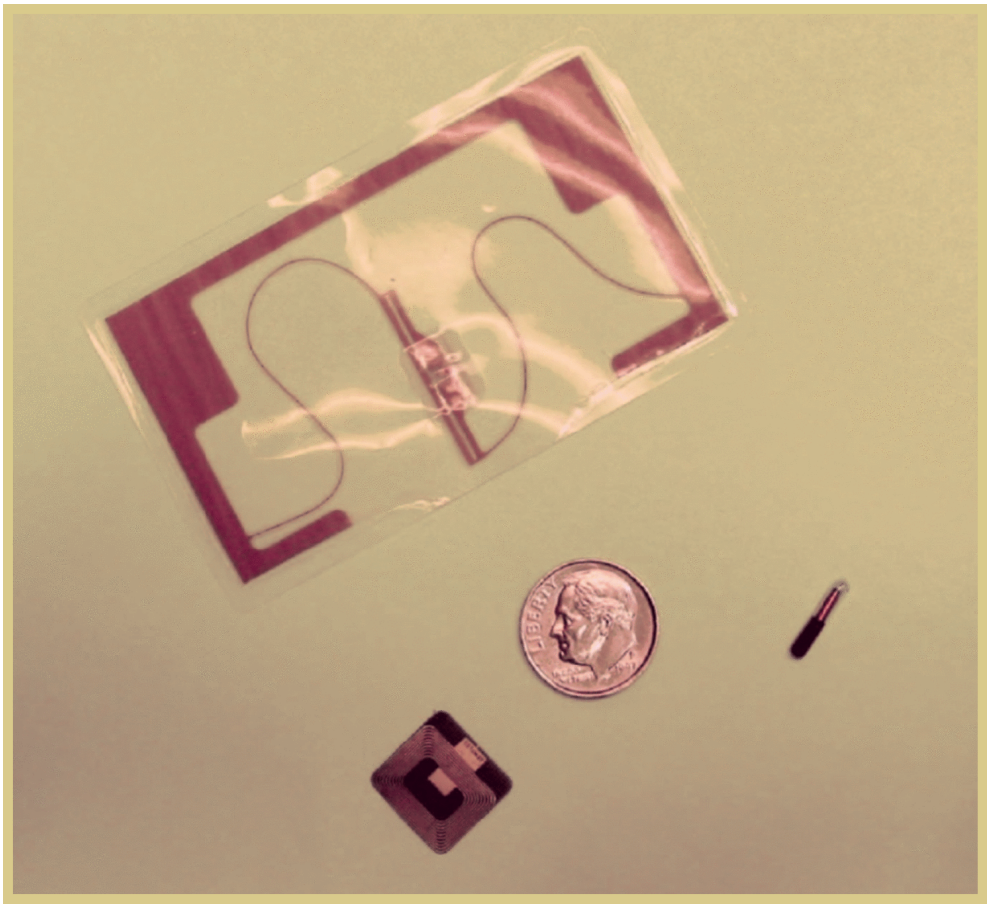


Figura 4.28
Etiquetas RFID en distintos tamaños y tipos.
Fuente: An introduction to RFID technology - Figure 1 [286]

RDID se emplea principalmente para la identificación de objetos en entornos acotados, de manera similar a los códigos de barras, pero con la ventaja de no requerir línea visual con el identificador al ser este transmitido mediante ondas de radio.

30 ↑ISO 10536,ISO 11784,ISO 11785,ISO 14443,ISO 15459,ISO 15693,ISO 15961,ISO 15962,ISO 16963,ISO 18000,ISO 18001,ISO 18046,ISO 18047,ISO 24710,ISO 24729,ISO 24730,ISO 24752,ISO 24753,ISO 24769,ISO 24770

El sistema consta de dos elementos principales: un elemento emisor que transmite el identificador y un lector que recoge e interpreta dicho identificador, cotejándolo contra una base de datos. El objeto emisor recibe el nombre transpondedor o etiqueta RFID debido a que generalmente son de pequeño tamaño (Figura 4.28).

Estas etiquetas pueden ser activas, semipasivas o pasivas, en función de si requieren fuente de alimentación interna [85] o no. Los diferentes tipos de tajertas RFID se clasifican según su esperanza de vida de la etiqueta y su alcance efectivo:

[85] RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication

RFID PASIVO Las etiquetas no poseen alimentación eléctrica y requieren de la inducción del lector sobre ellas para emitir su respuesta. Debido a esto su rango práctico es inferior a $10cm$. Si embargo su simplicidad hace que su coste sea muy barato, situándose por debajo del dolar. Dado que no requieren de alimentación su vida útil no tienen caducidad.

RFID ACTIVO Las etiquetas poseen su propia fuente de alimentación, con una duración que se situa entre 1 y 10 años. Esta batería que permite aumentar la potencia de emisión hasta un alcance efectivo de entre los 100 y 500 metros. Sin embargo, la alimentación hace que su coste aumente entre los 30 y 90 dólares, en función de su autonomía y alcance.

RFID SEMIPASIVO Las etiquetas poseen su propia fuente de alimentación, aunque esta no es empleada para la transmisión de la señal, funcionando de igual manera que una etiqueta RFID pasiva, con alcance similar.

RFID se ha convertido en un sistema ampliamente empleado debido a su versatilidad en cuanto a coste y alcance.

Las etiquetas de corto alcance se emplean comúnmente en la identificación de animales [47, 62, 148, 283], sistemas de seguimientos de libros [186], como mecanismo antirrobo de pequeños artículos comerciales [66] o como sistema de arranque de automóviles [260].

[47, 62, 148, 283] Smart RFID antenna system for indoor tracking and behavior analysis of small animals in colony cages, Tracking animals in freshwater with electronic tags: past, present and future, Animal situation tracking service using RFID, GPS, and sensors, A complete farm management system based on animal identification using RFID technology

[186] Privacy and security in library RFID: Issues, practices, and architectures

[66] Anti-theft device with alarm screening

[260] Near field communication (NFC) in an automotive environment

Las etiquetas de largo alcance se utilizan en seguimientos de contenedores de mercancías [271], equipajes, vehículos de transporte o para el cobro de peajes en carretera [260].

[271] Anti-theft method for detecting the unauthorized opening of containers and baggage

[260] Near field communication (NFC) in an automotive environment

4.5.1 *Adecuación de RFID para la monitorización de dispositivos*

[92, 236] MASCAL: RFID Tracking of Patients, Staff and Equipment to Enhance HospitalResponse to Mass Casualty Events. RFID tracking of anesthesiologist and patient time

Si bien RFID se ha empleado en numerosas ocasiones con éxito para monitorizar personas como por ejemplo pacientes en entornos hospitalarios [92, 236], su uso requiere de la implantación de una etiqueta RFID en cada paciente. Esto supone una ventaja en entornos donde la monitorización sea crítica, pero el requerimiento un elemento externo no permite su implantación de forma masiva e inadvertida a las personas a ser monitorizadas.

La mayoría de los dispositivos de los dispositivos inteligentes, en los que se centra esta tesis, no hacen uso de etiquetas RFID en sus elementos hardware, por lo que no pueden ser detectados mediante el empleo de esta tecnología. Esto hace que se desestime las comunicaciones RFID para la captación de dispositivos para la monitorización de personas y vehículos.

Sin embargo, aunque sea desestimado para el propósito de esta tesis, el empleo de etiquetas RFID es actualmente uno de los sistemas de monitorización más extendidos a lo largo del mundo para la monitorización crítica tanto de elementos, personas y mercancías.

4.6 TELEFONÍA INALÁMBRICA

Los `smartphones` no dejan de ser dispositivos telefónicos, por la tanto se considera que su principal tecnología de comunicación inalámbrica empleada es la comunicación telefónica móvil, que emplean tanto para la transmisión de voz como de datos.

Una red telefónica móvil está formada por dos elementos principales, una red de comunicación formada por grandes antenas repartidas por la geografía y los propios terminales que acceden a dicha red. Tanto los terminales como la red de comunicación son emisores-receptores de ondas electromagnéticas con frecuencias que oscilan entre los $900MHz$ y los $2000MHz$ en función de la tecnología empleada.

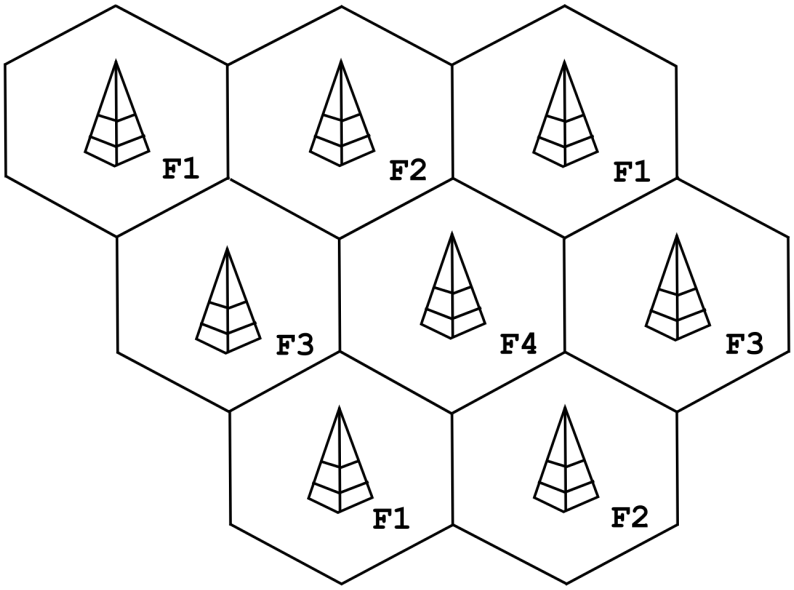


Figura 4.29
Ejemplo de red celular empleada en las comunicaciones móviles.
Fuente: Wikimedia Commons

La red de comunicación se compone por numerosas antenas repartidas en el área siguiendo una distribución normalmente hexagonal. Se denomina célula o hexágono cada área cubierta por una antena (Figura 4.29). En cada célula hay una estación base, que es el nombre que recibe la antena con amplidad necesaria para emitir y recibir en ese hexágono o célula. En función de la tecnología que implemente la estación base se denomina `BTS/BSC` para `2G`, nodo `B/RNC` para `3G` o `e-nodoB` para `4G`. En función de la tecnología empleada, se emplean un bandas de frecuencias diferentes [261]:

[261] Principles of mobile communication

2G (o GSM) emplea las bandas de frecuencia de $900MHz$ y $1800MHz$.

3G (o UMTS) utiliza las bandas de frecuencia de $900MHz$ y $2GHz$.

4G (o LTE) funciona en las bandas de frecuencia de $800MHz$, $1500MHz$ $1800MHz$ y $2600MHz$.

Las bandas de frecuencias empleadas son licencialas, por lo que su uso se encuentra restringido, requiriendo autorización gubernamental. Estas licencias son otorgadas a empresas de telecomunicaciones, que son los únicos organismos con permiso para emplazar y establecer estaciones bases.

Cada terminal se encuentra identificado por un código `IMEI`[31] que lo identifica de forma exclusiva a nivel mundial y es trasmitido por el terminal en sus comunicaciones con las estaciones bases.

4.6.1 *Adecuación de las comunicaciones móviles para la monitorización de dispositivos*

Aunque las empresas de telecomunicaciones pueden emplear las comunicaciones móviles para monitorizar dispositivos conectados a sus antenas [24, 239, 246], unicamente pueden hacer uso de este sistema cuando una orden judicial lo autoriza según regula el Real Decreto de 14 de Septiembre de 1882 [220] y la Ley 11/1998 General de Telecomunicaciones [164].

[24, 239, 246] Cellular/GPS system for vehicle tracking, Vehicle tracking system using cellular network, Vehicle tracking system

[220] Real Decreto de 14 de Septiembre de 1882

[164] Ley 11/1998 General de Telecomunicaciones

Además, dado que el código `IMEI` tiene que estar vinculado a un ente jurídico, ya sea una persona o una institución, se vulneran los derechos de privacidad amparados por la Ley Orgánica de 15/1999 de Protección de Datos de Carácter Personal (LOPD) [165], que vela con especial cuidado los códigos que permiten identificar a personas, como los `IMEI` o las matrículas de vehículos.

[165] Ley Orgánica de 15/1999 de Protección de Datos de Carácter Personal (LOPD)

Los propios dispositivos móviles puede posicionarse gracias a la triangulación a las distintas estaciones bases en el rango, ya que esta información puede ser recogida por el terminal sin problema. Sin embargo identificar donde se encuentra un dispositivo móvil concreto sin emplear las estaciones base reguladas por las empresas de telecomunicaciones, requiere simular ser una estación base a la que el dispositivo se conecta. Aunque existen investigadores que han realizado estas simulaciones y han logrado aproximar la posición de un teléfono móvil [65] sin emplear las infraestructuras de las empresas de telecomunicaciones, esto vulnera la legalidad vigente siendo constitutivo de un delito.

[65] IMSI-catch me if you can: IMSI-catcher-catchers

Si bien, por tanto, la tecnología de comunicación celular telefónica provee de todos los mecanismos para la monitorización de dispositivos la legislación vigente regula que dicho mecanismo de monitorización sólo puede ser empleado por las empresas de telecomunicaciones licenciadas y amparadas siempre bajo una orden judicial.

Esto implica, que dicho sistema de comunicaciones sea desestimado para el ámbito de esta tesis doctoral.

31 ↑International Mobile Station Equipment Identity.

5

PROPUESTA DE MONITORIZACIÓN POR MEDIO DE CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS

Mi proyecto, mi trabajo, mi responsabilidad.
Tengo que ser yo. Cualquier otro podría hacerlo mal.

— Mordin Solus (Mass Effect III)

En este capítulo se aborda el prototipo de sistema de monitorización que ha sido empleado para estudiar la viabilidad de la fuente de datos propuesta en esta tesis. En primer lugar, se presenta su funcionamiento teórico indicando los requisitos y retos que tiene que abordar. Posteriormente, estos fundamentos teóricos son materializados en el diseño y desarrollo de un prototipo funcional. Este prototipo funcional tiene tanto componentes hardware como software y aglutina las herramientas empleadas para la realización de los estudios aplicados de esta tesis. Este prototipo es fruto de la materalización de la investigación realizada.

Índice del capítulo

5.1 FUNDAMENTOS DE LA MONITORIZACIÓN POR CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS

En esta sección se detallan los fundamentos de la monitorización por medio de la captación de comunicaciones inalámbrica. Para ello se describirán varios conceptos que serán ampliamente empleados en el transcurso de la tesis.

5.1.1 *Nodo de monitorización*

Al igual que otros sistemas de monitorización presentados en la Sección 3.1 la monitorización por medio de la captación de comunicaciones inalámbricas requiere de un hardware detector emplazado en la zona a monitorizar denominado, en el ámbito de esta tesis, **nodo de monitorización** o simplemente **nodo**. Este **nodo** está compuesto tanto por componentes hardware (Sección 5.4) como software (Secciones 5.5 y 5.6). El principal cometido de este **nodo** es el de la **detección** de dispositivo mediante sus comunicaciones inálámbricas.

5.1.2 *Detección de dispositivos*

Para la **detección** de los dispositivos, el **nodo** está constantemente en la búsqueda de dispositivos Bluetooth y/o capturando tramas WiFi según se ha detallado en las Secciones 4.2 y 4.3 respectivamente. Para ello está dotado de tarjetas de red WiFi y Bluetooth cuyo alcance marca el radio de acción máximo del **nodo** (Figura 5.1).

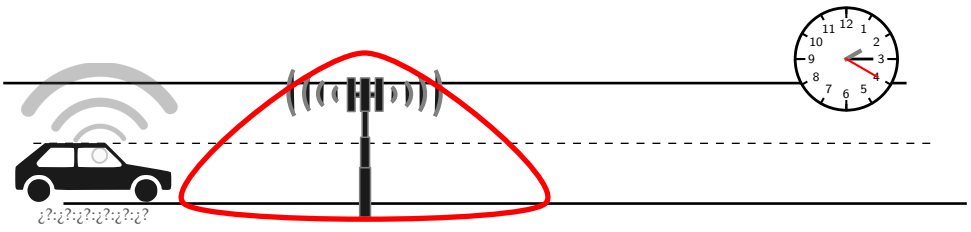
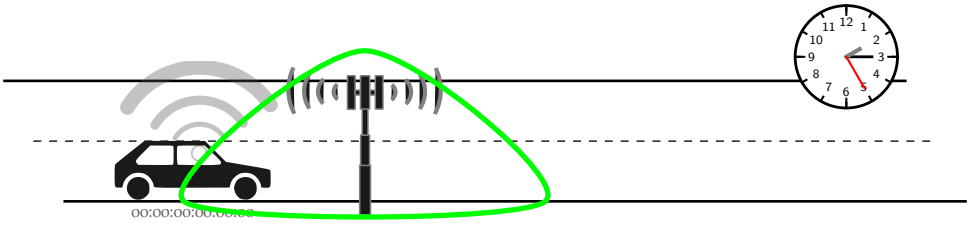


Figura 5.1
Monitorización por medio de captación de comunicaciones inalámbricas:
El dispositivo se encuentra fuera del radio de acción del **nodo**, por lo que no es detectable.

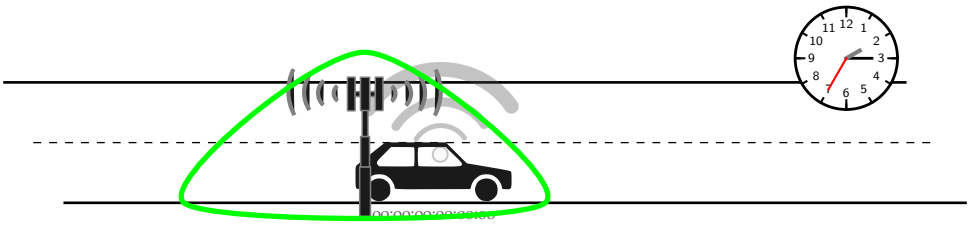
Una vez que un dispositivo entra dentro del radio de acción de la antena es detectable por el **nodo**, el cual lo identifica mediante su MAC gracias a los procedimientos de detección e identificación descritos en las Secciones 4.2.2 y 4.3.5 para Bluetooth y WiFi respectivamente. El **nodo** dispone de un reloj interno con el que es capaz de anotar el instante de tiempo con el que se ha realizado la **detección**. (Figura 5.2).



```
Información registrada por el nodo:
02:15:25 - Detectado 00:00:00:00:00:00
```

Figura 5.2
Monitorización por medio de captación de comunicaciones inalámbricas:
Un dispositivo ha sido detectado, el **nodo** identifica al dispositivo por su dirección MAC y anotar el instante de tiempo en el que el dispositivo ha sido detectado por primera y última vez.

Mientras el dispositivo se encuentre dentro del radio de acción del **nodo**, será detectado en distintos instantes a lo largo del tiempo, ya sea mediante la recepción de nuevos paquetes FHS o la captura de tramas WiFi. En cada nuevo escaneo y detección, el **nodo** es consciente del instante de tiempo en el que el dispositivo a sido detectado (Figura 5.3).



```
Información registrada por el nodo:
02:15:25 - Detectado 00:00:00:00:00:00
02:15:27 - Detectado 00:00:00:00:00:00
02:15:30 - Detectado 00:00:00:00:00:00
02:15:34 - Detectado 00:00:00:00:00:00
```

Figura 5.3
Monitorización por medio de captación de comunicaciones inalámbricas:
Una vez que el dispositivo se aleja del alcance del **nodo**, se deja de actualizar la información de la última detección.

En el ámbito de esta tesis, se denomina **detección** a la información obtenida por un **nodo** cuando un dispositivo entra en sus inmedicaciones. Esta información recoge tanto el identificador del **nodo**, la dirección MAC del dispositivo, así como el instante de tiempo en el que ha sido detectado.

La dirección MAC provee de información sobre el fabricante del dispositivo, tal y como se ha presentado en las Secciones 4.2.2 y 4.3.5 para Bluetooth y WiFi respectivamente. Las direcciones MAC se constituyen siguiendo el formato EUI-48, mediante el cual un organismo certificador entrega rangos de MAC a cada fabricante. Esta asignación es pública [129] y puede ser empleada para obtener el fabricante de un dispositivo concreto detectado, aportando esta información a la **detección** de dicho de dispositivo.

[129] IEEE BT (2013). The IEEE public BT OUI listing. Retrieved Dec 30, 2013, from http://standards.ieee.org/develop/regauth/oui/oui.txt

En función de la tecnología que haya sido empleada para la detección se puede completar la información de la **detección**.

Información adicional en Bluetooth

En el caso de Bluetooth, como se ha abordado en la Sección 4.2.3.5, el paquete `FHS` obtenido de retorno en la búsqueda contiene el campo `Class of Device` que proporciona información sobre la naturaleza del dispositivo. Esta naturaleza, puede ser inspeccionada y almacenada como información adicional a la **detección**. En el Anexo A.1 se presentan la clasificación de dispositivos Bluetooth en función de su `Class of Device` y `Subclass of Device`.

Información adicional en WiFi

En el caso de las comunicaciones WiFi, las tramas presentadas en la Sección 4.3.5, no contienen ninguna información sobre la naturaleza del dispositivo. Sin embargo, para cada trama capturada mediante el modo monitor es posible conocer el indicador de fuerza de la señal recibida (`RSSI`[1]). En el caso de Bluetooth, el protocolo establece que esta información sólo puede ser conocida en caso de establecer una vinculación[2]

El protocolo mediante el cual se transmite esta información es conocido como RadioTap[3] y no forma parte del estándar 802.11. Las cabeceras Radiotap proveen de información adicional que suplementa a la información de la trama 802.11 capturada.

```
▶ Frame 6: 298 bytes on wire (2384 bits), 298 bytes captured (2384 bits)
▼ Radiotap Header v0, Length 36
    Header revision: 0
    Header pad: 0
    Header length: 36
  ▶ Present flags
    MAC timestamp: 1204080838
  ▶ Flags: 0x10
    Data Rate: 6,0 Mb/s
    Channel frequency: 2412 [BG 1]
  ▶ Channel flags: 0x00c0, Orthogonal Frequency-Division Multiplexing (OFDM), 2 GHz spectrum
    SSI Signal: -59 dBm
  ▶ RX flags: 0x0000
    SSI Signal: -59 dBm
    Antenna: 0
▶ 802.11 radio information
▶ IEEE 802.11 Beacon frame, Flags: ........C
▶ IEEE 802.11 wireless LAN management frame
```

Figura 5.4
Cabecera Radiotap incorporada sobre la trama 802.11 capturada. La Fuerza de la señal recibida o RSSI es transmitida mediante dicha trama.

La intensidad `RSSI` es una escala de referencia (en relación a 1mW) para medir el nivel de potencia de las señales recibidas por un dispositivo en redes inalámbricas. Esta escala emplea el valor 0 como central representando odbM. Aunque en el aspecto teórico puede darse el caso de medirse valores positivos,

1 ↑Received Signal Strength Indicator
2 ↑Aunque hay fabricantes que han modificado el protocolo para permitir conocer el `RSSI` de una comunicación Bluetooth sin necesidad de establecer conexión con el dispositivo, aunque eso vaya en contra de las especificaciones del protocolo.
3 ↑http://www.radiotap.org/

en la práctica generalmente la escala se expresa dentro de valores negativos, cuanto más negativo mayor pérdida de la señal. Se hace necesario destacar que el RSSI indica la intesidad recibida, no calidad de la señal, ya que esta última se determinaría contrastando la intensidad de la señal respecto de la relación señal/ruido (Eb/No).

Según los valores de esta escala muchos fabricantes suelen implementar en sus interfaces de usuario "rayas de conexión", habitualmente en los cortes de $-30dBm$, $-57dBm$, $-70dBm$, $-80dBm$ y $-90dBm$. El protocolo WiFi establece que el RSSI mínimo para la conectividad básica se sitúa en torno a los $-80dBm$ e inoperable a partir de los $-100dBm$.

El valor RSSI de una transmisión inalámbrica implica a varios factores, entre los que se encuentran la potencia de transmisión (TX), las ganancias de los pares de antenas emisora y receptora, las perdidas en los cables conectados a las antenas, la frecuencia de emisión y la distancia entre ambas antenas.

Debido a que el RSSI de una transmisión inalámbrica WiFi es inversamente proporcional a la distancia entre el emisor y el receptor, en los últimos años han surgido estudios que emplean distintas señales WiFi emisoras o APs para triangular posiciones. Para ello se basan en la intensidad RSSI con la que se detectan varios APs en un punto geográfico concreto [167, 182, 206, 217]. Para ello, los distintos APs tienen que estar determinados mediante un par de coordenadas geográficas, y además, se requiere distancias de calibración, con lo que obtener los valores RSSI de cada APs a distintas distancias determinadas (Figura 5.5)

[167, 182, 206, 217] A real-time indoor WiFi localization system utilizing smart antennas, Mobile device and method for determining location of mobile device, Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points—A Trilateration Approach, Cooperation among smartphones to improve indoor position information

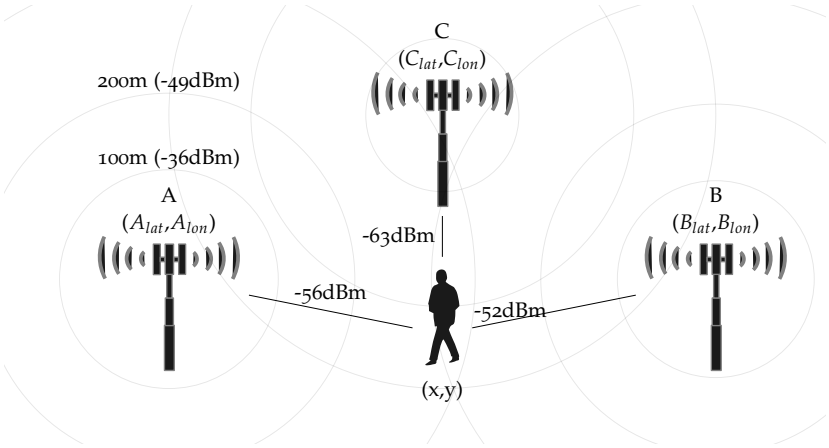
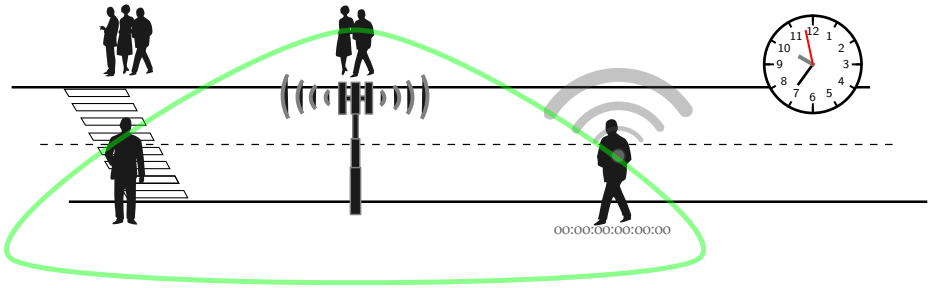


Figura 5.5
Escenario de triangulación mediante WiFi, donde tres puntos de acceso (A,B,C) están emitiendo tramas Beacon anunciando una BSS. En un punto (x,y) cada APs se detecta con un RSSI determinado. Cada AP se encuentra posicionado con unas coordenadas concretas, y además, se han realizado mediciones del RSSI a distancias del AP de referencia medidos como distancias, en la figura sólo se detallan las marcas para el AP A.

La incorporación del RSSI a la información de la **detección** puede servir para varios propósitos, como filtrar dispositivos detectados por el **nodo** con valores RSSI muy bajos (y por tanto se encuentran emitiendo demasiado lejos al **nodo**) o realizar triangulaciones como la que se presenta en la Figura 5.5.

5.1.3 *Paso de dispositivos*

Un mismo dispositivo es detectado en múltiples ocasiones mientras se encuentra en el rango del **nodo**. Todo este conjunto de detecciones se resumen en un **paso**, que en el ámbito de esta tesis contiene el marco temporal de la primera y última vez que un dispositivo ha sido detectado (Figura 5.6). Estos dos instantes de tiempo, a lo largo de la tesis serán denominados `tinicio` y `tfin`, correspondientes al instante de tiempo de la primera y ultima detección respectivamente.

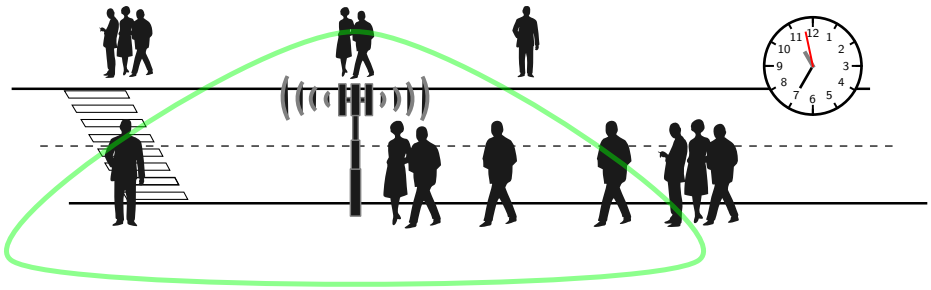



```
Información registrada por el nodo:
00:00:00:00:00:00 - ...10:34:20 , ...10:34:26
4A:12:45:11:CA:1D - ...10:29:49 , ...10:36:12
```

Figura 5.6
Monitorización por medio de captación de comunicaciones inalámbricas:
Un mismo dispositivo es detectado sucesivamente mientras se encuentra en el alcance, el **nodo** actualiza el tiempo de última visita. En este caso, el dispositivo con MAC 00:00:00:00:00 lleva varios segundos sin ser detectado por el **nodo**. Es posible que sea debido a que se haya alejado del **nodo** o que a este no le haya dado tiempo de volver a detectarlo.

De esta forma, es posible disponer de información sobre cuanto tiempo ha estado en las inmediaciones del **nodo** cada dispositivo. Debido a que los dispositivos son detectables por intervalos de tiempo determinados, uno de los problemas que plantea este sistema, es establecer cuanto tiempo es requerido sin tener contacto con un dispositivo para saber si se ha alejado en las inmediaciones o si aún no le ha dado tiempo a volver a ser detectable por el **nodo**

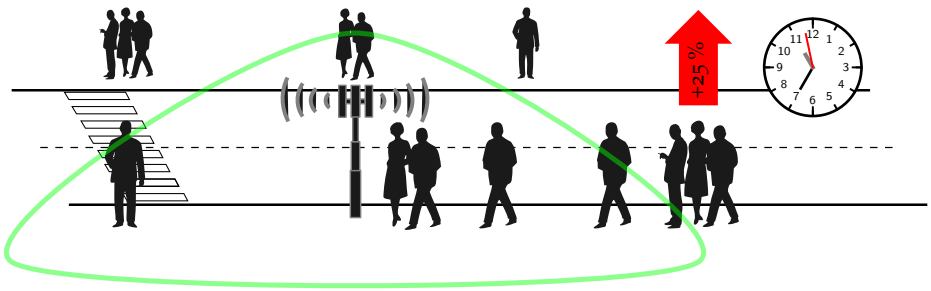
Una vez el dispositivo se determina que se ha alejado del **nodo**, este dispone de información sobre la duración de la estancia del dispositivo. En base a la información de la estancia de todos los dispositivos en un marco temporal, el sistema puede aproximar cuantos dispositivos han pasado por las inmediaciones en dicho marco temporal (Figura 5.7).



```
Información registrada por el nodo:
6C:4D:73:B7:6C:52 - ...10:34:20 , ...10:34:26
B0:48:1A:CF:37:B5 - ...10:12:52 , ...10:31:09
BC-C1-31-93-82-9C - ...10:44:07 , ...10:44:52
2C-6A-42-B2-EE-8A - ...10:50:20 , ...10:57:23
25-18-F2-F6-E4-A6 - ...10:58:13 , ...11:18:30
C3-23-9A-17-D2-81 - ...11:08:20 , ...11:12:18
49-DF-B1-74-CE-4D - ...11:11:38 , ...11:23:26
4A:12:45:11:CA:1D - ...11:29:49 , ...11:34:12
Número de pasos en última hora: 155 dispositivos.
Duración media de la estancia: 5 minutos.
```

Figura 5.7
Monitorización por medio de captación de comunicaciones inalámbricas:
El **nodo** dispone de los marcos temporales de todos los dispositivos que han sido detectados en las inmediaciones. Esta información le permite inferir cuando dispositivos se encontraban cerca en momento concreto.

El sistema es capaz de calcular la variación del número de **pasos** dispositivos en un marco temporal concreto respecto a lo normal, nutriéndose de la información histórica detectada y considerando lo normal a lo promedio detectado por dicho **nodo** en circunstancias similares a las marco considerado. Estas circunstancias puede venir determinadas por la hora del día, el día de la semana, algún evento o cualquier otro criterio establecido (Figura 5.8).




```
Información registrada por el nodo:
..:..:..:..:..:.. - .....:..:.. , .....:..:..
Número de pasos en última hora: 194 dispositivos.
```

Figura 5.8
Monitorización por medio de captación de comunicaciones inalámbricas:
Basándose en información histórica, el sistema puede calcular la variación de un estado respecto a lo considerado normal. Por ejemplo, en este caso, se están detectando un 25 % más de los habitual en dicho escenario.

Los **pasos** de varios dispositivos pueden ser agrupados a lo largo de un intervalo de tiempo acotado. Por ejemplo, se puede contabilizar el número de dispositivos que han pasado a cada hora o cada cierto de minutos por el **nodo**. En esta tesis, se considera que un **paso** sólo se puede contabilizar en un único intervalo de tiempo, aunque abarque más tiempo su estancia. Para determinar la pertenencia a un intervalo u otro, se considera el `tinicio` del paso. Así, para que el **paso** pertenezca al intervalo, debe haberse iniciado posteriormente al comienzo de dicho intervalo (Figura 5.9).

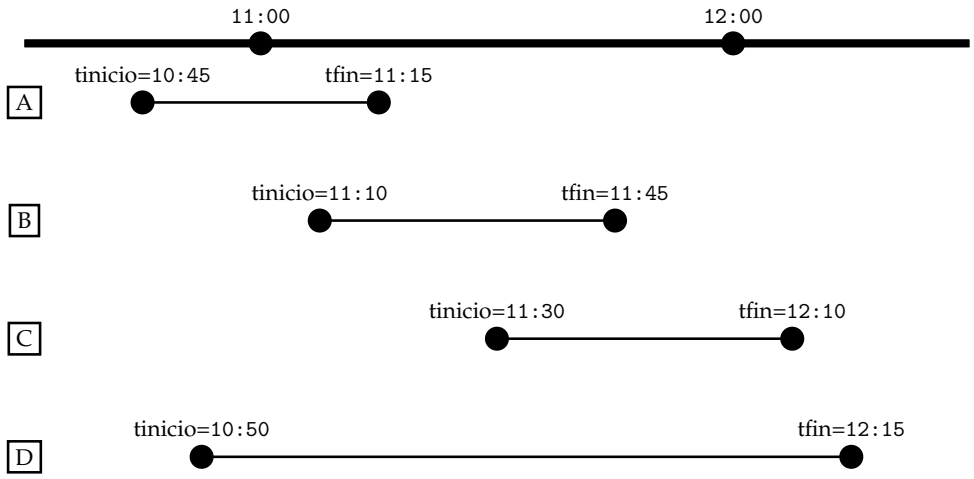


Figura 5.9
Agrupamiento de pasos en base a un intervalo.
Se agrupan los pasos de los dispositivos A,B,C,D enmarcados temporalmente por sus respectivos `tinicio` y `tfin` en intervalos de una hora para las 11 horas. De esta forma, se contabilizará el número de dispositivos que han pasado desde las 11:00 a las 12:00 por el nodo
En el caso de A, su `tinicio` es menor de las 11:00, por lo que su visita no contabilizaría para el intervalo de las 11:00, aunque su `tfin` sea inferior a valor del siguiente intervalo. Este paso sería por tanto contabilizado para el intervalo de las 10:00, no para el de las 11:00.
En el caso de B, su `tinicio` es mayor que el valor del intervalo, y su `tfin` es menor que el valor del siguiente intervalo, por lo que se contabilizaría para el intervalo de las 11:00.
En el caso de C, su `tinicio` es mayor de las 11:00, aunque su `tfin` es mayor que el siguiente valor del intervalo. Sin embargo, el paso se inició dentro del intervalo de las 11:00, por lo que se contabilizaría para este intervalo.
En el caso de D, su `tinicio` es menor del valor del intervalo y el `tfin` es mayor que el siguiente valor del intervalo. Este paso por tanto, abarca tanto los intervalos de las 10:00, como el de las 11:00 y las 12:00. Sin embargo, como se inició en el intervalo de las 10:00, sólo se puede contabilizar en este intervalo.

5.1.4 *Paso de dispositivos únicos*

Al hacer el recuento, se puede adicionalmente restringir que un mismo dispositivo sólo pueda ser considerado una única vez para ese intervalo. De esta forma, varias visitas espaciadas por el mismo dispositivo, se contemplaría una única vez para el conteo. Este recuento, en el ámbito de esta tesis, se denomina **pasos únicos** (Figura 5.10).

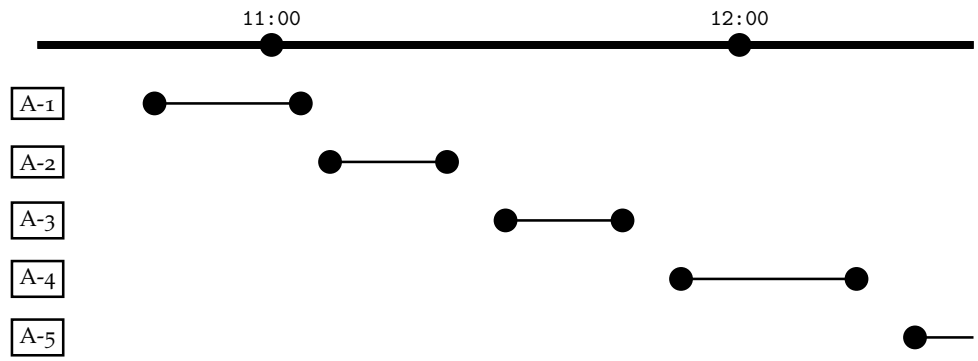


Figura 5.10
Agrupamiento de pasos únicos en base a un intervalo.
Se agrupan los pasos únicos en intervalos de una hora para las 11 horas.
El primer paso del nodo A (A-1) no se contabilizaría para el intervalo de las 11:00 a las 12:00, porque pertenece al intervalo anterior de las 10:00, ya que se originó en ese intervalo.
El segundo paso del nodo A (A-2) si sería contabilizado para el conteo de pasos únicos del intervalo de las 11:00, ya que se inició dentro de dicho intervalo.
El tercer paso (A-3) y cuarto paso (A-4) no serían contados para el intervalo de las 11:00, ya que el dispositivo A ya ha sido contabilizado una vez para este intervalo. El paso (A-4), tampoco sería contabilizado para el intervalo de las 12:00, ya que se inició antes de ese intervalo de tiempo.
El quinto paso (A-5) sería contabilizado para el intervalo de las 12:00, ya que es el primer paso que se ha detectado iniciado posteriormente a este intervalo.

Contabilizar los pasos es muy útil para establecer el volumen o flujo de dispositivos por la zona en un intervalo de tiempo (Sección 3.2.2).

5.1.5 *Dispositivos simultáneos*

Aunque el conteo de pasos es muy útil, y es el tipo de métrica más proporcionada por los sistemas de monitorización actuales, el sistema propuesto puede ofrecer también la ocupación (Sección 3.2.3) y detección de presencia (Sección 3.2.1).

Ambos conceptos, en el ámbito de esta tesis, se recogen bajo el término de conteo de dispositivos **simultáneos** en un instante de tiempo dado. Al contrario que los **pasos** que se enmarcan en una ventana temporal definida entre el intervalo y el siguiente, en los dispositivos simultáneos sólo se requiere un instante de tiempo. De esta forma, el sistema puede responder a cuantos dispositivos estaban siendo simultáneamente detectados en un instante de tiempo concreto (Figura 5.11).

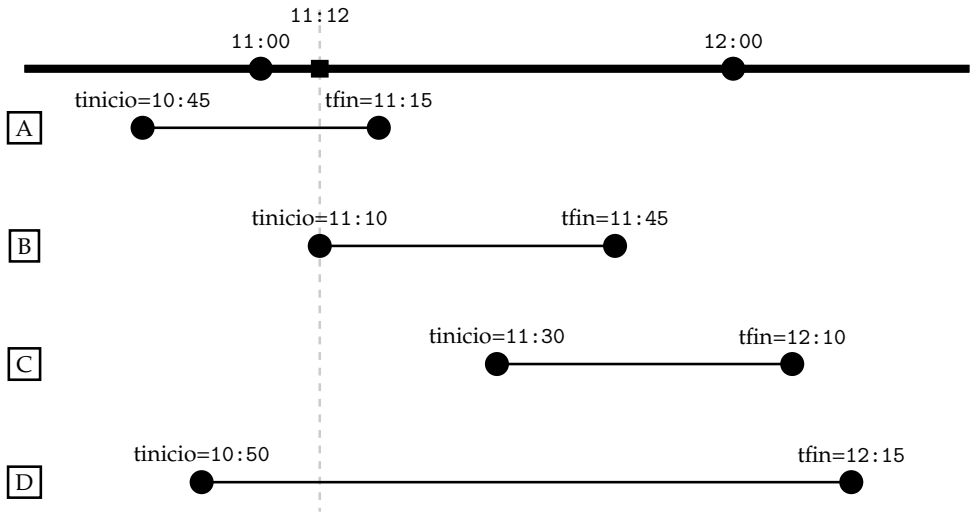
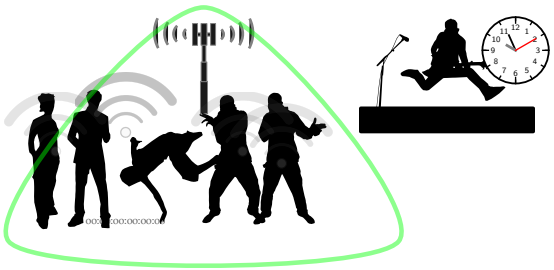


Figura 5.11
Conteo de dispositivos simultáneos en un instante de tiempo.
Para que un **paso** de un dispositivos se considere para el cálculo, su `tinicio` y `tfin` han de contener el instante de tiempo del muestreo. De esta forma, en el ejemplo, solamente los dispositivos A,B y D estaban en el instante de tiempo de las 11:12 siendo detectados de forma simultánea.

Así, una de las principales fortalezas del sistema de monitorización, es que no solamente permite consultar el número de dispositivos que han pasado en un intervalo de tiempo, sino que también es posible reconstruir el número de dispositivos que estaban de forma simultánea en las inmediaciones del **nodo**. Esto lo convierte en una herramienta muy útil para controlar la presencia y densidad de dispositivos detectados en numerosos escenarios, como tanto el tránsito de personas como de vehículos desplazándose por la calle (Figura 5.6) o en una sala o habitación (Figura 5.12).



```
Información registrada por el nodo:
00:00:00:00:00:00 - ...09:15:34 , ...10:55:54
4A:12:45:11:CA:1D - ...09:11:12 , ...10:12:21
78:43:A1:B4:74:A3 - ...09:43:44 , ...10:50:17
..:..:..:..:..:.. - .....:..:.. , .....:..:..
Número de pasos en última hora: 71 dispositivos.
Duración media de la estancia: 55 minutos.

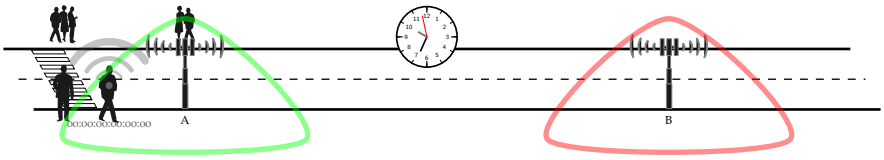
Número de dispositivos simultáneos actualmente: 55 dispositivos
```

Figura 5.12
Monitorización por medio de captación de comunicaciones inalámbricas: Ejemplo aplicado a la detección de personas.

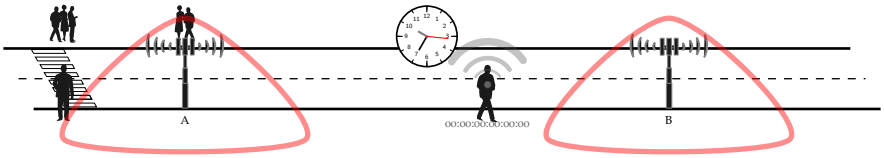
Esto permite que el sistema de monitorización propuesto pueda proveer información en una gran cantidad de escenarios donde se encuentren dispositivos susceptibles de ser detectados.

5.1.6 *Trazabilidad de dispositivos*

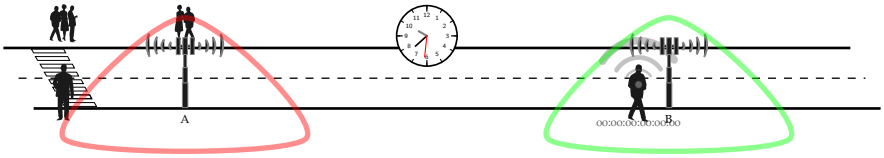
Cómo se ha presentado en la Sección 2.2.3.1 una SmartCity puede nutrirse de una red de sensores. La propuesta del sistema de monitorización, no debe de contemplar únicamente **nodos** independientes, sino **nodos** emplazados cerca de ellos que permita cruzar su información o **pasos**. En el ámbito de esta tesis se denomina **traza** al evento que relaciona el **paso** de un dispositivo por las inmediaciones de dos o más **nodos** en un periodo de tiempo proporcional a la distancia entre ambos. (Figura 5.13)



(a) 10:34:58



(b) 10:35:16



(c) 10:38:31

```
Información registrada por el sistema sobre el dispositivo 00:00:00:00:00:00:
A,00:00:00:00:00:00 - ...10:34:20 , ...10:35:14
B,00:00:00:00:00:00 - ...10:37:54 , ...10:37:54
A,4A:12:45:11:CA:1D - ...10:29:49 , ...10:34:12
```

Figura 5.13
Ejemplo de una red de **nodos** para la trazabilidad de los movimientos de los dispositivos inteligentes.

Con una red de **nodos** amplia como la que se presenta en la Figura 5.14, se puede extraer información sobre como los dispositivos se mueven entre las distintas zonas de la ciudad, así como el tiempo promedio que lleva en promedio hacer dichos desplazamientos empleando para ello las magnitudes para el estudio de tráfico presentadas en la Sección 3.2.

Figura 5.14
Ejemplo de red de **nodos** sensores desplegado en un entorno urbano.

La existencia de una red compleja permite además realizar estudios sobre la predilección entre distintas rutas. Por ejemplo, en el escenario reflejado en la Figura 5.15 un dispositivo determinado ha sido detectado en el nodo sensor *A*. Es esperable que en el futuro, ese dispositivo sensor sea detectado por alguno de los otros nodos sensores (*B* y *C* en este escenario)[4]. El nodo sensor por el que haya sido detectado, determinará la elección de la ruta del dispositivo.

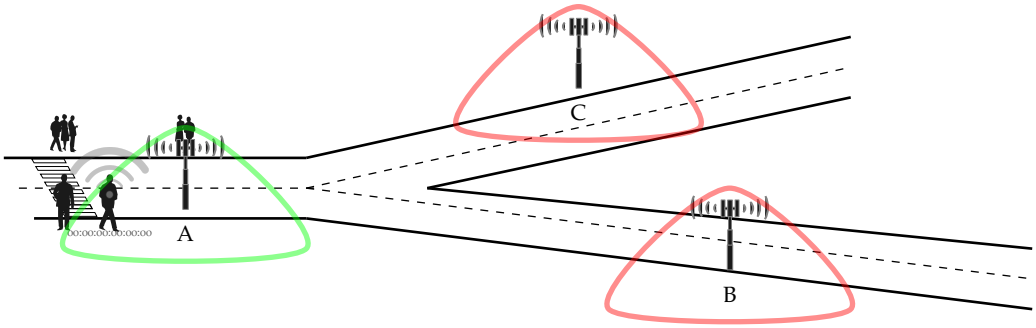
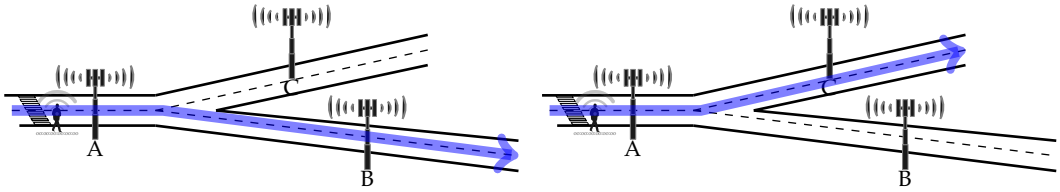


Figura 5.15
Ejemplo de una red de más de dos **nodos** para la trazabilidad de los movimientos de los dispositivos inteligentes. Es esperable que el dispositivo detectado en *A* sea posteriormente detectado en *B* o *C*, dependiendo de la ruta que dicho dispositivo haya seguido.

El poder determinar la ruta para un dispositivo determinado es poco relevante, pero cobra mayor importancia cuanto más dispositivo sean detectados y trazados. Por ejemplo, la Figura 5.16 muestra las dos posible rutas que

4 ↑Obviamente también es probable que el dispositivo no sea detectado en ninguno de los otros nodos sensores, ya que la cobertura en el escenario teórico propuesto no es absoluta, de igual manera que en la práctica será muy difícil alcanzara una cobertura total.

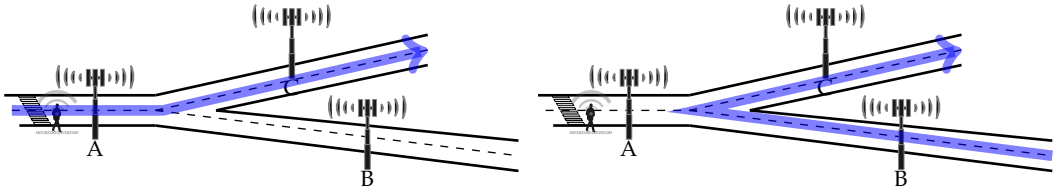
pueden ser tomadas desde A, siendo Traza $A \rightarrow C$ y Traza $A \rightarrow B$ respectivamente. En este caso, se ha determinado el origen para determinar sus posibles destinos.



(a) Traza $A \rightarrow B$ (b) Traza $A \rightarrow C$

Figura 5.16
Ejemplo de trazas con origen determinado, donde el sujeto de estudio es el destino de los dispositivos detectados en un nodo sensor determinado.

De igual manera las trazas permiten estudiar para un destino determinado cuales han sido los nodos sensores de origen, es decir, determinar donde se ha originado el movimiento. En la Figura 5.17 se presentan las trazas resultantes al determinar un destino, siendo en este escenario las trazas $A \rightarrow C$ y $B \rightarrow C$.



(a) Traza $A \rightarrow C$ (b) Traza $B \rightarrow C$

Figura 5.17
Ejemplo de trazas con destino determinado, donde el sujeto de estudio es el origen de los dispositivos detectados en un nodo sensor determinado.

Esta forma de notar las trazas es de vital importancia para comprender los normalizados que se producirán en los agrupamientos que impliquen varios dispositivos detectados, con el fin de determinar los porcentajes de predilección de rutas. De esta forma, no es lo mismo normalizar respecto al origen que al destino. En el estudio 5.11.1 se presenta un ejemplo de la diferencia entre el normalizado de las matrices entrada/salida[5] respecto a origen o destino.

5 ↑Esta estructura de datos es presentada en profundida en la Sección 5.11.2.

5.2 REQUISITOS DEL SISTEMA DE MONITORIZACIÓN

Presentado los fundamentos en los que se sustenta el sistema de monitorización, se hace necesario establecer que requerimientos se van a imponer al prototipo funcional que implementa dicho sistema de monitorización y que será empleado como herramienta para los estudios presentados en esta tesis.

Se presentan en primer lugar los retos a los que tiene que hacer frente el prototipo de sistema, los requisitos funcionales impuestos al sistema y finalmente se presentan los componentes funcionales del mismo.

5.2.1 *Retos o requisitos no funcionales*

Cualquier mecanismo de monitorización que se aplique a personas tiene que lidiar con numerosas limitaciones o impedimentos tanto desde el punto de vista logístico, tecnológico, administrativo, económico o incluso legal. Estos impedimentos suponen retos que el prototipo debe ser capaz de saldar para resultar viable su aplicación en el mundo real, no limitándolo solamente al marco teórico. Estos retos acogen también los conceptos y aspectos funcionales que son primordiales en el diseño e implementación del prototipo.

Además, si se desea ofrecer un carácter diferenciador a las tecnologías y métodos actuales presentados en la Sección **??**, debe hacerse salvando las limitaciones o carencias que estos presenten, ofreciendo una alternativa que en dichos aspectos resulte igual o superior a la competencia.

A continuación, se enumeran estos retos a los que se enfrenta el prototipo:

Privacidad

Uno de los principales retos o impedimentos de un sistema que se encargue de monitorizar personas, es que éstas tienen concedidos derechos y libertades (al menos en la gran mayoría de países del mundo). Dichos derechos, están amparados tanto por organismos globales como regulaciones locales propias de cada país o región.

Por ejemplo, en máxima instancia, el artículo 12 de la Declaración Universal de Derechos Humanos adoptada por la ONU en el año 1948 defiende el derecho a la privacidad personal; así como el artículo 13 de la misma que defiende la libre circulación personal. Un sistema de monitorización que desee implantarse y no quiera vulnerar los Derechos Humanos, debe ser diseñado para ser capaz de amparar en todo momento los derechos de las personas monitorizadas.

Dichas regulaciones no sólo abarcan al propio concepto de la monitorización, sino que también regulan la utilización y almacenamiento de toda la información recogida. De esta manera, por ejemplo en España, la Ley

Orgánica de Proteción de Datos de Carácter Personal ampara el trato justo y privado de la información obtenido de la ciudadanía.

Es por ello, que un sistema de monitorización que no vulnere la legalidad vigente debe garantizar en todo momento el anonimato y privacidad de las personas que están siendo monitorizadas, contemplando desde el diseño el correcto trato de los datos e información adquiridos. De la misma forma, deben ser procesados y almacenados siguiendo los requerimientos legales necesarios.

El correcto abordaje de este reto, aunque no esté implicado en los fundamentos teóricos y tecnológicos del sistema, será el mayor escollo a la hora de la implantación real del mismo. Su tratamiento, es por tanto, uno de los principales retos a ser abordados en el prototipo.

Transparencia

Si bien grandes compañías como Google[132], Apple[130] o Facebook[131] monitorizan a millones de personas valiéndose que estás aceptan las condiciones de uso de los productos que dichas empresas despliegan por todo el mundo, el sistema propuesto debe ser capaz de funcionar sin necesidad del acuerdo expreso de las personas que están siendo monitorizadas. Es decir, de forma transparente para los individuos monitorizados.

Esto implica que pueda monitorizar de forma masiva y global sin necesidad de requerir una confirmación o acción expresa por parte de las personas siendo monitorizadas. Nuevamente, este reto o limitación roza la vulneración no ya de leyes locales, sino de derechos humanos, por lo que resulta un reto bastante delicado de abordar.

De igual manera, la transparencia debe entenderse en el procesamiento y lógica del propio prototipo. Un sistema que desee ser implantando y empleado tanto por organismos públicos como empresas privadas, debe de tener una filosofía transparente, con el fin de no realizar más acciones y tratamiento de la información que aquella que sea estrictamente necesaria y definida para el cometido del sistema.

Por último, la transparencia debe de aplicarse también a los datos. Si el sistema es empleado por una administración pública, los datos recogidos no privados ni personales, como por ejemplo datos derivados como series temporales o resúmenes históricos, deben de ser libremente accesibles para la ciudadanía.

La transparencia por tanto, en sus múltiples acepciones debe de ser contemplada en todo momento del diseño del prototipo sistema.

Bajo coste

A pesar de la situación favorable a nivel tecnológico descrita en la Sección 2.1, en lo económico se vive un periodo de recesión o “crisis” a nivel mundial.

Dado que se pretende que la fuente de datos propuesta sea una alternativa cuya implantación masiva sea viable, el aspecto económico resulta de vital importancia e impone una limitación considerable al prototipo.

Como se ha presentado en la Sección **??**, actualmente existen sistemas encargados de la monitorización tanto de personas como de vehículos. Sin embargo, en términos generales, su implantación a nivel masivo requiere un gran desembolsos económicos y conlleva costes de mantenimiento muy elevados.

Es por todo ello que uno de los retos del prototipo a ser contemplado es el de resultar de bajo coste. Entendiendo bajo coste tanto desde el punto de vista económico de su producción, como a nivel de desarrollo, de implantación y mantenimiento. El termino bajo coste, se aplica también a los aspectos de consumo asociado al sistema: como el consumo eléctrico o de recursos de procesamiento, almacenamiento o comunicación.

Globalidad

Relacionado y enunciado en el punto anterior se encuentra el aspecto de la globalidad. Si se desea que la aplicabilidad del prototipo sea abordable de forma global, el bajo precio del mismo no es el único requisito. Se requiere hacer uso de recursos y medios, que estén lo más extensamente implantados en todo el mundo.

Desarrollar cualquier producto tecnológico, sin tener presente la globalidad del mismo, viviendo cada vez más una sociedad globalizada puede suponer un fracaso en la aplicación del mismo. Es por ello, que la realización del sistema hace uso de medios disponibles de forma global a lo largo del mundo, como por ejemplo el uso de tecnologías BT y WiFi presentadas anteriormente, que resultan las más extendidas en todo el mundo, hasta el punto de ser consideradas un estándar de facto en las comunicaciones inalámbricas personales de corto y medio alcance.

Esta globalidad permite la utilización del sistema prácticamente en cualquier lugar del mundo donde los dispositivos a ser monitorizados se encuentren, sin tener bloqueos de región o necesitar características locales propias de cada escenario.

Escalablilidad

La globalidad lleva a su vez irremediablemente asociado el concepto de escalabilidad. Cualquier sistema informático que se desarrolle en la actualidad con pretensión de ser global, se debe desarrollar con la mentalidad de que su utilización va a ser masiva, y que su escalabilidad, por tanto, es vital.

En los últimos años ha surgido el concepto de Big Data, paradigma que hace referencia al procesamiento y almacenamiento eficiente de grandes cantidades de datos. El prototipo diseñado, si bien puede quedar a las puertas

del Big Data debido a su limitada implantación experimental, debe de ser constituido siguiendo los paradígmas y técnicas comúnmente empleadas y adoptados en los problemas categorizados como Big Data, con el fin de garantizar la mayor escalabilidad de los datos del prototipo.

Eficiencia cercana al Tiempo Real

De la mano de la escalabilidad, aparece el término de la eficiencia. En el prototipo se ambiciona lograr una eficiencia de monitorización cercana al Tiempo Real, atendiendo a que se desea minimizar el tiempo de notificación de la ocurrencia de una detección.

Esta eficiencia abarca tanto los aspectos propiamente de código eficiente y optimizado, así como un uso eficiente de los recursos disponibles, ya sean las comunicaciones entre los distintos elementos del prototipo, las transmisiones por red, el acceso a los datos o la ejecución de aplicaciones derivadas del prototipo.

Plantear como reto del prototipo lograr la mayor eficiencia en todos los aspectos del mismo, proporciona una monitorización rápida y eficiente, lo que permitiría detectar fluctuaciones o anomalías sin necesidad de largas esperas de tiempo. De igual manera, proporcionar un acceso muy optimizado a los datos históricos o actuales permite construir modelos de forma rápida y en tiempo cercano real. Y partiendo de esos modelos, se puede conseguir información útil sobre el conocimiento recién adquirido a partir de la aplicación de técnicas analíticas sobre los datos recogidos.

Robustez

Aunque se disponga de un prototipo eficiente, su tolerancia a fallos es otro componente de vital importancia. El concepto de robustez abarca no solamente al propio software, sino que debe ser extrapolable a todos los aspectos del prototipo.

Se presupone una robustez tanto en todos los componentes software implicados como en los componentes hardware. La robustez del software con mecanismos tales como detección, recuperación y notificación de errores, resulta de vital importancia para un prototipo que va a ser implantando, en la mayoría de las ocasiones, a kilómetros de distancia del laboratorio y sin un fácil acceso físico.

Esta imposibilidad de acceso físico supone un reto también para el hardware y los componentes software cercanos a él, como el sistema operativo. Debe de garantizarse la robustez del hardware empleado así como proveerle de un sistema operativo donde la robustez del mismo sea un requisito prioritario.

Por último, se desea que el almacenamiento de datos sea también robusto, tolerable a fallos y con redundancias para subsanar posibles pérdidas futuras.

Es por ello, que por las necesidades e imposiciones puestas al prototipo, la robustez es un elemento siempre presente en el diseño del mismo.

Autonomía

Como se ha indicado en el reto anterior, no se tiene garantizado el acceso físico a todos los componentes del prototipo de monitorización. Debido a que el sistema propuesto ha de ser implantado en edificios, calles o carreteras para captar los movimientos de personas y vehículos, y a que se espera que su utilización pueda ser global, el sistema de monitorización debe diseñarse de forma que pueda operar de forma autónoma.

Esta autonomía no radica únicamente en ser capaz de funcionar de forma independiente y aislada en caso de errores de interoperación con algún otro componente del sistema, sino que implica que el sistema debe ser capaz de funcionar en base a información actualizada o desactualizada, eligiendo por sí mismo cuál es su modo de funcionamiento más adecuado.

Además, dado que el sistema va a estar distribuido como una serie de nodos de captación, se desea que estos sean capaces de realizar la mayor parte del procesamiento requerido para la monitorización, con la finalidad de disponer no sólo de un red de captación, sino también de una red de procesamiento. De esta forma, siendo cada nodo de captación autónomo en su propio procesamiento, se consigue una mayor eficiencia al no delegar a otras partes del sistema más centrales el procesamiento de cada nodo.

Confiabilidad

Uno de los últimos retos del sistema, es ofrecer una información confiable. Si se desea que el sistema de monitorización propuesto sea capaz de proporcionar información con valor añadido para el funcionamiento de una Smartcity, tanto los componentes del sistema como sus datos deben ser confiables.

La confianza en los datos implica que estos deben de ser coherentes, consistentes y veraces en relación al fenómeno que está siendo monitorizado. En el caso del sistema diseñado, las variaciones en el flujo de personas y vehículos.

Es por ello, que los datos obtenidos por el sistema han de ser validados frente a otras fuentes de datos que midan o reflejen la misma magnitud. De esta forma, se debe garantizar que el sistema ofrece en todo momento información confiable sobre el fenómeno que está siendo monitorizado. De esta forma se ofrecerá en última instancia, una demostración empírica de que la monitorización mediante la captación de comunicaciones inalámbricas de dispositivos inteligentes es un medio confiable para el estudio de la movilidad de las personas y vehículos.

Modularidad

Debido a que se desea que el prototipo sea implantado en numerosos escenarios distintos, muchos de los cuales dispondrán de medios diversos y puede requerir una funcionalidad personalizada, se debe diseñar el prototipo haciendo uso de mecanismos de modularidad. Esto implica el prototipo sea dividido en partes más pequeñas que interoperen entre ellas con cierto grado de independencia funcional. Además, estos modulos deben de poder intercambiados en caso de necesidad o de ampliar la funcionalidad del prototipo sin requerir modificaciones en otros módulos.

El realizar un diseño modular del prototipo facilita el mantenimiento del software del mismos, su versatilidad ante distintos escenarios y la capacidad de expansión mediante nuevos módulos funcionales.

5.2.2 *Requisitos del sistema*

Una vez presentados los retos a los que se enfrenta el prototipo, se detallan los requisitos funcionales. Estos requisitos detallan qué debe ofrecer el prototipo del sistema, relegando a las siguientes secciones de este capítulo a detallar el cómo lo realizan.

El prototipo debe ser capaz de proveer información sobre los dispositivos cercanos a un emplazamiento geográfico.

Esta información debe ser capaz de enmarcar temporalmente la estancia en las cercanías de dicho dispositivo.

En base a esta información debe de proporcionar las siguiente métricas presentadas en la Sección 3.2 relativas al estudio del tráfico.

Presencia de dispositivos

El prototipo debe ser capaz de detectar a un dispositivo que se encuentre en las inmediaciones y discernir durante cuanto tiempo ha permanecido en ella. Además, el prototipo ha de ser capaz de recordar al dispositivo en detecciones sucesivas, identificándolo de forma única. Por último, el dispositivo ha de ser capaz de enmarcar temporalmente las estancias de cada dispositivo detectado.

Cuantificación o densidad de dispositivos

El prototipo en base a la detecciones de dispositivos en las inmediaciones enmarcadas temporalmente ha de ser capaz de cuantificar el número de dispositivos que se han encontrado y/o se encuentran en las inmediaciones.

Trazabilidad de los desplazamientos

El sistema ha de ser capaz de cruzar la información de varios puntos geográficos monitorizados para extraer información sobre los desplazamientos de los dispositivos que han realizado movimientos de uno a otro sitio monitorizado. El prototipo ha ser capaz de cauntificar dichos desplazamientos así como ser capaz de extraer información relativa a ellos, como el tiempo implicado o la frecuencia de las mismas.

Extracción de conocimiento

En base la monitorización de los dispositivos, el prototipo del sistema ha de ser capaz de ofrecer información histórica resumida sobre las magnitudes de las detecciones y monitorizaciones. Esta información debe de ofrecer conocimiento aplicado extraible sobre el comportamiento de los dispositivos.

Predicción de comportamientos

En base a la información histórica disponible del sistema, este ha ser capaz de elaborar predicciones sobre las magnitudes y comportamiento de los dispositivos detectables en el futuro.

5.3 COMPONENTES DEL SISTEMA DE MONITORIZACIÓN

Una vez definidos los requisitos que se le imponen al prototipo del sistema, se presentan los componentes del mismo. Cómo se ha presentado en la Sección 5.1.6, se prevé que el sistema esté compuesto por varios **nodos** sensores. Se propone que la información que generen dichos **nodos** sea enviada a un **servidor de cómputo** encargado de almacenar y cruzar la información de la red para generar históricos. Finalmente, se propone que la información y conocimiento generados por el sistema sean almacenados en la **nube**, con el fin de publicarlos y hacerlos disponibles (Figura 5.48).

Figura 5.18
Componentes principales del prototipo de sistema de monitorización propuesto en esta tesis: el hardware de captación, el servidor de cómputo y la nube.

A su vez, cada uno de estos componentes se divide en distintos subcomponentes y módulos que entraman el prototipo del sistema (Figura 5.19). La correcta interacción y comunicación entre estos módulos resulta clave para la finalidad del mismo, así como los aspectos de diseño e implementación de los mismos satisfaciendo los requisitos no funcionales presentados en la Sección 5.2.1.

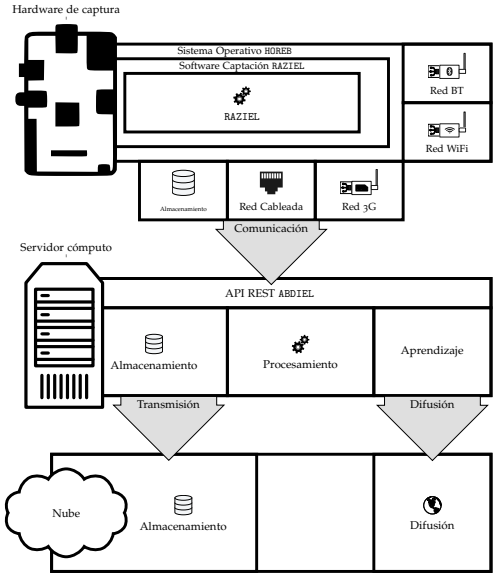


Figura 5.19
Elementos componentes del sistema de monitorización propuesto en esta tesis. Se distinguien tres elementos principales: el hardware de captación, el servidor de cómputo y la nube.

5.3.1 *Nodo de monitorización*

El dispositivo de monitorización cosntituye todos elementos, tanto físicos como lógicos, que son emplazados para realizar la monitorización.

5.3.1.1 *Hardware*

Se debe de proveer de un dispositivo físico de bajo coste que permita la captación de las comunicaciones inalámbricas propuestas en esta tesis. Dicho dispositivo, por tanto, debe disponer de hardware capaz de monitorizar comunicaciones BT y WiFi. Se requiere que el dispositivo sea de bajo coste tanto a nivel energético como monetario. Además, dado que se plantea que su uso sea lo más global posible, el dispositivo propuesto tiene que estar preparado para operar tanto en interiores como exteriores, en rangos de temperatura ambientales habituales y ser lo más estándar posible en cuestiones de alimentación eléctrica y de componentes.

Finalmente, el dispositivo debe de disponer de suficiente capacidad de cómputo para ser capaz de operar de forma autónoma y robusta, así como disponer de medios que faciliten la intercomunicación con el fin de difundir en tiempos cercanos al real la información recolectada.

5.3.1.2 *Sistema operativo del dispositivo*

El hardware necesitará un Sistema Operativo (SO) que facilite la comunicación entre los elementos físicos y los elementos software del nodo de captación. Se necesita disponer de un sistema operativo que se ajuste a las necesidades del sistema de monitorización propuesto, siendo tolerable a errores y corrupciones tanto por fallos en el hardware del dispositivo como por acción del tiempo. Dicho SO debe de ser capaz de funcionar de forma ininterrumpida a lo largo de cantidades indeterminadas de tiempo, siendo capaz de garantizar el arranque a pesar de fallos por factores tanto internos como externos, como por ejemplo, cortes de alimentación eléctrica.

Además, debe de facilitar las interfaces de comunicación BT y WiFi para operar en modo descubrimiento y modo monitor respectivamente. Por último, el SO debe de ser lo más extendido y estándar posible, con el fin de facilitar las tareas de integración e implantación del sistema de monitorización.

5.3.1.3 *Software de captación*

Se requiere elaborar un software que habilite la captación y monitorización de dispositivos. Este software de captación y monitorización ha de ser portable, independiente de la plataforma y en tiempo cercano al real. Debe habilitar la captura de los beacons BT y las tramas de red WiFi emitidas en las inmediaciones por todos los dispositivos en un rango cercano y determinado.

Este software debe de ser capaz de extraer cuanta más información posible tanto de los beacons como de las tramas sobre los dispositivos que las generan, sin vulnerar en ningún momento las directrices de protección de datos y privacidad personal. Para ser viable para la monitorización, debe de contabilizar de forma eficiente y con la menor cantidad de pérdidas e interferencias posible el tránsito de dispositivos por la zona, disponiendo de mecanismo de recuperación ante errores y trato de anomalías.

Debido a que se exige que opere de forma robusta y aislada, debe de disponer de mecanismos de control y configuración en caliente que permitan los cambios y actualizaciones, sin tener garantizado el acceso físico o remoto al SO. Dicho software debe de ser capaz de operar de forma autónoma, tanto para los parámetros de configuración como para la notificación de la información capturada. Debe ser capaz de operar tanto de forma incomunicada, como formando parte de una red de nodos de captación más grande integrado en el sistema.

5.3.2 *Arquitectura de comunicación*

Dado que se desea que la notificación de la información gestionada por los dispositivos de captación sea lo más cercana al tiempo real, se deben ofrecer vías de comunicación entre los dispositivos de captación y los servidores (físicos o cloud) encargados del almacenamiento y extracción de información y conocimientos derivados. Dichas vías de comunicación tienen que ser escalables, eficientes y hacer uso de formatos estándares para permitir su adaptación a las infraestructuras y necesidades de los puntos geográficos donde sean ubicados los dispositivos de captación.

Dicha arquitectura de comunicación, debe de ser independiente del medio de transmisión físico que se realice para las comunicaciones (LAN ETHERNET, WiFi, 3G, VPNs...), empleando una vía u otra en función de las necesidades y limitaciones del escenario o de las infraestructuras existentes con las que tenga que co-existir el nodo de captación.

Además, se debe de ofrece servicios que permita la comunicación entre los dispositivos y los servidores para el intercambio de la información entre ellos. Estos servicios deben de ser eficientes y seguros.

5.3.3 *Servidor de cómputo*

El servidor de cómputo sirve de paso intermedio entre los dispositivos de captación y la difusión en la nube de la información capturada. Su naturaleza puede ser física o basada igualmente en la nube. Sin embargo, en el prototipo propuesto se ha impuesto la necesidad de que se trate de una máquina física. Dicha necesidad será discutida más adelante.

5.3.3.1 *Almacenamiento local*

El sistema debe ser considerado cercano al Big Data debido a la cantidad de datos que es factible que sea generado por un número creciente de nodos de captación. Se deben de ofrecer por tanto unos principios para el almacenamiento escalable para los datos obtenidos por un número indeterminado de dispositivos de captación funcionando de forma simultánea y continuada a lo largo del tiempo.

Para la extracción de información y conocimiento en tiempo real se tiene que garantizar que dicho almacenamiento permite el acceso eficiente a los datos, facilitando a los algoritmos de análisis las operaciones de sumarización, agregación y cotejamiento.

5.3.3.2 *Procesamiento eficiente*

Los datos obtenibles por el sistema son empleados por algoritmos y métodos que resulten eficientes con el fin de disponer de resultados en tiempos cercano al real. Muchos de estos métodos necesitan estructuras más complejas derivadas de los datos, estructuras que deben de ser generadas por el sistema en el **servidor de cómputo**.

Se debe por tanto ofrecer una selección de métodos de procesamiento eficientes, que permitan posteriormente interpretar de forma heurística los datos almacenados. Para ello, estos métodos deben de ser capaces de transformar los datos en crudo en estructuras complejas que permitan su análisis, tales como series temporales, matrices entrada / salida, grafos dirigidos o itinerarios de desplazamiento.

5.3.3.3 *Análisis*

Partiendo de las estructuras generadas, el sistema debe de ser capaz de analizar los datos y ser capaz de elaborar conclusiones y resultados sobre el desplazamiento de las personas y vehículos. Se entiende el análisis de los datos tanto en tiempo real (con el fin de satisfacer otros elementos del sistema) como análisis históricos a modo de informes sobre el tráfico.

La posibilidad de elaborar estos análisis de la movilidad en base a los datos y sus estructuras derivadas suponen uno de los objetivos diferenciadores del sistema. Se desea por tanto ser capaz de elaborar mecanismos de análisis que sean capaces de funcionar de forma lo más automatizada posible, con el fin de ser capaces de explotar el carácter analítico del sistema, no únicamente contabilizador.

Es por ello que el **servidor de cómputo** debe ser capaz de ofrecer métodos analíticos sobre las estructuras de datos obtenidos, que permita obtener información relevante para la interpretación de los datos de la movilidad de personas y vehículos.

5.3.3.4 *Aprendizaje automático*

Debido a que el sistema está planteado para ser operado de forma ininterrumpida, el análisis de los datos y sus estructuras debe ser realizado a lo largo del tiempo. De esta forma, los sucesivos análisis pueden ser empleados para extraer conocimiento sobre como evoluciona la movilidad de las personas y los vehículos. Estas fluctuaciones habilitan la posibilidad de establecer mecanismos de aprendizaje sobre los estados de los flujos de movimientos.

El sistema ha de ser capaz de ofrecer algoritmos que sean capaces de obtener información adicional partiendo de las estructuras generadas en base al conocimiento previo adquirido por el sistema, realizando búsqueda de patrones y detección de anomalías sobre el flujo. Una vez lograda una base de conocimiento de cada **nodo** de detección, es factible el empleo de dicho conocimiento para ser capaces de elaborar predicciones sobre estados futuros del tráfico. Estas predicciones pueden ser realizadas tanto sobre las magnitudes como por las preferencias de los desplazamientos o cualquier otra fuente de datos generada por el sistema. Además es deseable ser capaz de predecir con ventanas de tiempo de varios días o instantes de tiempo inmediatos del futuro, como por ejemplo, la próxima hora o a corto o inmediato plazo. El **servidor de cómputo** debe ser capaz de realizar predicciones sobre el estado futuro a largo, medio, corto o inmediato plazo de los entornos cercanos a cada nodo, que permitan prever el estado futuro de la movilidad de los puntos monitorizados y su iteración entre ellos.

5.3.4 *Nube*

5.3.4.1 *Almacenamiento en la nube*

Debido al requisito de transparencia impuesto al sistema, este debe de ofrecer la posibilidad de consulta histórica de los datos y de la información generada, que esté disponible para empleo y aprovechamiento tanto de otros investigadores, la ciudadanía, instituciones o empresas que desean hacer uso de ella. Este almacenamiento público, se planeta se encuentre alojado en la **nube** con el fin de ofrecer los datos respaldados de forma accesible y barata.

5.3.4.2 *Difusión*

la difusión resulta de vital importancia para la aplicación práctica del sistema, por lo que el prototipo del sistema debe de ser capaz de ofrecer mecanismos para la publicación en tiempo cercano al real de la información generada o predicha de forma accesible, estándar y global.

5.4 NODO DE MONITORIZACIÓN: HARDWARE

En esta sección se detallan los elementos hardware que componen el prototipo de **nodo** de monitorización. Este **nodo**, como se ha indicado en la Sección 5.1.6, será emplazado en las zonas donde se desee monitorizar los dispositivos, y por extensión, las personas y vehículos.

Tal y como se ha especificado en la Sección 5.1 este **nodo** ha de disponer de interfaces de red Bluetooth y WiFi con la que poder realizar la búsqueda de dispositivos cercanos (en BT) y y captura de tramas (en WiFi) valiéndose de los mecanismos detallados en las secciones 4.2 y 4.3 respectivamente.

El **nodo** ha de disponer de capacidad de procesamiento para interpretar los paquetes `FHS` y las tramas WiFI así como memoria suficiente para poder albergar al menos los identificadores y marcos temporales de los dispositivos actualmente en las inmediaciones. Además, debe de disponer de interfaces que permitan la transmisión de la información de monitorización generada al **servidor de cómputo**.

Finalmente, el hardware del **nodo** debe de satisfacer los requisitos no funcionales presentados en la sección 5.2.1.

Se presentan a continuación los componentes hardware del **nodo** que serán descriptos en esta sección:

- Computador de placa única en la página 147.
- Interfaz de red Bluetooh en la página 150.
- Interfaz de red WiFi en la página 151.
- Interfaz de red 3G/2G en la página 152.
- Tarjeta microSD en la página 153.
- Periféricos adicionales en la página 154.
- Emplazamiento en la página 159.
- Coste del hardware en la página 165.
- Consumo energético en la página 166.

5.4.1 *Computador de placa única*

Un computador de placa única (`Single Board Computer` o `SBC`) concentra en una sola placa el microprocesador, la RAM y los periféricos de entrada/salida habituales en un computador, haciéndolo de forma compacta en la placa base. Esta placa base autocontiene todos los elementos requeridos para el funcionamiento básico del computador.

En los últimos años, con el auge del Internet de las Cosas (Sección 2.1) se han popularizado los `SBC` por su capacidad de ofrecer computadores de reducido tamaño, integrables en multitud de objetos cotidianos o para su uso en entornos industriales y sistemas embebidos.

Entre los `SBC` más populares se encuentran las familias `Arduino`[13], `Gumstix` [110] y `RaspberryPi` [219]. Además, al estar basadas en la filosofía de hardware libre, existen numerosas variaciones e implementaciones çlónicas"sobre estos `SBC`.

Si bien `Arduino` es el `SBC` más popular, su uso se encuentra más centrado en pequeños controladores y procesadores/actuadores de señales. Es por ello que los modelos disponibles suelen constar de escasa memoria RAM.. entre 1KB y 8KB de RAM, resultando insuficiente para la monitorización de miles de dispositivos. [6]. Además, se consideran dispositivos autónomos, por lo que de forma nativa no suelen incorporar medios de red. Por último, `Arduino` no emplea un Sistema Operativo al uso, sino que la programación se realiza directamente sobre el Firmware, por lo que decaería en el desarrollo del prototipo la implementación de todos los recursos necesarios.

Es por ello que para el prototipo se ha optado por emplear el otro `SBC` más empleado en prototipado rápido de hardware: `Raspberry Pi` [273]. Es un `SBC` desarrollado en Reino Unido por la Fundación Raspberry Pi, y es de hardware abierto en lo que al `SoC`[7] se refiere. El lo referente al software, si es de código abierto, existiendo varias distribuciones del sistema operativo (que será discutido en la sección 5.5).

[273] Raspberry Pi user guide

Existen varias revisiones de `Raspberry Pi`, habiendo sido la empleada en el prototipo presente la `Raspberry Pi 2` (Tabla 5.1) que cuenta con un procesador `ARM Cortex-A7` [11] con 4 núcleos de procesamiento y 1024MB de RAM. Si bien las primeras versiones del prototipo se empezaron a desarrollar sobre una `Raspberry Pi B` con un procesador `ARM11` con un sólo núcleo de procesamiento.

[11] ARM Cortex A7 Specifications

6 ↑Supuesto que la Dirección MAC son 6 Bytes y una marca de tiempo son 8 bytes, dedicando únicamente la RAM al almacenamiento de los pasos, serían 46 dispositivos simulatáneos los almacenables por el modelo de 1KB.

7 ↑El procesador incorporado suele ser propietario por lo que la fundación no libera sus esquemas. Sin embargo, si libera las especificaciones del resto de elementos de la placa.

Tabla 5.1
Comparación de los principales de modelos de Raspberry Pi existentes en el mercado.

	RASPBERRY PI 3	RASPBERRY PI ZERO W	RASPBERRY PI ZERO	RASPBERRY PI 2	RASPBERRY PI B	RASPBERRY PI A+
						
Detalles						
LANZAMIENTO	29-Feb-2016	28-Feb-2017	30-Nov-2015	1-Feb-2015	15-Feb-2012	10-Nov-2014
PRECIO	35$	10$	5$	35$	30$	20$
SOC						
TIPO	Broadcom BCM2837	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2835	Broadcom BCM2835
CORE	Cortex-A53	ARM1176JZF-S	ARM1176JZF-S	Cortex-A7	ARM1176JZF-S	ARM1176JZF-S
Nº CORES	4	1	1	4	1	1
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV
RELOJ CPU	1.2 GHz	1 GHz	1 GHz	900 MHz	700 MHz	700 MHz
RAM	1024 MB	512 MB	512 MB	1024 MB	512 MB	256 MB
Conectividad						
USB	✓(4)	✓(1)	✓(1)	✓(4)	✓(2)	✓(2)
ETHERNET	✓	✗	✗	✓	✓	✓
HDMI	✓	✓	✓	✓	✓	✓
SPI	✓	✓	✓	✓	✓	✓
I2C	✓	✓	✓	✓	✓	✓
GPIO	✓	✓	✓	✓	✓	✓
LCD	✓	✗	✗	✓	✓	✓
CAMERA	✓	✓	✓	✓	✓	✓
SD/MMC	✓	✓	✓	✓	✓	✓
Conectividad inalámbrica en placa (On-Board)						
WIFI	✓	✓	✗	✗	✗	✗
BLUETOOTH	✓	✓	✗	✗	✗	✗
Dimensiones						
ALTURA	56.5 mm	30 mm	30 mm	85.6 mm	85.6 mm	65 mm
ANCHURA	85.6 mm	65 mm	65 mm	56.5 mm	53.9 mm	56.5 mm
PROFUNDIDAD	17 mm	5 mm	5 mm	17 mm	17 mm	10 mm
PESO	45 gm	9 gm	9 gm	45 gm	45 gm	23 gm
Alimentación						
POTENCIA N.	800 mA	180 mA	160 mA	800 mA	700 mA	200 mA
FUENTE	USB o GPIO	USB o GPIO	USB o GPIO	USB o GPIO	USB o GPIO	USB o GPIO

La principal ventaja del empleo de la `Raspberry Pi 2` a parte de su mayor potencia de computo paralela, es la inclusión de 4 puertos `USB` que permiten una capacidad de ampliación superior mediante dispositivos comerciales, en nuestro caso, las tarjeta de red, y que dispone de un bus con un ancho de banda de $480 mbps$[8]. Además, al igual que el resto de placas de la familia `Raspberry Pi`, incluye pines de entrada/salida de propósito general comunmente llamados `GPIO` (40 en el caso del modelo 2). Por último la placa consta de puerto Ethernet 10/100 LAN y de un puerto HDMI (Figura 5.20).

Para contener tanto el sistema operativo como el software a ejecutar en la `Raspberry Pi`, se requiere de una tarjeta `microSD` que contenga a ambos. Esta tarjeta será insertada dentro de la placa base dentro del slot correspondiente, cuya velocidad del bus está limitado a $20 MBs$ [233] tanto para operaciones de lectura como de escritura.

[233] RPi SD cards

8 ↑Este ancho de banda es compartido por todos los dispositivos conectados por USB y por el puerto Ethernet, con el que comparte bus.

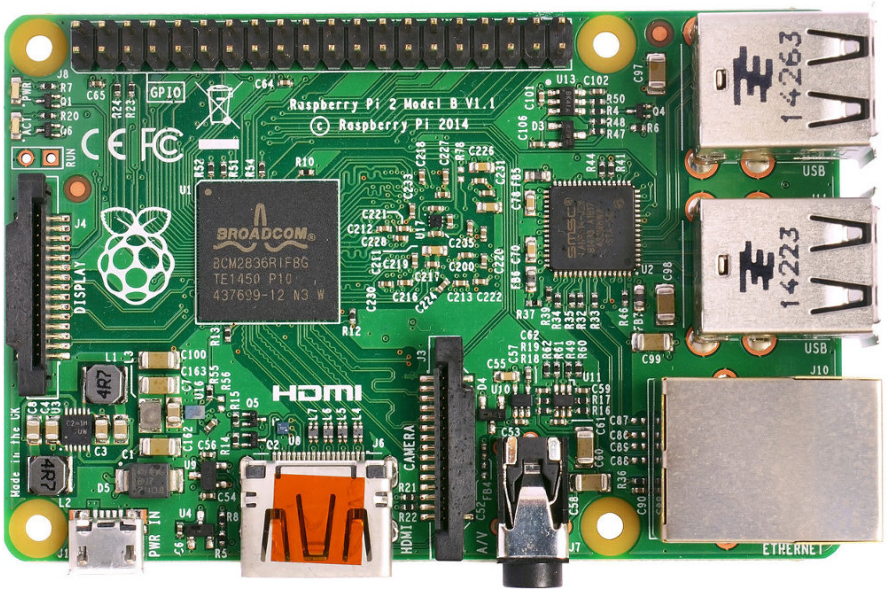


Figura 5.20
Fotografía de una Raspberry Pi 2, donde se distinguen los 40 pines GPIO, los 4 puertos USB (en bloques de 2), el puerto Ethernet, el puerto HDMI y el puerto de alimentación micro-usb.

Los SBC de la familia Raspberry Pi han resultado ser de los más empleados para la sensorización en entornos inalámbricos [81, 284]. Además, han resultado ser increiblemente robustos y resistentes, hasta el punto de que la NASA ha empleado este SCB para instrumentación en algunas de sus misiones, siendo la más relevante la Pi-Stat [64] consistente en un satélite de bajo coste basado en Raspberry Pi.

[81, 284] Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications, Raspberry Pi as a wireless sensor node: performances and constraints

[64] Pi-Sat: A Low Cost Small Satellite and Distributed Spacecraft Mission System Test Platform

Durante el desarrollo de esta tesis doctoral, se han producido numerosos avances en este dispositivo de placa única, siendo el último de ellos el anuncio de su revisión 4, incorporando un mejor procesador y mayor cantidad de memoria RAM. La elección de este dispositivo de computación de placa única, frente a las alternativas, ha servido para garantizar la viabilidad constante del prototipo, al no haber experimentado ni problemas de stock ni de soporte, al contrario que con otras placas alternativas existentes en el mercado.

5.4.2 *Interfaz de red BT*

Los dispositivos Bluetooth, como se ha presentado en la tabla 4.1 de la Sección 4.2, se distinguen en tres clases en función de su radio de alcance. Para el prototipo se ha optado por un tarjeta de red Bluetooth de Clase 1, con alcances teóricos de hasta 100 metros, `Conceptronic 1004108`, que encapsula un chip `Qualcomm CSR8510`[9].

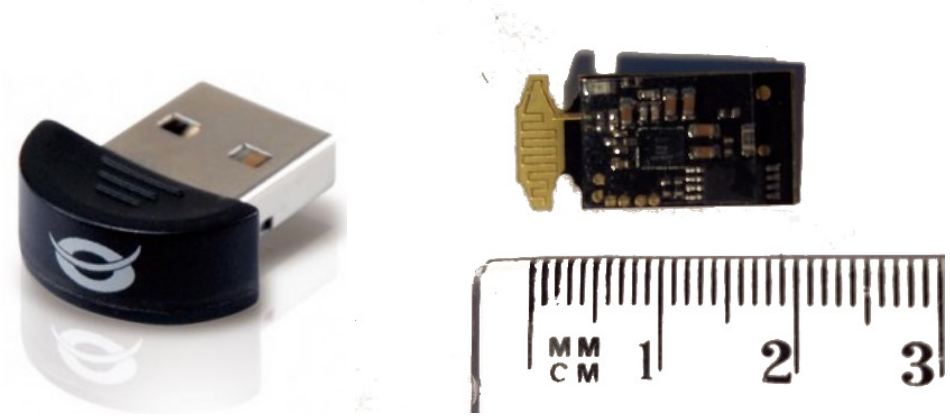


Figura 5.21
Tarjeta de red Bluetooth empleada en el prototipo. En la izquierda la versión comercial con los embellecedores de plástico. A la derecha la versión desnuda de la tarjeta de red.

El empleo de una tarjeta de red Clase 1 se justifica por las necesidades de alcance de la antena. Esta tarjeta se conecta en la placa de `Raspberry Pi` mediante la interfaz USB. Su cortas dimensiones permite extraer todo el módulo de la placa mediante un cable USB de la placa principal, con el fin de ubicarlo en una mejor posición.

Si bien revisiones del dispositivo de placa única (ver Tabla 5.1) incorporan conexión Bluetooth, esta es de clase 2, por lo que su alcance efectivo es menor. Además al encontrarse integrado en la placa, el emplazamiento en mejores posiciones de la antena no resulta viable sin tener que manipular la placa.

9 ↑https://www.qualcomm.com/products/csr8510

5.4.3 *Interfaz de red WiFi*

Para la interfaz de red WiFi es imperativo que soporte modo monitor, como se ha presentado en la Sección 4.3.4, pues el modo de captura de paquetes sin vinculación a un `AP` que establece el protocolo WiFi.

En el caso del prototipo se ha optado por un tarjeta `TP-LINK TL-WN722N` que encapsula un chip `Atheros AR9271L` que permite el funcionamiento de dicho modo.

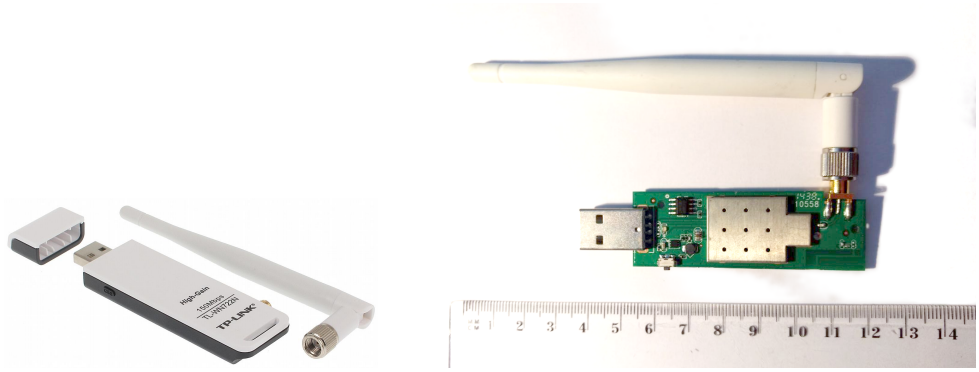


Figura 5.22
Tarjeta de red WiFi empleada en el prototipo. En la izquierda la versión comercial con los embellecedores de plástico. A la derecha la versión desnuda de la tarjeta.

Dicha tarjeta que se conecta a la placa mediante la interfaz USB, y dispone de una conexión coaxial que permite ubicar la antena en una posición separada del módulo con el fin de alcanzar una mejor posición para la Pantallaización. Dicha interfaz opera únicamente en la banda de frecuencia de los $2.4GHz$ quedándose sin cubrir la banda de los $5GHz$. Esto es debido a una decisión deliberada, pues como se ha presentado en la Sección 4.3, los dispositivos inteligentes que emplean esta banda son menos frecuentes y además disponen de un alcance menor.

Su ancho de banda máximo, según especificaciones del fabricante, se sitúa en $150Mbps$ ($18.75MBps$). Más adelante se estudiará (Sección 6.1) que dicho ancho de banda resulta más que suficiente para abarcar las necesidades del prototipo desarrollado.

5.4.4 *Interfaz de red 3G/2G*

Como tarjeta de red 3G/2G se emplean las tarjeta Huawei e173 y Huawei E3531 capaces de operar con los protocolos de red EDGE, GRPS, GSM, HSDPA, HSUPA y UMTS. Para el protocolo GSM opera en las bandas 850/900/1800/1900 y para el protocolo UMTS en las bandas 900/2100. Se conecta al dispositivo de placa única mediante la interfaz USB. Alcanza un ancho de banda de hasta $7,2 Mbps$ empleando el protocolo HSDPA. El motivo del empleo de dos tarjetas, es que la primera es descatalogada durante el desarrollo del prototipo.

Figura 5.23
Tarjeta de red 3G Huawei e173 empleada en el prototipo.

Ambas tarjetas disponen adicionalmente de un lector de tarjeta microSD, así como un LED que permite conocer el estado de la conexión (Tabla 5.25). Según el fabricante, tienen un rango de temperatura de funcionamiento de $-10C$ a $45C$.

Tabla 5.2
Significado de los colores del LED de la tarjeta 3G

COLOR	PARPADEO	SIGNIFICADO
Verde	3s - Dos veces	Alimentado
Verde	3s - Una vez	Registrándose en red 2G
Azul	3s - Una vez	Registrándose en red 36
Verde	Sólido	Conectado a red 2G
Azul	Sólido	Conectado a red 3G
Apagado	–	Fallo de alimentación o apagado

5.4.5 *Tarjeta microSD*

Al contrario que otros SBC como ARDUINO que disponen de una memoria ROM incorporada en la placa, las Raspberry Pi necesitan de una tarjeta de memoria microSD[10] donde se almace el software que debe ejecutar la placa, así como el sistena operativo.

Se opta por una tarjeta micro SD Clase 10 UHS-3 marca Kingston Technology, con 16GB de capacidad. Dicha tarjeta dispone de una velocidad de $90MB/s$ en operaciones de lectura y de $45MB/s$ en operaciones de escritura. Cómo se ha indicado en la sección 5.4.1 el bus de la Raspberry Pi limita la tasa de lectura y escritura a $20MBs$ [233] ampliamente superior a la velocidad de la tarjeta elegida. Aunque pueda parecer excesivo esta tasa de velocidad adicional será grata a la hora de realizar los volcados de las imágenes del sistema operativo y software en la tarjeta en su puesta a punto para una primera utilización (Figura 5.24).

[233] RPi SD cards

Además de que la diferencia de precio entre tarjetas de menor velocidad disponibles en el mercado(por ejemplo, clase 4) es asumible. Así por ejemplo, mientras que la tarjeta elegida tiene un precio de venta de unos 12, una tarjeta Clase 4 del mismo fabricante[11] tiene un precio de 10 ofreciendo una tasa de lectura y escritura de tan sólo $4MB/s$.

Figura 5.24
Tarjeta microSD empleada, con el código de significado.
Fuente:https://www.kingston.com/latam/flash/microsd_cards/sdca3

Sin embargo, la vida útil de este tipo de tarjetas es muy limitada, aspecto que será tratado especialmente en la diseño del sistema operativo (Sección 5.5.1)

10 ↑O SD en la versión A+
11 ↑La gama inferior según su propio catálogo:
https://www.kingston.com/us/flash/microsd_cards

5.4.6 *Periféricos adicionales*

El hardware presentado anteriormente es el imprescindible para el funcionamiento del prototipo. Sin embargo, otros elementos hardware pueden hacer más sencilla y cómoda la depuración e interoperación del prototipo. A continuación se enumeran y presentan brevemente los elementos hardware adicionales que han sido empleados en el desarrollo del prototipo.

Leds de notificación de estado

En primera instancia, para facilitar las tareas de depuración y detección de errores en las fases tempranas de testeo del prototipo, se le dotó de un capa software que era capaz de comunicarse con el exterior mediante mensajes codificados en leds de colores.

Si bien el prototipo actualmente es ubicado sin visibilidad directa, estos leds y su software asociado son actualmente contemplados en la implentación del prototipo.

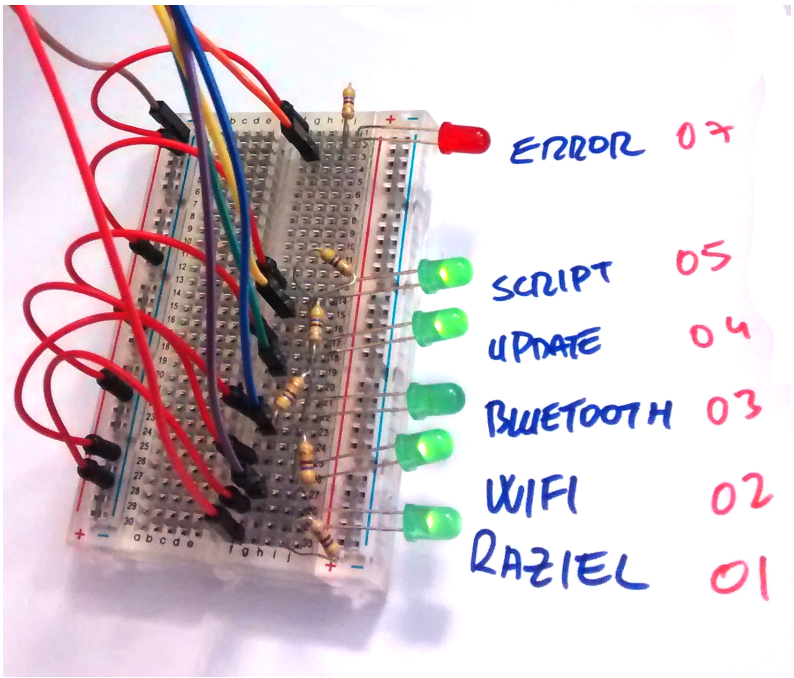


Figura 5.25
Leds de notificación de estado del **nodo**, conectados a la placa mediante una protoboard conectada a los pines `GPIO`.

En caso de no haber ningún error, los leds verdes numerados de 1 a 5 (Figura 5.25) se encienden de forma constante. En caso de producirse algún error el led 7 (de color rojo) se enciende y se apaga el led correspondiente a la funcionalidad en la cual se ha producido el error.

Si bien este modo de depuración es bastante simpled puede servir para detectar errores en la ejecución del **nodo**.

Pantalla Integrada

Para facilitar las opciones de Monitorización en tiempo real, así como la depuración, se puede dotar al prototipo de una pantalla integrada. En el caso del prototipo presentado en esta tesis, se trata de una pantalla de 4 pulgadas de diámetro modelo `4 inch RPI LCD` del fabricante `Waveshare Spotpear`. Esta pantalla se conecta mediante los pines `GPIO` de la placa y se alimenta por medio ellos (Figura 5.26).

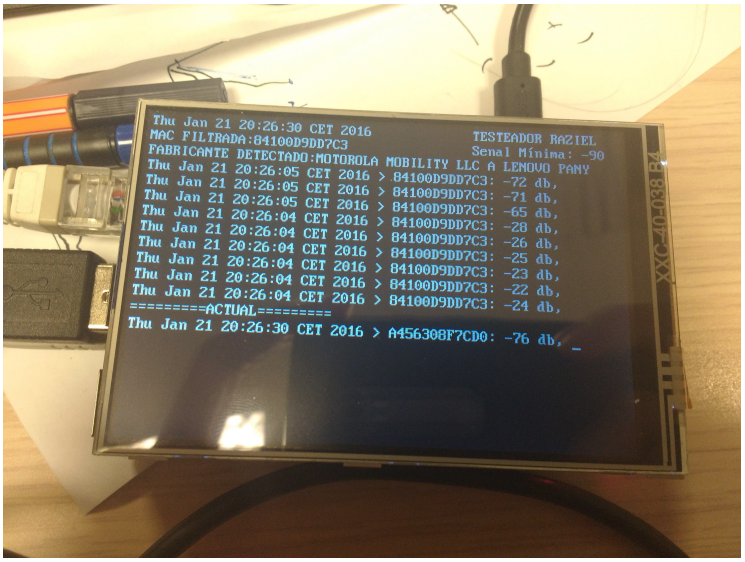


Figura 5.26
Pantalla integrado en la placa de 4 pulgadas. En la captura, la pantalla muestra un testeador de Monitorización WiFi.

Si bien la resolución de la pantalla está limitada a $320x480$ píxeles, estos suponen una mejor interpretación del estado del sistema que los leds, permitiendo al prototipo mostrar información sobre las magnitudes de monitorización llevada a cabo.

A pesar de ello, no se hace necesario dotar al prototipo de esta pantalla en su versión a ser implantada en los lugares a monitorizar, sino que su utilidad se presenta en escenarios donde haya que realizar demos o pruebas de la tecnología o bien haya que realiza estudios previos al emplazamietno de los **nodos**.

Pantalla Externa

La pantalla de 4 pulgadas presentada anteriormente tiene la limitación de que debe estar integrada en la placa. Dado que el **nodo** es susceptible de instalarse en cualquier entorno, su acceso puede resultar complicado en multitud de escenarios en los que el **nodo** se encuentre situado a cierta altura. Para ello se complementa la pantalla en placa con una pantalla externa marca `TONTEC` empleada habitualmente en videovigilancia. Esta pantalla se conecta a la placa mediante la interfaz HDMI y requiere alimentación eléctrica externa, por lo que su empleo resulta menos compacto (Figura 5.27).

Figura 5.27
Pantalla externa conectado a la placa mediante la interfaz HDMI. Esta pantalla requiere de alimentación externa

La pantalla dispone de una resolución de 1920×1080 píxeles en 7 pulgadas de diámetro, lo cual permite una visualización similar a la ordenador portátil. Su empleo para tareas de detección de errores y depuración en caliente, resulta muy útil.

Teclado

Los dispositivos presentados anteriormente permiten la visualización de los errores, pero no permiten intervenir directamente con el **nodo**. Mediante un teclado `Logitech k400r` se habilita la conexión con el **nodo**. Este teclado dispone de un stick usb que conecta el teclado con el dispositivo donde se conecte (Figura 5.28).

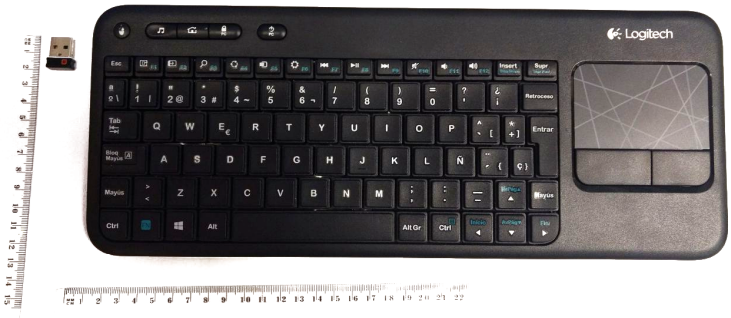


Figura 5.28
Teclado portátil empleado para la comunicación con el prototipo.

Gracias a este elemento, es posible realizar modificaciones en caliente de la configuración del **nodo** en la instalación del dispositivo.

Reloj

Las placas de la familia `Raspberry pi` carecen de un reloj en tiempo real (`RTC`), lo cual implica que no disponen de mecanismos para almacenar la fecha cuando el dispositivo deja de recibir alimentación eléctrica. Aunque esta carencia puede ser subsanada mediante software (cómo por ejemplo mediante un servidor NTP como se presenta en la Sección 5.5.4).

En escenarios donde no se ha posible subsanar la carencia de reloj mediante software, es posible acoplar a la placa `Raspberry pi` un módulo de reloj mediante los pines `GPIO`. En concreto en el prototipo se han realizado pruebas con un módulo `ZS-042` que encapsula un chip `DS3231` (Figura 5.29).

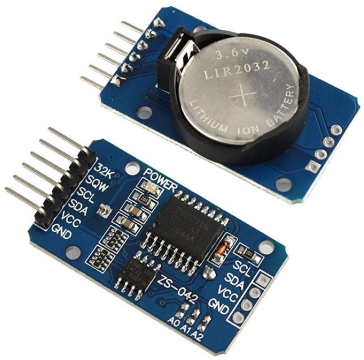


Figura 5.29
Módulo de reloj o `RTC` empleado para el almacenamiento de la fecha y hora en el prototipo.

Debido a que la monitorización requiere de ventanas de tiempo, la gestión de la correcta fecha y hora dentro del prototipo del sistema es de vital importancia. Por ello, en caso de ser requerido en el prototipo se le puede dotar del módulo de reloj. Este se alimenta por una batería `CR2032` que según especificaciones del fabricante tiene una autonomía aproximada de 10 años. La información del chip es accesible y modificable mediante el bus `I2C` de los pines `GPIO`. El chip `DS3231` tiene una precisión de $1s$ y genera fechas válidas hasta el 2100.

Cable de alimentación y baterías

La placa `Raspberry Pi` se alimenta mediante el puerto microUSB o mediante los pines `GPIO`. En el prototipo, se opta por realizar la alimentación mediante un cargador microUSB de $5V$ DC y $2000mA$, con una longitud de 1 metro y el transformador alojado en el enchufe. En caso de requerir mayor distancia, se puede hacer una instalación eléctrica estándar para acercar el enchufe a la toma eléctrica.

En casos donde no haya suministro eléctrico o haya que realizar pruebas, la placa puede alimentarse con una batería externa o powerbank. En el caso del prototipo se han realizado pruebas con baterías `AUKEY` de $20000mAh$.

Figura 5.30
El prototipo funcionando en modo demostración haciendo uso de una batería externa.

En la Subsección 5.4.8 se presentarán los consumos energéticos de los distintos elementos del prototipo.

Otros componentes

En diversas colaboraciones emergentes, se está trabajando por dotar al prototipo de mayor cantidad de sensores como sensores medioambiantes, sonoros,... Para ello, se emplea hardware diseñado a medida por ingenieros electrónicos conectados a la placa mediante los pines `GPIO` (Figura 5.31).

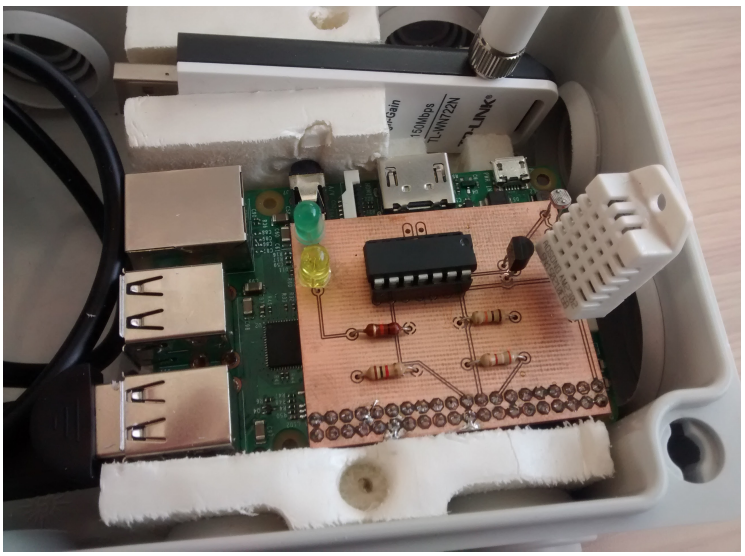


Figura 5.31
Sensor de temperatura y humedad conectado a la placa mediante los pines `GPIO`

La incorporación de nuevos sensores en la placa es factible en el prototipo gracias a su capacidad de extensión mediante los pines `GPIO` así como los puertos USB del que dispone.

5.4.7 *Preparado para emplazamientos*

El **nodo** de monirotización ha de ser emplazado tanto en entornos exteriores como interiores. En esta sección se describen los elementos que permiten su implantación en estos escenarios.

El principal contenedor del **nodo** es una caja estanca `SOLERA Ref.716` (Figura 5.32) con unas dimensiones exteriores de $160x120x71cm$ y unas dimensiones interiores de $153x110x65cm$. Dispone de 8 conos de comunicación M25 y dos conos M32.

Esta caja estanca tiene una certificación IP55, la cual según el estándar `UNE 20324` [78] le otorga una protección casi total contra la penetración del polvo, así como un aislamiento contra los chorros directos de agua[12], más que suficiente para resistir la lluvia más fuerte.

[78] UNE 20324:1993 - Grados de protección proporcionados por las envolventes (Código IP). (CEI 529:1989).

Además tiene una certificación *IK7* que le otorga una resistencia considerable ante golpes directos en su partes consideradas más débiles, más concretamente, capaz de soportar el impacto de una masa esférica de acero de 500g arrojada desde una altura de 400mm sin generar ningún tipo de desperfecto.

Finalmente esta caja dispone de un aislamiento externo Clase II, que le dota de doble aislamiento eléctrico sin necesidad de toma tierra.

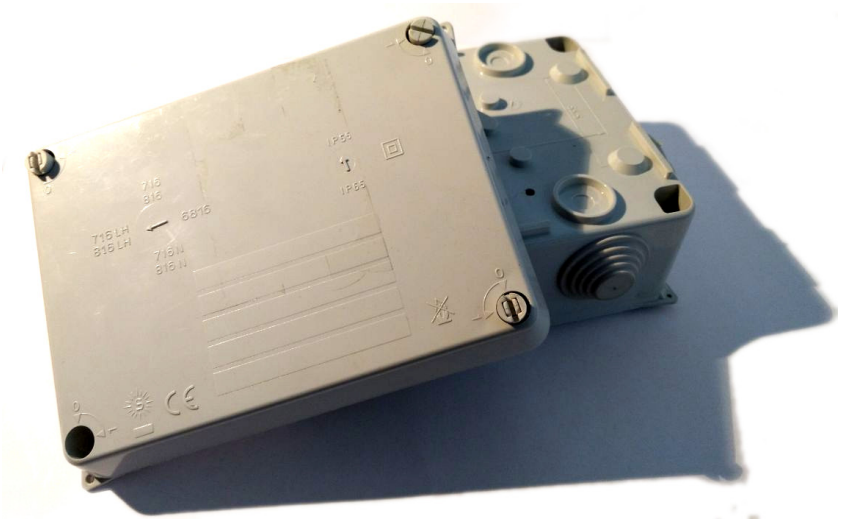


Figura 5.32
Caja estanca empleada en el prototipo para contener los diversos elementos hardware.

Para acoplar la placa y los otros elementos hardware dentro de la caja, se emplea poliestireno extruido (o `XPS` o `styrofoam`) para realizar una pieza compacta. El `XPS` es un excelente aislante térmico, no conductor y presenta una baja absorción de humedad[13]. Además, resulta ligero y fácilmente modelable. Se realiza un diseño CAD (Figura 5.33) del soporte a contener los elementos con un frontal protector.

12 ↑El siguiente grado de certificación es el aplicado a los objetos a ser resistentes contra la mar gruesa.

13 ↑Inferior al 0.7 % a inmersión total

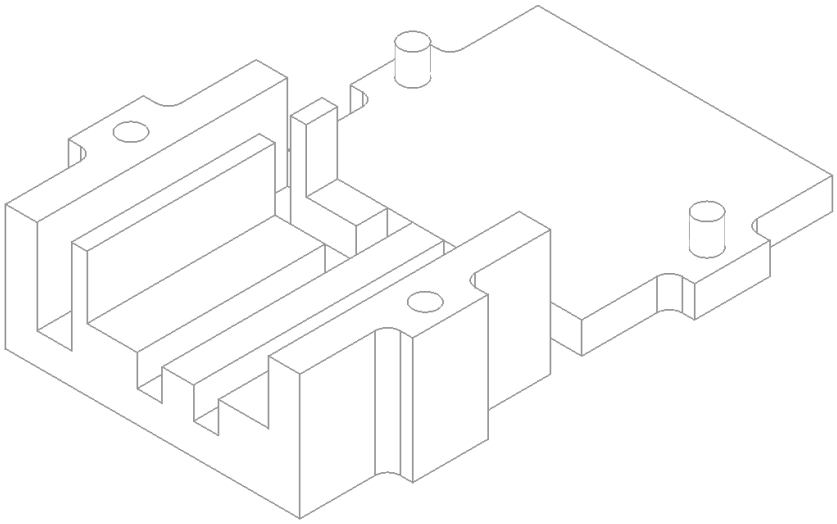


Figura 5.33
Diseño CAD del soporte y aislante de la placa `Raspberry Pi`.

Mediante el uso de un programa de automatizado en CAD se realiza el diseño del soporte y se subcontrata al Laboratorio de Mecatrónica el automatizado del fresado para la obtención de las diversas piezas mediante una plancha de `XPS` (Figura 5.34).

Figura 5.34
Proceso del fresado del soporte y aislante del prototipo.

Se obtienen dos piezas que encajan la una en la otra formando un cierre (Figura 5.35). La pieza principal (más grande) contiene a los elementos más grandes como la placa `Raspberry Pi` y la tarjeta de red WiFi. Los huecos y espacios están diseñados para que estos elementos queden sujetos de forma firme y segura. Además, hay hendiduras por donde pueden pasar los cables que conectan los distintos elementos de forma ordenada. La pieza pequeña es una tapadera frontal que encaja sobre la pieza más grande y tiene tres funciones principales, ya que sirve de aislante térmico hacia el calor externo así como de disipador del calor generado por la placa. Sirve también para ocultar los componentes ante posibles aperturas de la caja por parte de vándalos y, por último, proporciona un espacio donde es fácil anotar información relativa al nodo, como su emplazamiento.

Figura 5.35
Fotografía del soporte y aislante de la placa, donde se emplazan los distintos elementos hardware del prototipo.

El soporte encaja dentro de la caja estanca, y permite albergar los componentes de forma sólida y ordenada (Figura 5.36). La placa y los componentes encajan perfectamente dentro del espacio creado entre las dos piezas, creando un sólido compacto al tapar la pieza pequeña sobre la más grande. Por último se cierra la tapa estanca de la caja. Adicionalmente, se puede añadir pequeños candados sobre las esquinas de la caja, para garantizar que no pueda ser manipulada.

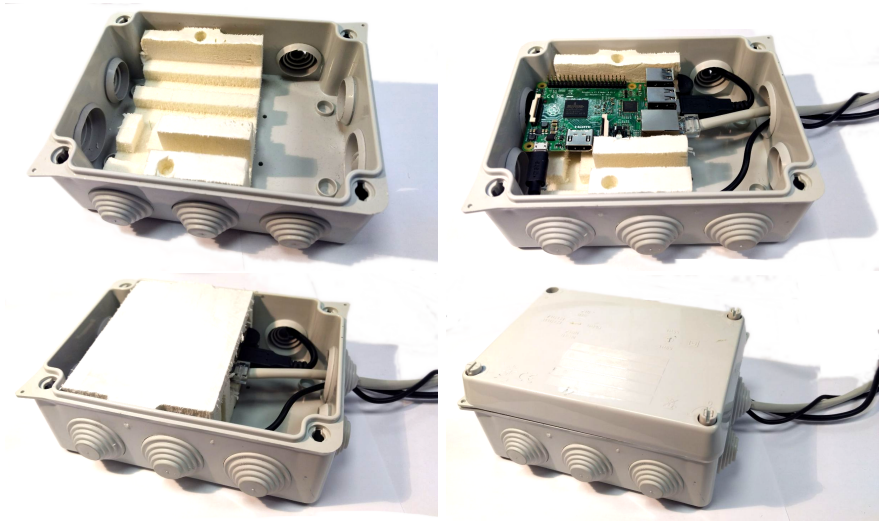


Figura 5.36
Acoplamiento de la placa y los componentes en la caja estanca, empleando el soporte y aislante diseñados en CAD.

La elección de una caja estanca corriente, en lugar de un soporte más estético, es debido a que este tipo de cajas son muy empleadas en las instalaciones eléctricas, por lo que permite disimular el emplazamiento del **nodo**, al poder situar este cerca de otras cajas estancas que ya existan en los edificios (Figura 5.37). De esta forma, el **nodo** de monitorización pasa de forma inadvertida para las personas, que están acostumbradas a ver este tipo de cajas. Ade-

más, al estar vinculadas estas cajas con instalaciones eléctricas, suponen un elemento disuasorio para que nadie se plantee manipularlas.

Figura 5.37
Instalación de los **nodos** de monitorización en interiores, empleando cajas estancas ubicadas cerca de otras cajas similares para camuflarlas.

En exteriores, la caja estánca permite emplazar un elemento compacto en diversos escenarios para realizar pruebas. De esta forma, las primeras pruebas en entornos urbanos se realizaban dentro de los armarios de comunicación (Figura 5.38).

Figura 5.38
Instalación provisional del **nodo** sensor en entornos urbanos, ubicado dentro de los armarios de conexiones.

En entornos urbanos fue posible instalar los **nodos** de monitorización dentro de los semáforos (Figura 5.39) emplazando las cajas estancas dentro de módulos individuales situados encima de las luces. El frontal de dichos módulos, está construido en plástico, de forma que no interfiere con las comunicaciones inalámbricas, al contrario que el resto del cuerpo del semáforo, que logra apantallar todo el tránsito de dispositivos que no esté situado frente al semáforo.

Figura 5.39
Instalación del **nodo** de monitorización en el interior de un semáforo en un escenario urbano, el frontal de la caja es de plástico, por lo que permite la repcepción de las antenas sin interferencias, a la vez que el cuerpo metálico apantalla todo el tráfico que no se encuentre delante del semáforo.

Para finalizar, el empleo de tanto cajas estancas como de los soportes de XPS permite convertir a todos los elementos hardware de los **nodos** en elementos compactos y de fácil transporte, lo cual facilita en gran medida su implantación (Figura 5.40).

Figura 5.40
Transporte de los **nodos** de monitorización. Los **nodos** se pueden transportar tanto dentro de la caja estanca (para ser instalados ambos al mismo tiempo) como dentro del soporte para ser acoplados a cajas estancas ya instaladas.

Este uso modular ha permitido que los nodos de monitorización, en última instancia, puedan ser instalados y emplazados por personas ajenas al proyecto, lo que ha permitido, que puedan ser emplazas en mejores zonas. En la Figura 5.41 se puede presenta una fotografía de los técnicos de la DGT colocando un nodo de monitorización en carretera, al lado de un elemento de señalización electrónica.

Figura 5.41
Emplazamiento de un nodo de monitorización en carretera por parte del personal de la DGT.

Al no requerir el nodo de monitorización de contacto directo con la vía permite una gran versatilidad en cuanto a lugares donde ser emplazado se refiere.

5.4.8 *Estudios relativos al hardware*

A continuación se recogen los estudios hechos para justificar el bajo coste, tanto económico como energético del hardware del prototipo.

Estudio 5.4.1: Coste del hardware de monitorización

Entre los requisitos no funcionales presentados en la Sección 5.2.1 se ha mencionado que el prototipo de sistema debe de ser de bajo coste para que su implantación pueda ser asumible.

El coste objetivo del prototipo funcional se sitúa por debajo los 100€ (Tabla 5.3) en lo relativo al coste del hardware implicado en él. A este precio habría que sumar los costes de mano de obra de implantación del dispositivo.

Tabla 5.3
Precio de los elementos que componen el hardware del **nodo** de monitorización.
Precios a fecha de Mayo de 2018.

	PVP	PRECIO BRUTO
Componentes principales		
Raspberry Pi 2	34.95€	28.88€
Tarjeta de red WiFi	12.55€	10.37€
Tarjeta de red BT	11.15€	09.21€
Tarjeta microSD	12.00€	09.92€
Cable alimentación 1A	08.35€	06.90€
Caja estanca	04.50€	03.72€
Soporte XPS	02.00€	01.65€
TOTAL	85.50€	70.65€
Componentes adicionales		
Módulo de reloj	2.15€	01.77€

Debido a que el **nodo** no es intrusivo tal y como se ha presentado en la Sección 3.4 su instalación no requiere obra civil y puede ser fácilmente implantando en la zona a monitorizar en pocos minutos, ya que en la mayoría de los casos únicamente requiere anclar la caja estanca (o ubicarla) y conectar las tomas de red eléctrica e internet.

Sobre los costos asociados a la mano de obra para el ensamblado de los componentes y la instalación y configuración del software, ambos procesos pueden ser realizados en unos 15 minutos por nodo, de los cuales la mayor parte del tiempo se implica en la copia de la imagen del sistema operativo y software a la tarjeta microSD.

Estos costes son los asociados al prototipo más básico con componentes adquiridos comercialmente, en un futuro con fabricación a medida, estandarización del proceso y adquisición de componentes en masa, es factible un abaratamiento significativo del **nodo**.

Estudio 5.4.2: Consumo energético del prototipo

El bajo coste del **nodo** de monitorización no implica solamente al coste de sus componentes, sino también al consumo energético del mismo. El prototipo haciendo uso de ambas interfaces de red consume un promedio de $2Wh$ (Tabla 5.4).

Tabla 5.4
Consumo energético de los componentes del hardware del **nodo** de monitorización.
Bajo una corriente de 5V.

	Potencia instantánea			Consumo
	min	avg	max	medio
Raspberry Pi 2	240mA	290mA	430mA	1.45Wh
+ Tarjeta de red WiFi	290mA	380mA	540mA	1.90Wh
+ Tarjeta de red BT	270mA	320mA	430mA	1.60Wh
+ Tarjeta de red WiFi + BT	340mA	400mA	660mA	2.00Wh
+ Tarjeta de red WiFi + BT + Pantalla	570mA	610mA	710mA	3.05Wh

COSTE ANUAL NODO+BT+WIFI (*)	0.2898€

(*) Supuesto un coste del kWh de $0.12925€/kWh$ y una potencia contratada de 3.45.

Este consumo es equivalente al de 30 bombillas tradicionales de 60W enchufadas todo el día o al de 4 bombillas leds equivalentes de 8W.

Para contextualizar el coste anual del consumo energético del **nodo**, según los prespuestos del Ayuntamiento de Granada para el años 2016 [216] el coste del alumbrado público asciende a $3718915,61€$ y a $156.671,46€$ para los semáforos e instrumentos asociados al control del tráfico. Añadir 100 **nodos** de monitorización basados en la actual tecnología aumentaría la factura eléctrica en menos de 30€ al año.

5.5 HOREB: SISTEMA OPERATIVO

Una de las principales ventajas de Raspberry Pi como SBC es que permite ejecutar un Sistema Operativo completo, con todos los componentes software que son habituales, además de relegarle ser el encargado de la comunicación e integración con el hardware. Disponer de un sistema operativo para la gestión del hardware permite centrar el desarrollo del sistema en los componentes software específicos.

Debido a la popularizad de Raspberry Pi, existen infinidad de sistemas operativos disponibles, la gran mayoría de ellos basados en GNU/Linux. Existen dos tipos de sistemas operativos, los de propósito general y los especializados, centrándose estos en convertir la Raspberry Pi en un entorno amigable con el usuario para un fin concreto, ya sea convirtiéndola en un sistema multimedia[14], un sistema de juegos[15] o o cualquier otro uso final.

Entre los sistemas operativos de propósito general más frecuentes para Raspberry Pi se encuentran los siguientes (Tabla 5.5).

Tabla 5.5
Sistemas operativos de propósito general más comunes para Raspberry Pi

NOMBRE	BASADO EN	ENLACE
Raspbian	GNU/LINUX	http://www.raspbian.org
Snappy Ubuntu Core	GNU/LINUX	https://developer.ubuntu.com/core
Ubuntu Mate	GNU/LINUX	https://ubuntu-mate.org/raspberry-pi/
Windows 10 IOT Core	WINDOWS	https://developer.microsoft.com/es-es/windows/iot
OpenSuse	GNU/LINUX	https://en.opensuse.org/HCL:Raspberry_Pi3
Arch Linux	GNU/LINUX	https://archlinuxarm.org/platforms/armv6/raspberry-pi
Pidora	GNU/LINUX	http://pidora.ca/
Risc OS	GNU/LINUX	https://www.riscosopen.org
RaspBSD	BSD	http://www.raspbsd.org/

La elección entre uno u otro sistema operativo basado en GNU/Linux suele obedecer más a las predilecciones de cada usuario sobre una distribución concreta que a la supremacía de uno u otro sistema. Sin embargo, para el desarrollo del prototipo se ha optado por emplear Raspbian, pues dispone de una enorme comunidad de usuarios así como de miles de paquetes[16] preparados y optimizados para Raspberry Pi. Además, de que dispone de que ofrece una versión "lite", sin entorno de escritorio, al contraria que otras que intentan ofrecer un sistema operativo de escritorio con el fin de convertir el SBC en un ordenador personal.

Raspbian es un port no oficial de Debian armhf[17] para Raspberry Pi con soporte optimizado para cálculos en coma flotante por hardware. Concre-

14 ↑Raspmc,Kobi,Xbian,OpenELEC+XBMC
15 ↑RetroPie
16 ↑Más de 35.000.
17 ↑https://www.debian.org/releases/jessie/armhf/

tamente la versión empleada en el prototipo es Raspbian GNU/Linux 7 de sobrenombre Wheezy. La versión del kernel de linux que empleada es la 4.0.9-v7+[18].

Aunque existen diferentes instaladores [19] que permiten realizar de forma más amigable la instalación del sistema operativo en la tarjeta microSD, se opta por emplear un copiado directo (Código 5.1) mediante dd.

Código 5.1
Copiado de la imagen ISO de Rasbian en la microSD

```
1  dd if=<fichero-raspbian.img> of=<ruta montaje microSD> bs=1M
```

Debido a que Raspbian es una distribución de GNU/Linux derivada de Debian, sigue la misma jerarquía de ficheros y funcionamiento que su distribución madre (Figura 5.42). Sin embargo, esta configuración no en todos los casos es la más optima para la ejecución en una SBC como Raspberry Pi, sobre todo si se quiere lograr un entorno de ejecución 24x7 robusto y confiable.

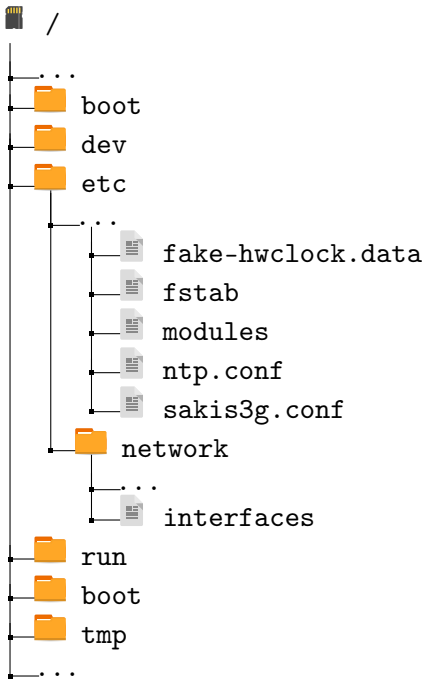


Figura 5.42
Ficheros empleados en la configuración del Sistema Operativo Raspbian

Para lograr que el sistema operativo sea lo más robusto posible, se tienen que modificar y configurar correctamente. En el ámbito de esta tesis, el

18 ↑V7+ indica que es la versión compilada para el conocido como kernel7 de Raspberry Pi, en contraposición con el kernel a secas. Esto es debido al cambio de arquitectura ARM que se dio en la Raspberry Pi 2, donde se pasó a emplear un chitsep ARMv7. Para garantizar la retrocompatibilidad con las placas antiguas, los sistemas operativos suelen disponer de ambas versiones del Kernel.

19 ↑Por ejemplo, NOOBS o BERRYBOOT.

sistema operativo `Raspbian` modificado se denotará **HOREB** para facilitar la nomenclatura, distinguiéndolo del sistema operativo original `Raspbian`.

La necesidad de estas modificaciones, principalmente, es debido a que el sistema operativo se ejecuta sobre una tarjeta microSD y estas tiene una vida limitada, por lo que de no hacer las modificaciones y configuraciones pertinentes, la esperanza de vida del **nodo de monitorización** se vería comprometida.

5.5.1 *Lidiando con la vida limitada de las microSD*

Las tarjetas microSD y en general las memorias SD son un tipo de memoria FLASH encapsuladas y por ello disponen de un número limitado de ciclos de escritura para cada sector físico [25]. Según las indicaciones del fabricante, las tarjetas microSD empleadas en el prototipo (Sección 5.4.5) tienen una vida garantizada de 30 000 ciclos de escritura por sector físico [263]. Esto indica que cada sector de la memoria flash puede ser borrado unas 30 000 antes de que se empiece a deteroriorar, haciendo peligrar la integridad del mismo para el almacenamiento.

[25] Introduction to flash memory

[263] Flash Memory Guide: Portable Flash memory for computers, digital cameras, mobile phones and other devices

Debido a esta caducidad de las tarjetas microSD debido a su limitación en el número de escrituras, es frecuente que este sea uno de los componentes que más problemas genera en cualquier proyecto que implique a una `Raspberry Pi`. Sirva de ejemplo, que de los más de 28 000 hilos que tiene el foro oficial de resolución de problemas de `Raspberry PI`, en más de 21 500 hilos el causante del error es la tarjeta microSD, siento en más de 3 000 que la tarjeta se ha corrompido y la única solución posible emplear una nueva[20].

La tarjeta microSD, por tanto, es un componente de vital importancia para el correcto funcionamiento del **nodo de monitorización**, ya que en ella se almacena tanto el sector arranque, el Sistema Operativo y demás componentes Software a ejecutar por la `Raspberry Pi`. Como se pretende un sistema robusto y tolerable a fallos (Sección 5.2), lidiar con la caducidad de la tarjeta microSD es un problema prioritario para minimizar los errores a largo plazo del **nodo**.

Además la placa `Raspberry PI` requiere de alimentación eléctrica para funcionar, siendo las operaciones de escritura en la tarjeta algo arriesgado, debido a la suceptibilidad de que se produzca un fallo eléctrico que provoque un corte en la alimentación, quedando corrupta la tarjeta microSD debido al reinicio del sistema.

Para lidiar con estas limitaciones se presenta a continuación las acciones acometidas.

20 ↑Datos por el motor de búsqueda Google

5.5.1.1 *Particiones en ram*

La mayoría de las operaciones de escritura que emprende un sistema operativo UNIX son dedicadas a generación de ficheros de bitácora (o LOG) en texto plano. A las bitácoras del sistema Operativo, hay que sumar los ficheros de LOG generados por el software de monitorización (Sección **??**).

Para minimizar el impacto del LOG en la vida de la tarjeta microSD, se emplean particiones TMPFS [250] para albergar dichos ficheros. Este tipo de particiones emplean la memoria volátil o RAM como si fuese un sistema de archivos, permitiendo que la velocidad de escritura se incremente considerablemente (Experimento **??**).

[250] tmpfs: A virtual memory file system

Con esta medida se minimizan el desgaste de la tarjeta microSD por agotamiento del número máximo de ciclos de escritura. Además, se quita la carga de escritura de los LOGs al sistema, siendo una operación de entrada y salida innecesaria en caso de correcta ejecución.

Por contra, en caso de errores de alimentación eléctrica o reinicios del sistema operativo se pierde la posibilidad de depurar en base a estos ficheros, pues al estar en particiones TMPFS se borran en cada reinicio . En la Secciones 5.6.5.11 y **??** se presentará como desde el software de monitorización se ha intentado paliar esa limitación. Sin embargo, debido a que los fallos eléctricos no son causados por fallos software que puedan ser depurados por los ficheros de LOG y los fallos que ocasionen reinicios del sistema no son competencia del software del monitorización, se opta por las particiones TMPFS para los ficheros de LOG debido a su beneficio para mejor la vida de las tarjetas microSD.

Igualmente, para la partición temporal (o/tmp/) se opta por este mismo tipo de partición, debido a la utilización de este fichero en el sistema de actualización remota de software que se presentará en la Sección 5.6.5.9.

5.5.1.2 *Tarjeta sólo lectura*

Con las particiones TMPFS se limita la mayoría de las escrituras en disco del sistema operativo, pero no todas. Si se desea que el sistema sea robusto ante fallos de corriente o reinicios, es necesario que la placa disponga en todos sus reinicios de todos los ficheros del sistema operativo en perfecto estado (sin corromper).

Para ello se propone que el resto de directorios del sistema operativo sean montados siempre en modo de solo lectura, de forma que sea imposible modificar ninguno de estos ficheros en tiempo de ejecución con el fin de preservar siempre su estado original.

Sim embargo, puede ser necesario que en caliente el sistema modifique alguno de estos ficheros, ya sea para cambiar una configuración o aplicar algún parche de seguridad del sistema operativo o alguno de sus componentes (Sección 5.6.5.10). Para ello se dota al sistema operativo de `scripts` que

permiten volver a montar en caliente la partición principal ya sea en modo sólo lectura (Código5.3) o en modo escritura (Código5.2).

Código 5.2
Script para remontar en modo escritura para realizar cambios en caliente.
Fichero: /usr/sbin/remountRW.sh

```
1  sudo mount -o remount,rw /dev/root /
```

Código 5.3
Script para remontar el sistema de archivos en modo sólo lectura
Fichero: /usr/sbin/remountRO.sh

```
1  sudo mount -o remount,ro /dev/root /
```

Con esta medida se consigue que la tarjeta microSD contenga todos los ficheros necesarios del sistema operativo, en su estado original para garantizar su arranque en todo momento.

5.5.1.3 *Sistema no-atime*

Debido a que el el sistema de archivos se ha configurado a modo de solo lectura de forma que ninguna escritura pueda hacerse, se puede deshabilitar el sistema `ATIME` encargado de actualizar la fecha de último acceso de cada de fichero. Esto supone que cada lectura del disco es acompañada con una escritura para actualizar la fecha.

Al estar en un sistema de sólo lectura, la actualización de esta fecha no puede ser realizada, por tanto es recomendable desactivarla.

5.5.1.4 *Partición para el almacenamiento de los datos*

Cómo se presentó en la Figura 5.19, debido a que el sistema ha de ser capaz de funcionar de forma autónoma incluso en el caso de que no exista la comunicación con el servidor, se hace necesario dotar de un almacenamiento adicional para almacenar los **pasos de dispositivos** detectados que no han podido ser enviados.

Para ello se emplea una partición de la tarjeta, que estará en modo escritura, al contrario que la partición principal del sistema que está en modo sólo lectura como se ha presentado anteriormente. Si bien es deseable que dicha partición no sea empleada, pues en caso ideal la comunicación con el servidor no debería fallar, su empleo en entornos incomunicados o para pruebas es de vital importancia.

5.5.1.5 *Copiado de los pasos en caliente*

En entornos en los que el nodo no funcione con una comunicación al servidor, se pueden extraer los pasos capturados mediante el empleo de un pendrive USB. Esta acción es realizada de forma automática mediante la definición de una regla `udev`[21] (Código 5.4) que ejecuta un `script` (Código 5.5) al detectar un dispositivo de almacenamiento conectado a la placa.

Código 5.4
Regla `udev` para el volcado de los pasos al conectar una memoria USB a la placa
Fichero: `/etc/udev/rules.d/60-usb_backup.rules`

```
1 ACTION=="add",KERNEL=="sda*",RUN+="/usr/bin/llaveRaziel.sh"
```

Código 5.5
`Script` para el volcado de los pasos y logs a la memoria USB
Fichero: `/usr/bin/llaveRaziel.sh`

```
1  #!/bin/sh
2  NAME=$(cat /usr/share/raziel/configuration.properties | grep nombreNodo | cut -f 2
   ↪  -d "=")
3  NAME=$NAME-$(date +%Y%m%d_%H%M%S)
4  sudo mount -o rw /dev/sda1 /mnt/;
5  echo $NAME>>/mnt/listado.txt
6  sudo rm -R /mnt/log/$NAME;
7  sudo mkdir -p /mnt/log/$NAME;
8  sudo cp -R /var/log/ /mnt/log/$NAME;
9  sudo cp /usr/share/raziel/*.csv /mnt/log/$NAME/;
10 sudo umount /mnt
```

A pesar de este `script` es muy útil en las fases de testeo del dispositivo, ya que además permite copiar los ficheros de log, en producción se encuentra por defecto deshabilitado para evitar posibles robos de información o manipulaciones. Sin embargo, es fácilmente modificable, ya únicamente requiere la modificación del `script` *llaveRaziel.sh* (Código 5.5).

Por último, este procedimiento es fácilmente securizable, añadiendo más `script` sobre el dispositivo USB a conectar a la regla `udev` o modificando el script para que requiera un fichero clave con un token de identificación en la memoria USB con información sólo conocible por los desarrolladores. Sin embargo, esta línea de securización no ha sido continuada, principalmente porque para explotarla por agentes externos requiere tener acceso físico al **nodo**, lo cual implicaría riesgos mucho mayores.

21 ↑http://www.reactivated.net/writing_udev_rules.html

5.5.1.6 *Conclusiones sobre el sistema de archivos*

Todas las modificaciones realizadas sobre el sistema de archivos son realizadas mediante el fichero `/etc/fstab` (Código 5.6) que indica como montar cada dispositivo y que configuración emplear para cada uno de ellos.

Código 5.6
Configuración del sistema de archivos
Fichero: `/etc/fstab`

```
root@raspberrypi:/usr/share/raziel/scripts# cat /etc/fstab
proc            /proc           proc    defaults          0       0
/dev/mmcblk0p1  /boot           vfat    defaults          0       2
/dev/mmcblk0p2  /               ext4    defaults,noatime,ro,errors=remount-ro  0
↪ 1
/dev/mmcblk0p3
↪ /usr/share/raziel        ext4        defaults,noatime        0        3
tmpfs         /tmp            tmpfs   nodev,nosuid,size=15M,mode=1777  0  0
tmpfs         /var/log        tmpfs   nodev,nosuid,size=30M,mode=1777  0  0
```

Se dedican para las particiones en RAM `TMPFS` 30*MB* para la partición de `LOGs` y 15*MB* para la partición temporal, lo cual sumado al resto de particiones RAM empleadas por el sistema operativo dejan un total de 862*MB* de Memoria RAM de la placa para uso del software en ejecución.

Esta cantidad de memoria es más que suficiente, por lo que se decide prescindir de una partición `SWAP`[22] para albergar el archivo de paginación. Esta decisión obedece también a la minimización de las escrituras en la tarjeta microSD con el fin de aumentar su vida útil. Para ello, existen multitud de variantes, prefiriéndose la de directamente remover esta característica del sistema operativo (Código 5.7)

Código 5.7
Deshabilitado de la memoria virtual `SWAP`

```
sudo dphys-swapfile swapoff
sudo dphys-swapfile uninstall
sudo update-rc.d dphys-swapfile remove
```

Finalmente, las 3 particiones en la tarjeta microSD (Código 5.8) se les dedica el siguiente espacio:

BOOT (`60MB`) partición para el gestor de arranque del sistema operativo.

HOREB (`5.5GB`) partición donde se alojan los principales componentes del sistema operativo.

RAZIEL (`10GB`) partición para el software RAZIEL y los datos capturados no enviados.

22 ↑Memoria virtual que aumenta la cantidad de memoria volatil disponible en el sistema, empleando parte de la memoria ROM como archivo de paginado

Código 5.8
Particiones del sistema de archivos

```
pi@raspberrypi /etc/network $ sudo fdisk -l

Disk /dev/mmcblk0: 16.0 GB, 16022241280 bytes
4 heads, 16 sectors/track, 488960 cylinders, total 31293440 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0002c262

        Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1            8192      122879       57344    c  W95 FAT32 (LBA)
/dev/mmcblk0p2          122880    10801151     5339136   83  Linux
/dev/mmcblk0p3        10801152    30449663     9824256   83  Linux
```

Si se contabilizan los bloques empleados por todas las particiones, se aprecia que no se emplea la totalidad de los bloques existente (Figura 5.43). Esto es debido a la apreciación de que no todas las tarjetas microSD empleadas (e incluso siendo del mismo fabricante) disponen del mismo tamaño real. Se opta por tanto por dejar unos $500MB$ libres de margen al final de la tarjeta microSD. Como la instalación del sistema operativo se realiza por copia directa (Código 5.1), en caso de que la imagen a copiar resultase mayor que el espacio real disponible en la tarjeta microSD se produciría un error.

Figura 5.43
Espacio libre al final de la tarjeta microSD existente para facilitar la copia de la imagen del sistema operativo a las tarjetas microSD, ya que no todas disponen exactamente del mismo tamaño.

Una vez realizada la copia en la tarjeta microSD, se puede emplear la herramienta `raspi-config` para ampliar las particiones de forma que hagan uso de todo el espacio disponible (Figura 5.44).

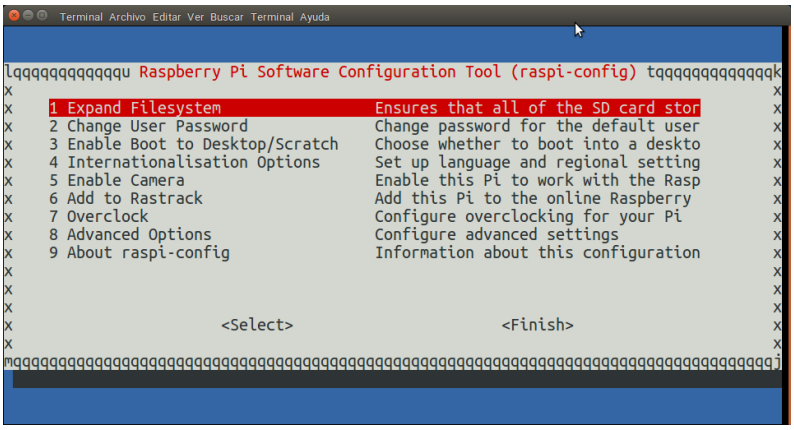


Figura 5.44
Expansión del sistema de archivos mediante `raspi-config`, para emplear el espacio libre dejado al final de las tarjetas microSD debido a que no todas disponen del mismo número de bloques.

5.5.2 *Controladores adicionales*

Una de las ventajas de tener un sistema operativo en la SBC es que se puede relegar la comunicación con el hardware a los controladores o drivers existentes. De esta forma, se emplean herramientas muy testeadas y comprobadas liberando esta carga de desarrollo del prototipo.

Tarjeta de red Bluetooth

Para la comunicación con la tarjeta de red Bluetooth (Sección 5.21) se emplea el conjunto de software, herramientas y demonios disponibles en el meta-paquete `bluetooth` [23] de Debian.

Tarjeta de red WiFi

Para la tarjeta de red WiFi (Sección 5.23) se emplea el meta-paquete `firmware-atheros`[24] que, entre otros, incluye los binarios del firmware para esta tarjeta.

Para habilitar el modo monitor y gestionarlo (Sección 4.3.4), se recomienda la instalación del el meta-paquete `aircrack-ng`[25] para hacer uso de su script `airmon-ng`[26]. El empleo de este paquete, permite que las capas más elevadas del software (Sección 5.1) sean más independientes de la tarjeta de red y sus drivers asociados, dotando al software de cierta independencia respecto al hardware.

Para iniciar el modo monitor en la interfaz de red `wlan0` mediante el empleo de este script, tan sólo se requiere una invocación del mismo `airmon-ng start wlan0`.

23 ↑https://packages.debian.org/wheezy/bluetooth
24 ↑https://packages.debian.org/wheezy/firmware-atheros
25 ↑https://packages.debian.org/es/wheezy-backports/aircrack-ng
26 ↑https://www.aircrack-ng.org/doku.php?id=es:airmon-ng

Tarjeta de red 3G/2G

Muchas tarjetas de red 3G/2G, como la empleada en este prototipo (Sección **??**) disponen de un almacenamiento adicional que contiene los drivers para el empleo de los modems en sistemas Windows, Linux y Mac (Figura 5.45). De esta forma, al conectarla por primera vez el sistema operativo detecta una unidad de almacenamiento masivo, instala los drivers mediante un `autorun` que requiere confirmación del usuario y ya con los drivers instalados en el sistema operativo, empieza a emplear la tarjeta como modem 3G.

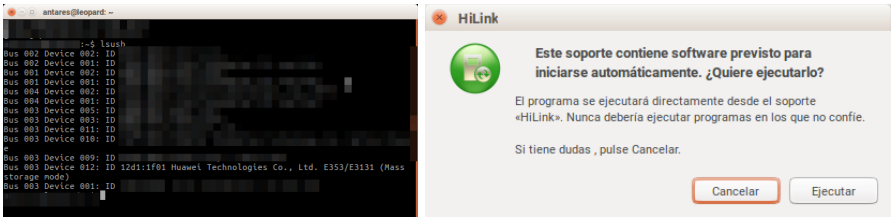


Figura 5.45
Tarjeta 3G montada como unidad de almacenamiento masivo y ventana de confirmación de instalación de drivers en entornos de escritorio.

En LINUX existe el proyecto `usb_modemswitch` [27] que permite realizar este proceso de forma automática para la mayoría de dispositivos USB que disponen de varios modos de funcionamiento. Se basa en una regla udev (Código 5.9) que se ejecuta por el sistema operativo al detectar el dispositivo deseado identificado por los códigos estándar del par vendedor-producto y ejecuta el script `usb_modeswitch`.

Código 5.9
Regla udev de usb_modemswitch

```
1 SUBSYSTEM=="usb",
2 ATTRS{idVendor}=="12da", ATTRS{idProduct}=="1f01", RUN+="usb_modeswitch '%b/%k'"
```

Para el correcto funcionamiento del script debe existir un fichero[28] nombrado como el par vendedor-producto. Este fichero contiene el mensaje binario que ha de enviarse al hardware para activar el cambio de modo, así como el nuevo par vendedor-producto que mostrará la tarjeta de red 3G una vez activado el cambio de modo (Código 5.10).

27 ↑ http://www.draisberghof.de/usb_modeswitch/
28 ↑Alojado en `/usr/share/usb_modeswitch`.

Código 5.10
Fichero 12d1:1f01 con el mensaje enviado para activar el cambio de modo y los nuevos identificadores.

```
# Huawei E353 (3.se)

TargetVendor=  0x12d1
TargetProduct= 0x14db

MessageContent="555342431234567800000000000000a1106200000000000100000000000000"
NoDriverLoading=1
```

Una vez se remonta la tarjeta 3G como modem, el módulo `usb_wwan`[29] se encarga de la comunicación con la tarjeta.

5.5.2.1 *Deshabilitar el ahorro de energía de los dispositivos USB*

Una de las prioridades común en la mayoría de los sistemas operativos es la de minimizar el consumo energético del hardware cuando este no se está siendo empleando. Sin embargo, en el **nodo** se ha observado que estos mecanismos dan errores con el software de captación, que hace un empleo constante del hardware, debido a que habitualmente el sistema operativo decide hibernar los dispositivos externos cuando cae su utilización.

Para deshabilitar esta funcionalidad del sistema operativo, se debe de modificar el fichero `/boot/cmdline.txt` (Código 5.11) cuyo cometido es similar al de la `BIOS` en un ordenador personal.

Código 5.11
Deshabilitado de la hibernación de los dispositivos USB
Fichero: /boot/cmdline.txt

```
pi@raspberrypi /boot $ cat cmdline.txt
dwc_otg.lpm_enable=0 console=tty1 console=ttyAMA0,115200 root=/dev/mmcblk0p2
↪  rootfstype=ext4 dwc_otg.speed=1 elevator=deadline rootwait usbcore.autosuspend=-1
↪  fbcon=map:10 fbcon=font:ProFont6x11 logo.nologo
```

29 ↑http://modules.libres.ch/browse/linux/v3.4/arm/usb_wwan/

5.5.3 *Interfaces de red*

El **nodo de monitorización** ha de ser capaz de comunicarse con el **servidor de computo** para transmitirle los pasos de dispositivos detectados. Los mecanismos de comunicación empleados serán descritos en la Sección 5.8, sin embargo, es necesario que para dicha comunicación las interfaces de red empleadas estén habilitadas.

Para ello se proponen varias alternativas, ya sea mediante el empleo de una red cableada ethernet, una conexión WiFi o mediante una comunicación 3G. Además, los parámetros de conexión pueden ser prefijados u obtenidos por medio del protocolo `DHCP`.

Finalmente, en entornos concretos se puede requerir que la conexión con requiera de algún mecanismos de seguridad adicional, como el empleo de proxys, pasarelas de red o VPNs. Este tema será tratado brevemente en el Sección 5.8.

5.5.3.1 *Conexión Ethernet*

El empleo de una conexión Ethernet para la comunicación entre el **nodo de monitorización** y el servidor de cómputo **servidor de computo** es la alternativa más deseable, sin embargo entre sus desventajas es que requiere de una infraestructura ya existente que dote de conectividad de red a la zona que se desea monitorizar (Tabla 5.6).

Tabla 5.6
Conexión por medio de cable Ethernet

VENTAJAS	DESVENTAJAS
• Método más robusto, seguro y tolerable a fallos. • Menor latencia y mayor ancho de banda (p.e. 1000 Base T) • No interfiere con el espacio radioeléctrico.	• Necesita el despliegue de cable de red • Requiere de infraestructura de red en la zona a monitorizar que pueda ser empleada • Satisfacer los requisitos de conexión a esa red y su empleo implica directamente a los administradores de dicha red.

Emplear una arquitectura de red ya existente, implica además tener que aceptar los requerimientos de los administradores y propietarios de las mismas. Esto a impuesto decisiones, como que el el **servidor de computo** sea físico y no esté alojado en la nube, pues para las administraciones el permitir la comunicación dentro de su red con una máquina ïndeterminada.[a]lojada en la nube es algo que implica una potencialidad de riesgo bastante elevada.

En entornos domésticos es más fácil emplear una configuración `DHCP` (Código 5.12) en la que las credenciales de conexión de red sean otorgadas por el punto de acceso.

Código 5.12
Conexión ETHERNET mediante DHCP:
Fichero: /etc/network/interfaces

```
1  ...
2  iface eth0 inet dhcp
3  ...
```

En entornos más securizados, las credenciales de red han de ser prefijadas en el **nodo** con anterioridad (Código 5.13)). Sin embargo esta manera de proceder implica que cualquier cambio en la arquitectura de la red, requiere que esta configuración sea modificada en el **nodo de monitorización** de forma presencial, ya que se perdería la conexión de red.

Código 5.13
Conexión ETHERNET mediante credenciales fijas:
Fichero: /etc/network/interfaces

```
1  ...
2  iface eth0 inet static
3  address 192.168.1.135
4  netmask 255.255.255.0
5  gateway 192.168.1.50
6  ...
```

La conexión mediante cable Ethernet ha sido la más empleada durante el desarrollo de esta tesis en los escenarios en los que se ha implantado el sistema de monitorización para estudiar la viabilidad de la fuente de datos.

5.5.3.2 *Conexión por WiFi*

Debido a que el **nodo de monitorización** ya incorpora una tarjeta de red WiFi para la monitorización de este tipo de dispositivos, puede ser empleada para establecer comunicación inalámbrica con el **servidor de cómputo**. Nuevamente se requiere de una infraestructura, inalámbrica en este caso, disponible para realizar esta conexión (Tabla 5.7).

Tabla 5.7
Conexión por medio de WiFi

VENTAJAS	DESVENTAJAS
• Método más robusto, seguro y tolerable a fallos. • No requiere de elementos físicos de interconexión.	• Requiere de infraestructura de red inalámbrica en la zona a monitorizar que pueda ser empleada. • Interfiere con el espacio radioeléctrico, lo cual perjudica la captura y monitorización. • Satisfacer los requisitos de conexión a esa red y su empleo implica directamente a los administradores de dicha red.

Sin embargo, emitir tramas WiFi para comunicar las tramas WiFi capturadas, supone una contaminación del espacio radioeléctrico, lo cual puede perjudicar enormemente la monitorización si se produce una saturación del mismo [162].

[162] Wireless internet access: 3G vs. WiFi?

En la Sección 4.3 se indicó brevemente que las comunicaciones inalámbricas WiFi pueden estar securizadas por varios protocolos distintos, requiriendo cada uno de estos unos parámetros y mecanismos de conexión distintos.

El método de securización más simple (actualmente desfasado y desaconsejado) es `WEP` (Código 5.14).

Código 5.14
Conexión WiFi mediante WEP y DHCP:
Fichero: `/etc/network/interfaces`

```
...
auto wlan0
allow-hotplug wlan0

auto wlan0
iface wlan0 inet dhcp
    wireless-essid testessid
    wireless-key 9 6f 75 72 20 70 61 73 73 77 6f 72 64
...
```

La mayoría de redes inalámbricas actuales emplean los protocolos `WPA-PSK` y `WPA2-PSK` (Código 5.14).

Código 5.15
Conexión WiFi mediante WPA-PSK y WPA2-PSK y credenciales fijas:
Fichero: `/etc/network/interfaces`

```
...
auto wlan0
allow-hotplug wlan0

iface wlan0 inet static
address 192.168.1.135
netmask 255.255.255.0
gateway 192.168.1.1
        wpa-ssid "mired"
        wpa-psk "miredsegura"
...
```

En entornos donde la seguridad es vital, se emplea el protocolo `WPA-EAP` (Código 5.14). Este requiere de un fichero (`supplicant`) con las credenciales de conexión.

Código 5.16
Conexión WiFi mediante WPA-EAP y DHCP:
Fichero: `/etc/network/interfaces`

```
...
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
...
```

Sin embargo, se desaconseja emplear la tarjeta de red WiFi para la monitorización y la conexión con el **servidor de cómputo**, ya que esta alternativa generado problemas en todos los escenarios en los que ha sido probado en esta tesis. La única alternativa encontrada para hacer viable esta conexión, ha sido restringir la comunicación inalámbrica a periodos de baja actividad de monitorización (p.e. de madrugada en edificios públicos).

5.5.3.3 *Conexión 3G*

Para la conexión 3G hace falta una tarjeta adicional de red 3G, así como una tarjeta SIM que lleve asociada una tarifa de transmisión de datos, lo cual implica un coste económico periódico (Tabla 5.8). Sin embargo este método no requiere de ninguna infraestructura de red adicional en la zona a monitorizar [30]

Tabla 5.8
Conexión por medio de 3G

VENTAJAS	DESVENTAJAS
• No require ningún tipo de infraestructura adicional.	• Alternativa más costosa. • Requiere de una tarjeta de red 3G adicional. • Necesaria una tarifa de datos, con coste periódico asociado. • La disponibilidad de la red no está garantizada. • La mayoría de los proveedores no permiten tráfico entrante a las SIMs.

5.5.3.3.1 Sakis3G

Para gestionar las comunicaciones 3G en LINUX existe el proyecto `Sakis3G`[31], que se integra como un servicio de sistema operativo más. En el Código 5.17 presenta el fichero de configuración de la conexión de red.

30 ↑Se presupone que la cobertura 3G actualmente abarca la mayoría de la geografía.
31 ↑https://github.com/Trixarian/sakis3g-source

Código 5.17
Configuración de la Conexión 3G:
Fichero: `/etc/sakis3g.conf`

```
MODEM="OTHER"
OTHER="USBMODEM"
USBMODEM="2357:0201"
USBINTERFACE="1"
CUSTOM_APN="ac.lolafone.es"
APN="CUSTOM_APN"
DIAL="*99#"
APN_USER="lolafone"
APN_PASS="lolafone"
USBDRIVER="option"
SIM_PIN="1234"
```

Sin embargo el proyecto `Sakis3G` se encuentra actualmente abandonado, por lo que el soporte a nuevos operadores de red o tarjetas 3G no es esperable por parte de la comunidad.

5.5.3.3.2 WVDIAL y PPP

`WVDIAL`[32] es una apliacción incluida en la mayoría de distribuciones de Linux que permite realizar conexiones a Internet por medio de módems empleando el protocolo Punto-a-Punto o `PPP` [247].

[247] The point-to-point protocol (PPP) for the transmission of multi-protocol datagrams over point-to-point links

`WVDIAL` carga su información de configuración del fichero `/etc/wvdial.conf` (Código 5.18). Dichero fichero contiene la información para establacer la conexión (puerto, velocidad y cadenas de iniciación o `init`), información del ISP[33], número de teléfono para la marcación y credenciales de autenticación.

32 ↑`http://freshmeat.sourceforge.net/projects/wvdial/`
`https://linux.die.net/man/1/wvdial`
33 ↑Proveedor de servicios de Internet

Código 5.18
Configuración de la Conexión 3G:
Fichero: /etc/wvdial.conf

```
[Dialer pin]
Init2 = AT+CPIN="0000"
Dial Attempts = 1

[Dialer check]
Init2 = AT+CREG?
Init2 = AT+COPS?

[Dialer Defaults]
Init1 = ATZ
Stupid Mode = 1
Init3 = ATQ0 V1 E1 S0=0
Init4 = AT+CGDCONT=1,"IP","inetd.vfes";
Username = { }
Password = { }
Phone = *99***1#
Modem Type = Analog Modem
Baud = 460800
New PPPD = yes
Dial Command = ATDT
Modem = /dev/ttyUSB0
ISDN = 0
Carrier Check = no
```

Para iniciar la conexión PPP, se marca con el módem al número indicado en la configuración y se manda una secuencia determinada de comandos AT [245]. Los comandos AT forman un lenguaje, desarrollado por Dennis Hayes, que se convertido en el estándar abierto no oficial de comandos para configurar y parametrizar módems, ofrececiendo comandos para el marcado, la gestión de la conexión y la manipulación de los parámetros de conexión. La mayoría de los módems actuales ofrecen soporte para estos comandos, aunque al no ser un estándar cerrado, existen multitud de variaciones sobre el mismo.

[245] *Data Communication Over the Telephone Network*

Además, en la primera comunicación con el modem, se debe de enviar el comando AT+CPIN para indicar el PIN, PUK o PH-PIN según el estado de bloqueo actual de la tarjeta SIM. El APN o nombre del punto de acceso, es un numbre resoluble mediante DNS que proporciona la IP que provee de servicio de red de datos a la red GPRS o 3G. El APN puede requerir credenciales de autenticáción, que deben de ser proporcionados en el fichero de configuración. La secuencia de comandos para el establacimiento de la conexión se detalla en la Figura 5.46.

Marcación al número `*99***1##`

`ATZ` Comienza la comunicación con el modem.

`AT+CPIN="0000"` Indica el pin de desbloqueo de la tarjeta SIM

`ATQ0 V1 E1 S0=0`

- `ATQ0` Habilita la transmisión de códigos de resultado.
- `V1` Establece formato de respuesta del modem.
- `E1` Habilita el `echo` para permitir el envío al modem de los caracteres recibidos por aplicaciones externas.
- `S0=0` Establece el número de tonos de llamada requeridos antes de la auto-respuesta a la llamada.

`AT+CGDCONT=1,"IP","INETD.VFES"` Especifica el contexto del `PDP`[a] y el `APN` [b] a emplear en la comunicicación.

a ↑Protocolo del paquete de datos
b ↑Access oint Name o nobre del punto de acceso

Figura 5.46
Secuencia de comandos AT para la conexión 3G

Una vez se ha iniciado el marcado, y se han enviado los comandos `AT` que negocian la conexión, el `ISP` proporciona las credenciales de red y se establece la conexión `PPP` que el sistema operativo puede emplear para acceder a internet. Los parámetros de esta conexión `PPP` se puede gestionar mediante el fichero `/etc/ppp/peers/wvdial` (Código 5.19).

Código 5.19
Configuración de la Conexión 3G:
Fichero: `/etc/ppp/peers/wvdial`

```
1 noauth
2 name wvdial
3 defaultroute
4 replacedefaultroute
5 debug
```

5.5.4 *Servidor de Fecha y Hora NTP*

Si bien `Raspberry Pi` es un excelente `SBC`, su principal limitación es que carece de un módulo de reloj que le permita mantener la hora sincronizada. Esta limitación conlleva que en entre arranques, no es capaz de mantener la hora correcta. Además, debido a que estima el tiempo mediante el reloj del procesador, su precisión a lo largo del tiempo se resiente.

Esta limitación se puede solventar de tres maneras distintas. Las dos primera de ellas son vía software y la última implica emplear los pines `GPIO` para incorporar un módulo de reloj (Sección 5.4.6).

5.5.4.1 *Empleo de `Fake hwClock`*

`Fake hwClock`[34] es un software disponible en Debian que almacena periódicamente la fecha del sistema en un fichero. Al iniciar el sistema, emplea la fecha almacenada como fecha del sistema. Si bien esto no permite solventar la carencia de un reloj, evita que el sistema arranque en 1970 ante un reinicio fortuito. La escritura de la fecha se realiza en un fichero en texto plano (Código 5.20) cuya ubicación es fija.

Código 5.20
Empleo de `fake-hwclock` para establecer la fecha y hora en el sistema
Fichero: `/etc/fake-hwclock.data`

```
1  2017-11-21 09:30:26
```

El estar constantemente escribiendo en el fichero la fecha atenta directamente contra el número limitado de ciclos de escritura que tienen las tarjetas SD (Seccíón 5.5.1), por lo que uso está desaconsejado.

Sin embargo, establecer este fichero de forma manual para la implantación de **nodos de monitorización** aislados o en pruebas puede ser un buen mecanismo para configurar correctamente la hora del sistema, aunque sea de forma aproximada.

5.5.4.2 *Empleo de `NTP`*

`Network Time Protocol` (o `NTP`) [184] es un protocolo para sincronizar los relojes de sistemas informáticos conectados en red.

[184] Network time protocol version 4 reference and implementation guide

En Debian se encuentra disponible mediante el demonio `ntpd`, que se ejecuta de forma periódica para mantener actualizado el reloj del sistema. La hora de referencia la obtiene de la comunicación con una lista de equipos accesibles dentro de la red. La lista de estos equipos puede ser configurada según las necesidades o restricciones de la red (Código 5.21).

34 ↑https://packages.debian.org/sid/main/fake-hwclock

Código 5.21
Configuración de NTP indicando la lista de servidores con los que sincronizar la fecha y hora
Fichero: /etc/ntp.conf

```
...
server 0.debian.pool.ntp.org iburst
server 1.debian.pool.ntp.org iburst
server 2.debian.pool.ntp.org iburst
server 3.debian.pool.ntp.org iburst
...
```

NTP[35] puede mantener los distintos equipos sincronizados con una precisión máxima de 10 milisegundos [183]. Si bien en algunos entornos este desfase de tiempo puede hacer que el sistema no sea suficientemente precisó, al hablar de desplazamientos entre magnitudes de varios metros, el margen de error fruto de la imprecisión de los relojes entre varios **nodos** es despreciable.

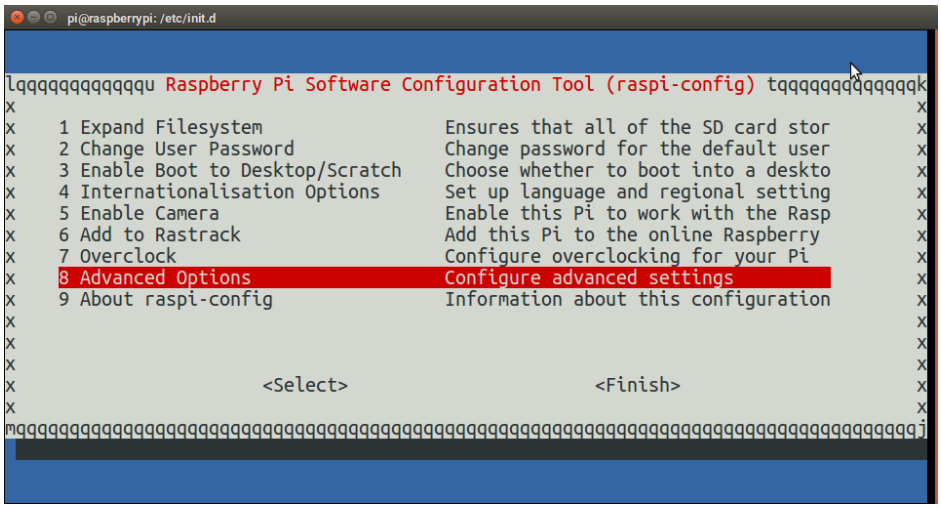
[183] Internet time synchronization: the network time protocol

El empleo de un servidor NTP es la solución más sencilla para le sincronización de los relojes, ya que el escenario más deseable para la implantación de un **nodo** es que este disponga de conexión con el **servidor de cómputo**, y este puede funcionar como servidor de NTP para que los relojes de los distintos **nodos** estén sincronizados.

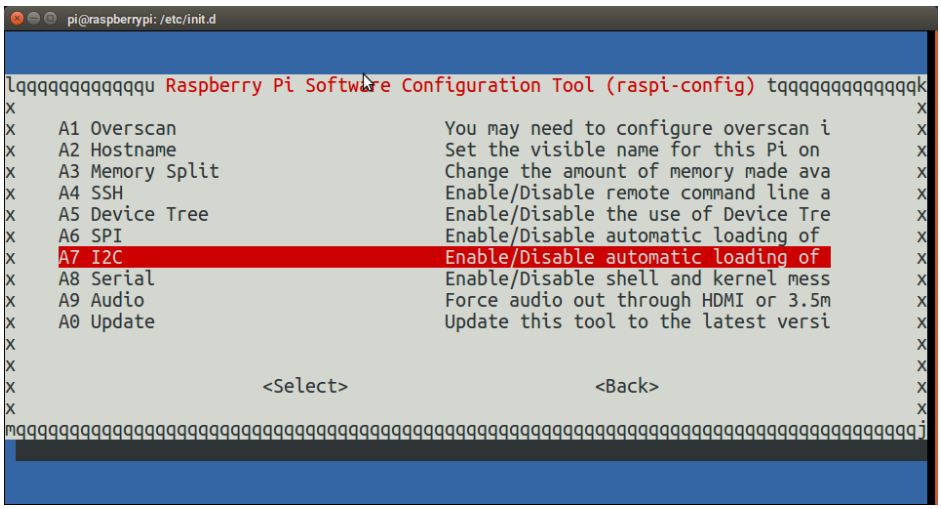
5.5.4.3 *Empleo de un módulo de reloj*

Para habilitar el empleo de un módulo de reloj en el **nodo**, como por ejemplo el módulo ZS-042 (Sección 5.29) se tiene que habilitar la interfaz I2C de los pines GPIO, lo cual puede hacerse fácilmente por medio de la herramienta `raspi-config` (Figura 5.47) preinstalada en Raspbian.

35 ↑En su versión 4 del protocolo y en adelante.



(a) Herramienta `raspi-config`



(b) Opción de habilitar `I2C`

Figura 5.47
Herramienta `raspi-config` para habilitar la interfaz `I2C` de los pines `GPIO`.

Para comprobar que la interfaz `I2C` está habilitada, puede comprobarse con `lsmod | grep i2c_`. Igualmente, para comprobar que existe comunicación con el hardware, puede emplearse el comando `sudo i2cdetect -y 1`.

Aunque el módulo `Z2-042` tiene un módulo `DS3231`, este no se encuentra soportado de forma nativa en `Raspbian`. Sin embargo, puede emplearse el módulo del chip `DS1307` pues emplean la misma interfaz de comunicación. Para habitar el módulo se modifica el fichero de módulos del sistema (Código 5.22) y se añade el módulo `rtc-ds1307`.

Código 5.22
Habilitado del módulo de reloj físico en el sistema operativo
Fichero: /etc/modules

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-bcm2708
i2c-dev
rtc-ds1307
```

El siguiente paso consiste en inicializar el dispositivo reloj en el arranque del sistema, para ello es necesario ejecutar el Código 5.23 al inicio del sistema[36]. Esto creará un dispositivo en /dev/rtc0 que será el reloj conectado.

Código 5.23
Inicialización del reloj físico.

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new\_device
```

Para gestionar la conexión con el reloj, se puede emplear el servicio hwclock[37] que se encargará tanto de leer la hora de reloj en el inicio, como de actualizar ambas.

El módulo puede ser empleado conjunto al servidor NTP, con la ventaja de que el módulo de reloj seguirá funcionando en caso de errores de alimentación o reinicios del **nodo**.

Si bien todos los **nodos** incorporan por defecto todos los mecanismos software para la colocación de un módulo de reloj, en los escenarios estudiados en esta tesis, ninguno de ellos ha contado con el módulo de reloj al disponer todos ellos de conexión mediante red a un servidor NTP y garantizar una alta disponibilidad de red en todos ellos. El módulo de reloj, sin embargo, ha sido probado con éxito en las pruebas de laboratorio y en caso de monitorizaciones de **nodos** aislados de la red.

36 ↑Se recomienda, hacerlo por ejemplo en el fichero /etc/rc.local
37 ↑https://manpages.debian.org/testing/manpages-es-extra/hwclock.8.es.html

5.5.5 *Actualización de fichero OUI remoto*

Cómo se ha indicado en la Sección **??**, la asignación de la MAC de los dispositivos por medio de los fabricantes es regida por un organismo público. Debido a que constantemente nuevos dispositivos son lanzados al mercado, esta asignación es periódicamente actualizada. Como los `nodos` son los encargados de obtener el fabricante a partir de la dirección MAC, estos deben de disponer de la lista actualizada, con el fin de detectar los fabricantes de los nuevos dispositivos.

Para ello se dota al sistema de un script encargado de actualizar la asignación de MAC-fabricantes (Código 5.24).

Código 5.24
Script: Actualizador del fichero OUI

```
#!/usr/bin/perl
use strict;
use warnings;
use LWP::Simple;

my $ua = LWP::UserAgent->new();
$ua->show_progress('true value');
my $url = 'http://standards-oui.ieee.org/oui.txt';
my $file = "oui_download.txt";
open (OUT, '>manufacturers.properties');
getstore($url,$file);
open my $info, $file or die "No he podido abrir el fichero $file: $!";

while( my $line = <$info>)  {
        if ($line=~/^\s+([0-9a-fA-F]{6})\s+\(base 16\)\s+(.+)$/){
                print OUT $1;
                print OUT " = ";
                my $name = $2;
                $name = uc($name);
                $name =~ s/\.//g; #QUITAMOS LOS PUNTOS DE LAS CONTRACCIONES
                $name =~ s/  +//g; #QUITAMOS LOS ESPACIOS DE MÁS
                $name =~ s/CO.?//g; #QUITAMOS LA CONTRACIÓN DE LA COMPAÑÍA
                $name =~ s/INC.?//g; #QUITAMOS LA CONTRACIÓN DE LA INC
                $name =~ s/LTD.?//g; #QUITAMOS LA CONTRACIÓN DE LA LTD
                $name =~ s/A\/S//g; #QUITAMOS LA CONTRACIÓN DE LA AS
                $name =~ s/S\/A//g; #QUITAMOS LA CONTRACIÓN DE LA SA
                $name =~ s/,//g; #QUITAMOS LAS COMAS QUE NO DE FALLO CON EL CSV
                $name =~ s/ +$//g; #QUITAMOS LOS ESPACIOS DE MÁS
                $name =~ s/ P$//g; #QUITAMOS LA CONTRACIÓN DE LA COMPAÑÍA
                print OUT $name;
                print OUT "\n";
        }
}
```

La ejecución de este script puede ser programada mediante su inclusión en el cron, o ser ejecutado de forma remota mediante indicación del **servidor de cómputo** (Sección 5.6.5.10).

5.5.6 *Eficiencia del Sistema Operativo*

Finalmente, en esta sección se presentará el consumo de recursos del sistema operativo en ejecución, indicando principalmente el consumo de Memoria y CPU consumida únicamente por el sistema operativo.

Estudio 5.5.1: Uso de recursos del sistema Operativo HOREB

En esta sección se presenta brevemente el rendimiento del sistema operativo y su uso esperable de los recursos computaciones disponibles en la placa. Se presentan tanto los valores con sólo el sistema operativo, como ejecutando el software de monitorización que se presentará en la Sección 5.6.

Ejecutando solamente el sistema operativo se tiene un consumo real de memoria RAM de unos 35*MB* (Figura 5.49), más 13*MB* en buffers y 57*MB* pre-cacheados. En cuanto a la carga del sistema, en se sitúa entorno al 0.20. Se hace necesario notar que el procesador de `Raspberry Pi` dispone de 4 hilos de ejecución, por lo que la situación de sobrecarga (u `overloaded`) se produciría por valores de carga por encima de 4.00.

Código 5.25
Consumo de memoria RAM del sistema operativo

```
pi@raspberrypi ~ $ free -m -h
             total       used       free     shared    buffers     cached
Mem:          862M       106M       756M          0B        13M        57M
-/+ buffers/cache:         34M       827M
Swap:           0B         0B         0B

pi@raspberrypi ~ $ uptime
 20:37:52 up 36 min,  1 user,  load average: 0.20, 0.22, 0.13
```

Ejecutando el software de monitorización (Sección 5.6), se tiene un consumo real de memoria RAM de unos 107*MB* (Figura **??**), más 15*MB* en buffers y 76*MB* pre-cacheados. En cuanto a la carga del sistema, se sitúa en torno al 0.40. Sin embargo, estas mediciones son realizadas en el laboratorio, haciéndose necesario realizar pruebas de estrés del software.

Este experimento será completado en el Estudio 5.6.3, donde se medirá el impacto del software desarrollado.

5.6 SOFTWARE DE CAPTACIÓN COMUNICACIONES INALÁMBRICAS: RAZIEL

En esta sección se detalla el software encargado de la tarea de monitorización de los dispositivos inteligentes. Se presenta la arquitectura del sistema software propuesto para la monitorización, con tres componentes primordiales para cada tecnología inalámbrica empleada. A continuación se enumeran las herramientas y librerías empleadas para el desarrollo de dicho software. Finalmente, se presentan brevemente los componentes que se integran en el sistema denominado en el ámbito de esta tesis **RAZIEL**.

5.6.1 *Arquitectura propuesta*

Para cada tecnología inalámbrica a monitorizar, se proponen tres elementos software con competencias bien diferenciadas: el escáner, el monitor y el notificador (Figura 5.48).

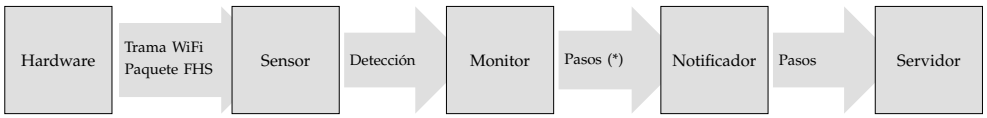


Figura 5.48
Secuencia de comunicación de los elementos del sistema. Desde el hardware se obtiene una Trama WiFi capturado o un paquete FHS de búsqueda Bluetooth, que es enviado y procesado por el Sensor. Este genera una detección, que es enviada al monitor. El monitor almacena la estancia del dispositivo durante un marco temporal. El Notificador, periódicamente, recibe los pasos de dispositivos que tienen que ser notificados al servidor, y realiza su envío o almacenamiento.

5.6.1.1 *Escáner*

El escáner (Scanner) es el módulo del sistema encargado directamente de comunicarse con las capas más afines al hardware para la obtención de tramas WiFi capturadas en modo monitor o de los paquetes FHS obtenidos en una búsqueda Bluetooth.

El escaneo debe ser realizado de forma continuada e ininterrumpida durante el funcionamiento del **nodo de monitorización**. Además, este componente es el encargado de iniciar las interfaces de red para la captura de tramas o búsqueda de dispositivos (Sección 5.5.2) así como la gestión y de recuperación de errores relativos a las interfaces de red.

Para cada detección, ha de ser capaz de extraer la dirección MAC que identifica al dispositivo detectado y aplicarle un cifrado unidireccional unívoco para garantizar la privacidad. Además, ha de marcar el instante de tiempo en el que la detección ha sido capturada por el hardware.

Por último, es deseable que se provean mecanismos que permitan delimitar los dispositivos detectados, ya sea mediante listas negras o criterios de

descarte (Como la intensidad `RSSI` de la trama WiFi, presentada en la Sección 5.1.2) con el fin de establecer un mecanismo de filtrado que permita acotar el ámbito y área de la monitorización.

5.6.1.2 *Monitor*

El monitor es el componente encargado de la monitorización de los dispositivos en las inmediaciones, nutriéndose de la información que le proporciona el escáner. Sus competencia principal es la de mantener el listado de dispositivos en las inmediaciones y el marco temporal de su estancia para cada uno de ellos. Además, ha de ser capaz de determinar cuando un dispositivo se ha alejado del nodo.

Para ello, el monitor recibe el evento de detección del dispositivo por parte del escáner. Este evento facilita la dirección MAC cifrada (**identificador**), el instante de tiempo donde ha sido detectado e información adicional dependiente de la tecnología inalámbrica[38] (Figura 5.50).

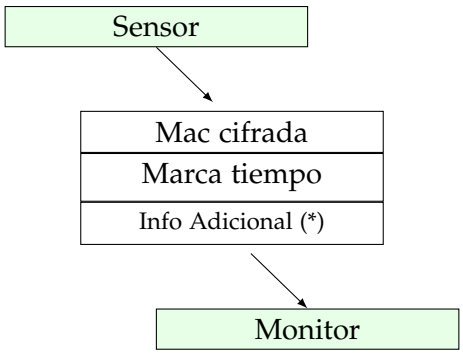


Figura 5.49
Información suministrada del sensor al monitor.
(*) La información adicional varía en función de la tecnología inalámbrica empleada y se encuentran descritas en la Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi.

El monitor mantiene un listado de dispositivos recientemente detectados (Figura 5.49). Este listado almacena para cada dispositivo su **identificador**, los instantes de tiempo de la primera y última vez que ha sido detectado y la información adicional que le haya proporcionado el sensor.

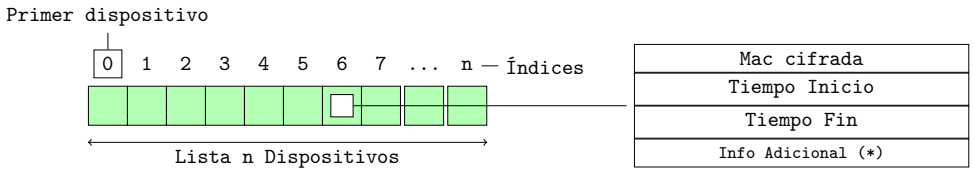


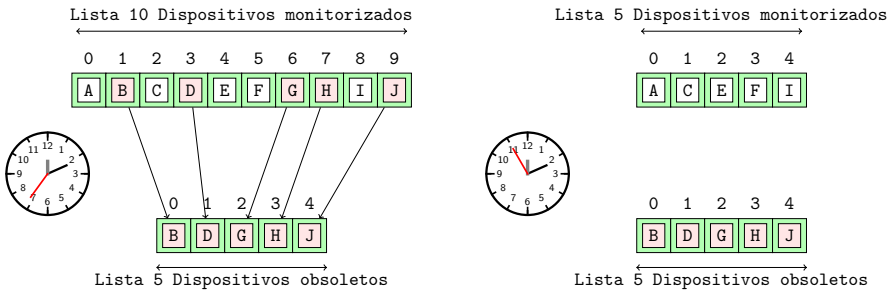
Figura 5.50
Información almacenada de cada dispositivo por el monitor.
(*) La información adicional varía en función de la tecnología inalámbrica empleada y se encuentran descritas en la Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi.

El monitor se encarga de actualizar la ventana de tiempo siguiendo el procedimiento descrito en la Sección **??**. Es decir, si es la primera vez que el

38 ↑Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi.

dispositivo es detectado, se almacena la información del paso marcando que la ventana de tiempo del estancia del dispositivo es igual a ambos instantes de tiempo. Si el dispositivo ya había sido detectado con anterioridad, se actualiza el instante de tiempo de última detección.

Además, periódicamente ha de decidir cuando un dispositivo del que hace tiempo que no se ha recibido ninguna detección ha abandonado la zona (Figura 5.51). Para ello, ha de iterar sobre la lista de dispositivos comprobando el tiempo que hace no se tiene una nueva detección sobre cada dispositivo. Este tiempo puede ser determinado de forma fija o establecido mediante una heurística en base a la localización del nodo, el día de la semana o la cantidad de dispositivos simultáneos localizados. Los dispositivos que se determinen que ya no están en las inmediaciones del **nodo** se denominan **obsoletos**.



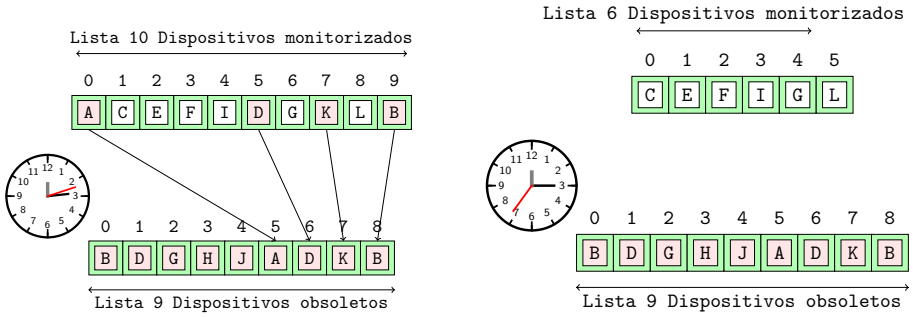
(a) Antes de la comprobación (b) Después de la comprobación

Figura 5.51
Comprobación periódica de dispositivos obsoletos por parte del monitor. El color rojo, indica que ese paso ha sido marcado como obsoleto por el monitor.
En (a) el monitor marca que los dispositivos `B`,`D`,`G`,`H` y `J` son obsoletos. Por lo tanto se pasan los pasos de estos dispositivos a un contenedor auxiliar de obsoletos.
En (b) el monitor mantendrá los pasos obsoletos en el contenedor hasta que sean requeridos.

Un mismo dispositivo puede ser detectado en múltiples ocasiones muy espaciadas en el tiempo, ocasionando tantos **pasos** como veces haya considerado el monitor que su visita ha quedado obsoleta (Figura 5.52) dado que el **paso** de un dispositivo consta también del marco temporal de la estancia.

Siempre que el monitor considere que la visita del dispositivo es obsoleta, lo sacará de la lista de monitorización y añadirá el **paso** a la lista de obsoletos, pudiendo estar múltiples veces el mismo dispositivo en esta lista con distintas ventanas temporales, las cuales se encuentran separadas, al menos, por tiempo a considerar obsoleto y sin superponerse una ventana sobre otra.

Esto es debido a que en la lista de dispositivos monitorizados, puede figurar una única vez un mismo dispositivo ya que si vuelve a ser detectado antes de marcar la estancia como obsoleta, se actualiza la marca temporal de última detección. Y esta marca temporal, es la que se emplea para considerar si la estancia del dispositivo se marca como obsoleta o no.



(a) Antes de la comprobación (b) Después de la comprobación

Figura 5.52
Comprobación periódica de dispositivos obsoletos por parte del monitor con repetición de un mismo dispositivo detectado y marcado obsoleto varias veces. El color rojo, indica que ese paso ha sido marcado como obsoleto por el monitor.
En (a) el monitor marca que los dispositivos `B`,`D`,`G`,`H` y `J` son obsoletos. Por lo tanto se pasan los pasos de estos dispositivos a un contenedor auxiliar de obsoletos.
En (b) el monitor mantendrá los pasos obsoletos en el contenedor hasta que estos datos sean requeridos. Si un dispositivo ya se encontraba en la lista de obsoletos, se añade un nuevo elemento al contenedor.

Por tanto, el monitor ha de ser capaz de procesar todas las detecciones que le provea el escáner, en un número escalable de estas, pues el número de detecciones es directamente proporcional al número de dispositivos en las inmediaciones. Además, ha de ser capaz de trabajar con la lista de dispositivos monitorizados y la lista dispositivos obsoletos de forma eficiente. Tiene que ser capaz de pasar de una lista a otra los dispositivos cuya estancia considere obsoleta.

5.6.1.3 *Notificador*

El notificador es el módulo del sistema encargado de obtener la información del monitor sobre los dispositivos monitorizados y almacenarla o enviarla al **servidor de cómputo** (Figura 5.93).

Figura 5.53
Información suministrada del monitor al notificador.
(*) La información adicional varía en función de la tecnología inalámbrica empleada y se encuentran descritas en la Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi.

Para ello solicita al **monitor** el listado de pasos que hay que comunicar al servidor o almacenar en la memoria. Se presentan dos modos de funcionamiento[39] en base a la respuesta esperada del monitor:

5.6.1.3.1 Modo Paso (STEP MODE)

En este modo el **monitor** sólo le comunica al **notificador** los pasos de aquellos dispositivos que han sido marcados como **obsoletos** (Figura 5.54(a)). Por ello, el **notificador** sólo dispone de la información de los dispositivos que se ha determinado que han abandonado las inmediaciones del **nodo**.

Este modo de funcionamiento es muy simple, pues solo propaga la información una vez que el dispositivo ha sido marcado como obsoleto, lo que implica que el **nodo** lleva mucho tiempo sin detectarlo.

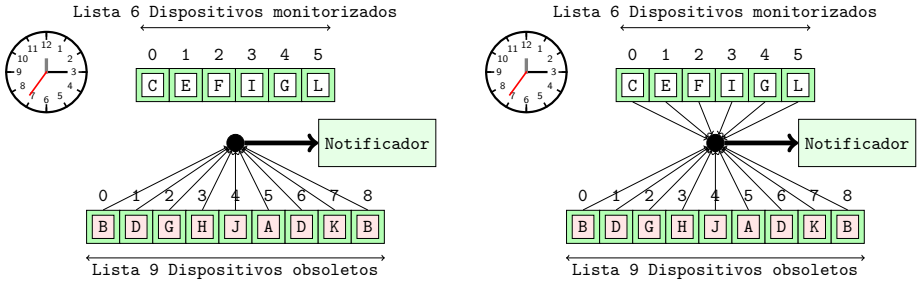
5.6.1.3.2 Modo Precoz (HASTY MODE)

En este modo el **monitor** comunica al **notificador** tanto los pasos de aquellos dispositivos que han sido declarados obsoletos como aquellos que actualmente se encuentran en estado de monitorización (Figura 5.54(b)), es decir, aquellos que aún se estima permanecen cerca del **nodo**.

Para economizar el tráfico de información, los dispositivos que actualmente se encuentran en estado de monitorización, solamente son notificados la primera vez que son detectados, estado que se controla mediante un simple booleano. Dado que no están marcados como obsoletos, tanto el **notificador** como las capas superiores, pueden asumir que el dispositivo aún se encuentra en las inmediaciones, aunque no disponga de la marca de tiempo final actualizada de su estancia.

Una vez que el **monitor** considera que el **paso** del dispositivo ha quedado obsoleto, este se actualizará al entrar en la lista de pasos obsoletos que son siempre entregados al **notificador**.

39 ↑Aunque en este prototipo, el segundo de ellos sólo ha sido implementando en WiFi por su mejor aducación para la detección de personas.



(a) Modo paso (b) Modo precoz

Figura 5.54
Modos de funcionamiento del monitor al enviar los pasos al notificador. El color rojo, indica que ese paso ha sido marcado como obsoleto por el monitor.
(a) En el modo paso, el monitor responde al notificador enviando únicamente aquellos pasos de dispositivos que han sido declarados obsoletos.
(b) En el modo precoz, el monitor responde al notificador enviado todos los pasos de los que dispone actualmente, incluyendo los pasos que pertenecen a dispositivos que están siendo monitorizados y no han sido declarados obsoletos.

En ambos casos, el **notificador** ha de realizar las comprobaciones necesarias para que la transmisión de la información sea la apropiada, así como informar al **monitor** de que **pasos** puede prescindir al haber almacenado o informado ya de ellos.

5.6.2 *Lenguaje, librerías y dependencias*

Para el desarrollo del **software de monitorización** se decide emplear JAVA, más concretamente la versión 7 de Oracle en la fase inicial del desarrollo, y posteriormente migrado a la versión 8.

Si bien el empleo de JAVA como lenguaje de programación puede ser controversial este no deja de ser uno de los lenguajes más empleados a nivel mundial [134], tanto en la investigación [46] o como el desarrollo orientado al Internet de las Cosas [317].

[134] Interactive: The Top Programming Languages 2018

[46] The 2017 Top Programming Languages

[317] 12 Popular Programming Languages for IOT Development

Su elección para el desarrollo del prototipo se debe principalmente a su independencia de la arquitectura. Esto permite que el **software de monitorización** sea independiente de la placa hardware (Sección 5.4) como del Sistema Operativo (Sección 5.5), manteniendo una única versión tanto del código como del ejecutable.

JAVA ofrece muchas facilidades para desarrollar software seguro, pues tanto el lenguaje, el compilador, el intérprete y el entorno de ejecución han sido desarrollados con la seguridad como objetivo[40] . De igual manera, ofrece multitud de mecanismos para garantizar la robustez y durabilidad del código ejecutado.

JAVA está diseñado para permitir de forma sencilla modelos de computación distribuida y multihebrada, lo cual resulta necesario para implementar el modelo de arquitectura propuesto en la Sección 5.6.1.

40 ↑http://oracle.com.edgesuite.net/timeline/java/

Además, debido a la popularidad del lenguaje, existen un amplio abanico de librerías, incluso para la gestión y manipulación tanto de tramas WiFi como de paquetes FHS (Ver AnexoA.3 para más información).

Todas estas ventajas suponen un contrapunto a los lenguajes alternativos que podrían haber ser empleados para el desarrollo del **software de monitorización**.

Por ejemplo, si bien `C` y `C++` son los lenguajes más empleados para el desarrollo de sistemas empotrados y con mayor rendimiento en la mayoría de benchmarks, construir la arquitectura propuesta requiere un esfuerzo mayor que en el caso de `JAVA`. Su principal baza es la mayor eficiencia en ejecución, sin embargo gracias al compilador `JIT` [3] de `JAVA` esta diferencia es cada vez menor [42, 181, 265] sobre todo en Software ejecutado a lo largo de mucho tiempo, no solamente en cortos periodos.

[3] Fast, effective code generation in a just-in-time Java compiler

[42, 181, 265] Benchmarking Java against C and Fortran for scientific applications, Ranking Programming Languages for Evolutionary Algorithm Operations, The Computer Language Benchmarks Game

Por otro lado, aunque `Python` es el lenguaje que más documentación y ejemplos tiene sobre `Raspberry Pi`, esto es principalmente a su uso académico, por lo que dichos ejemplos son excesivamente sencillos, cortos y empleados para prototipado muy rápido. La limitación por causa del `GIL`[41] en la ejecuciones multihilo y su lentitud relativa al ser un lenguaje interpretado, lastran su elección como lenguaje [42] para el desarrollo del prototipo.

Otros lenguajes como `Javascript` o `Go` se encuentran en estados de desarrollo demasiados tempranos como para ofrecer un entorno de ejecución los suficiente robusto y maduro para el **software de monitorización** del prototipo.

Si bien la elección de `JAVA` no descarta el posible empleo en el futuro de código de más bajo nivel de abstracción para la implementación de las zonas críticas del código, aplicable fácilmente gracias a la librería `JNI`[43]. Esta y otras librerías empleadas en el desarrollo se presentan en el Anexo A.3.

En el ámbito de esta tesis, se denominará **RAZIEL** al **software de monitorización** desarrollado.

41 ↑Globar interpeter lock

42 ↑`http://python-notes.curiousefficiency.org/en/latest/python3/multicore_python.html`

43 ↑`Java Native Interface`

5.6.3 *Entorno de configuración del software*

El **software de monitorización** se desarrolla para funcionar como un demonio del sistema. Sus ficheros principales se alojan en `/usr/share/raziel` (Figura 5.55).

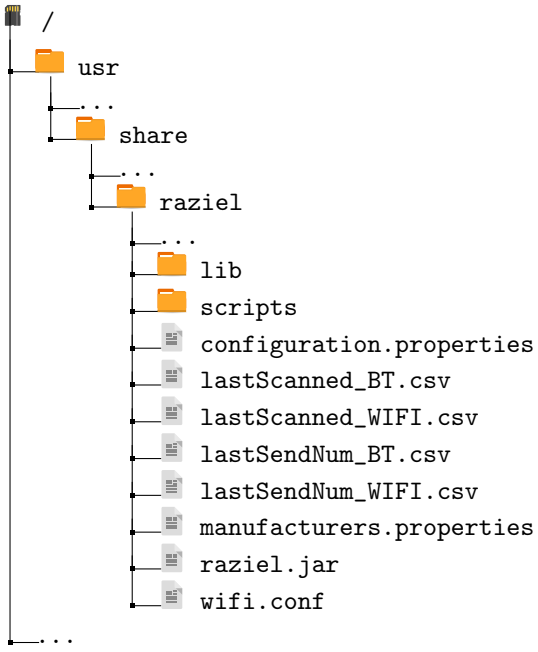


Figura 5.55
Ficheros principales empleados en la ejecución del Software RAZIEL

En dicho directorio se ubican el fichero de configuración (Figura 5.26), los contenedores temporales (Secciones 5.6.5.4.2 y 5.6.5.5.2), la relación de MAC-fabricantes (Sección 5.5.5), las librerías (Anexo A.3), scripts auxiliares y el ejecutable `jar` del software **RAZIEL**. Este directorio se corresponde con la partición de sólo lectura presentada en la Sección 5.5.1.4.

El fichero de configuración (Figura 5.26) contiene las variables de entorno (Tabla 5.9) que definen tanto el comportamiento del software de monitorización como identifican al **nodo** de forma unívoca.

Código 5.26
RAZIEL: Ejemplo de fichero de configuración, las variables entre los símbolos <> aparecen ocultas de forma deliberada en este documento.

```
bluetoothScan=1
wifiScan=1
wifiConnection=0
wifiMinDB= -80

#Modo de funcionamiento [step,hasty]
## step: Captura el "paso" del dispositivo, lo que implica la "entrada" y "salida"
↪  del dispositivo.
## hasty: El modo impaciente. Se notifica del dispositivo en cuanto es observado
↪  por primera vez.
WifiMonitorMode= hasty

#Tiempo "necesario" para considerar que el dispositivo "se ha ido para siempre"
tiempoObsoleto= 300000
tiempoCheckObsoleto= 30000

idNodo=90
idNodoBTService=91
idNodoWFService=92
nombreNodo= "Debug"
latitud=0.0
longitud=0.0
intervalo=300000
idUsuarioWs=<usuario>
passWs=<password>
urlWs=<url>
```

Tabla 5.9
Variables de configuración e identificación del software de monitorizacion RAZIEL.

VARIABLE	TIPO	DESCRIPCIÓN
bluetoothScan	(Booleano)	Bandera que indica si el nodo ha de detectar dispositivos Bluetooth.
wifiScan	(Booleano)	Bandera que indica si el nodo ha de detectar dispositivos WiFi.
wifiConnection	(Booleano)	Bandera que indica si el nodo ha de conectarse a una red WiFi
wifiMinDB	(Integer)	Umbral de la fuerza de la señal para descartar tramas WiFi.
WifiMonitorMode	(step|hasty)	Modo de monitorización de dispositivos WiFi. Ver Sección 5.6.1.3.
tiempoObsoleto	(Integer)	Establece el tiempo requerido sin detección de un dispositivo para considerar el paso asociado a él obsoleto. Ver Sección 5.6.1.2.
tiempoCheckObsoleto	(Integer)	Establece el tiempo que marca el periodo cada cuanto comprobará el monitor si sus pasos actuales se han quedado obsoletos.
idNodo	(Integer)	Identificador en el sistema del nodo de monitorización.
idNodoBTService	(Integer)	Identificador del sensor Bluetooth del nodo de monitorización.
idNodoWFService	(Integer)	Identificador del sensor WiFi del nodo de monitorización.
nombreNodo	(String)	Nombre del nodo para una identificación más sencilla.
latitud	(Float)	Latitud en la que se encuentra emplazada el nodo de monitorización.
longitud	(Float)	Longitud en la que se encuentra emplazada el nodo de monitorización
intervalo	(Integer)	Intervalo de actuación del notificador.
idUsuarioWs	(String)	Usuario de autenticación del nodo. Ver Sección 5.8.
passWs	(String)	Credenciales de autenticación del nodo. Ver Sección 5.8.
urlWs	(String)	Dirección del servidor que atiende las peticiones. Ver Sección 5.8.

5.6.3.1 *Identificación de los nodos y sensores*

De entre estas variables, merece una mención especial el identificador de nodo y sensores. Estos se forman siguiendo un código numérico, de tal forma que el último dígito (las unidades) representa el sensor del nodo, con un 1 indicando que es el sensor Bluetooth del nodo, y con un 2 indicando que es el sensor WiFi. Los siguientes dígitos (decenas y centenas) indican el número de nodo. Los siguientes dígitos, son empleados para agrupar los nodos. En el ámbito de esta tesis, se han empleado este identificador de grupo para distinguir entre los distintos proyectos en los que el sistema ha sido empleado. Por ejemplo, el identificador 1062 (Figura 5.56) se refiere al sensor wifi del nodo 1060. El sensor 0 por tanto, es empleado cuando se desea referir todo el nodo, no únicamente a alguno de sus sensores.

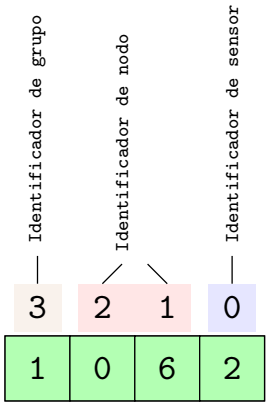


Figura 5.56
Ejemplo práctico de la composición del identificador de nodo y sensor.
En el ejemplo, se presenta el identificador 1062. El 2 indica que se refiere al sensor WiFi del nodo 106. Además, el 1 indica que se refiere al grupo 1 de nodos.

Esta nomenclatura de identificación puede resultar confusa, pero permite de manera sencilla trabajar con los pasos independientemente de su naturaleza (Bluetooth o WiFi), o cualquier otro sensor que se desease implantar en el futuro en el nodo independientemente de su naturaleza. Además, permite tener un control de los nodos, mantiéndolos agrupados por los proyectos que los han instalado. Sin embargo esta nomenclatura puede ser variada en cualquier momento a voluntad, porque el archivo de configuración permite establecer cualquier identificador tanto para el nodo como para cualquiera de sus sensores.

5.6.3.2 *Definición de la ubicación de los nodos*

Los nodos de monitorización son geoposicionados mediante el fichero de configuración de los mismos, estableciendo su coordenadas (latitud y longitud). Estas coordenadas son asociadas a cada paso de dispositivo realizado que el notificador envía hasta el servidor de cómputo.

Esto obedece a una decisión de diseño, de forma que un mismo nodo identificado por un identificador, puede ser movido sin que los pasos detectados anteriormente se vean afectados por ese movimiento.

Además, en un futuro las coordenadas de posicionamiento del nodo pueden ser establecidas de forma dinámica, por ejemplo mediante la incorporación de un módulo GPS, de forma que el nodo esté también en movimiento y mandado pasos de distintas ubicaciones.

Por lo tanto, estas coordenadas son enviadas desde el nodo al servidor anexadas a cada paso realizado por el notificador.

5.6.4 *Clases base empleadas en el software*

En la arquitectura (Sección 5.6.1) se han presentado tres elementos que realizan comunicaciones entre ellos. En esta sección, se presentan brevemente las clases que codifican los mensajes intercambiados entre esos elementos. Dichas clases serán necesarias para comprender el funcionamiento del sistema atendiendo a la información que manejan en todo momento sobre los dispositivos monitorizados.

Las clases `BTData` (Tabla 5.10) y `WiFiData` (Tabla 5.11) codifican la información relativa a la detección de un dispositivo (5.1.2) mediante comunicaciones Bluetooth y WiFi respectivamente. Esta información es intercambiada entre el scanner y el monitor (Sección 5.6.1.2).

Tabla 5.10
Clase `BTData` que codifica la información de la detección de un dispositivo Bluetooth.

`BTData`		
`_time`	(`Long`)	Marca de tiempo que determina el instante en el que el dispositivo ha sido detectado. Precisión de milisegundo.
`_mac`	(`String`)	Dirección MAC (Sección 4.2.2) del dispositivo.
`_name`	(`String`)	Nombre que identifica el dispositivo. Aunque esta información es conocida en el instante de la detección, se considera información sensible por lo que no es propagada a los pasos.
`_majorDevClass`	(`String`)	Identificador tipo principal de dispositivo. Ver Sección 4.2.3.5.
`_minorDevClass`	(`String`)	Identificador suptipo de dispositivo. Ver Sección 4.2.3.5.
`_serviceDevClass`	(`String`)	Identificado los servicios ofrecidos. Ver Sección 4.2.3.5.

La clase `StepData` (Tabla 5.12, codifica la información relativa a los pasos de dispositivos (Sección 5.1.3). Al contrario que en la detección, se ha optado porque la información del paso de un dispositivo sea común a los dispositivos detectados por una u otra tecnología inalámbrica. Esto permite trabajar con

Tabla 5.11
Clase `WiFiData` que codifica la información de la detección de un dispositivo WiFi.

`WiFiData`		
`_time`	(`Long`)	Marca de tiempo que determina el instante en el que el dispositivo ha sido detectado. Precisión de milisegundo.
`_mac`	(`String`)	Dirección MAC (Sección 4.3.5) del dispositivo.
`_signal`	(`int`)	El indicador de fuerza de la señal. Ver Sección 5.1.2.

el concepto de pasos con una capa de abstracción a la tecnología por la que se detectó al dispositivo. Esta información es la almacenada por el monitor (Sección 5.6.1.2) y la intercambiada con el notificador (Sección 5.6.1.3).

Tabla 5.12
Clase `StepData` que codifica la información del paso de un dispositivo.

`StepData`		
`_obsolete`	(`boolean`)	Bandera que indica si el paso ha quedado obsoleto. Ver Sección 5.6.1.3.
`_new`	(`boolean`)	Bandera que indica si el paso es nuevo. Se emplea en el modo hasty. Ver Secciones 5.6.1.3.2 y 5.6.5.5.3.
`_idDevice`	(`String`)	Identificador del dispositivo. Se obtiene cifrando la dirección MAC mediante SHA1.
`_manufacturer`	(`String`)	Nombre del fabricante del dispositivo. Ver Sección 5.1.2.
`_idSensor`	(`Long`)	Identificador del sensor que ha realizado la detección. Ver Sección 5.6.5.3.
`_latitude`	(`Float`)	Latitud del nodo en el momento de realizar la detección. Ver Sección 5.6.5.3.
`_longitude`	(`Float`)	Longitud del nodo en el momento de realizar la detección. Ver Sección 5.6.5.3.
`_tinitial`	(`Long`)	Marca de tiempo del instante inicial en el que ha sido detectado el dispositivo.
`_tfinal`	(`Long`)	Marca de tiempo del último instante de tiempo en el que ha sido detectado el dispositivo.
`_sinitial`	(`int`)	El indicador de fuerza de la señal de la primera detección del dispositivo. Ver Sección 5.1.2.
`_sfinal`	(`int`)	El indicador de fuerza de la señal de la última detección del dispositivo. Ver Sección 5.1.2.
`_majorDevClass`	(`String`)	Identificador tipo principal de dispositivo. Ver Sección 4.2.3.5.
`_minorDevClass`	(`String`)	Identificador suptipo de dispositivo. Ver Sección 4.2.3.5.
`_serviceDevClass`	(`String`)	Identificado los servicios ofrecidos. Ver Sección 4.2.3.5.

Ya que el `StepData` contiene información de ambos tipos de tecnologías, hay campos que solamente son empleados cuando el dispositivo empleado ha sido detectado con dicha tecnología. En ese caso, se mantiene a `null` dichos campos. El notificador sabe omite esta información a la hora de realizar el envío al **servidor de cómputo**.

5.6.5 *Componentes del sistema*

El software **RAZIEL** está compuesto de 11 módulos principales, muchos de ellos ejecutados como hebras independientes del proceso principal. **RAZIEL** se comporta como un demonio dentro del sistema operativo **HOREB**, lo que le permite operar con los recursos del sistema con privilegios administrativos, necesarios principalmente para la captura de tramas las wifi.

La modularidad del sistema le permite ser configurado en función de las necesidades a cubrir, así como la fácil modificación y expansión del

funcionamiento del software. Además, permite facilitar la detección de errores en las fases de desarrollo.

A continuación se enumeran los distintos módulos en los que se descompone el sistema.

MAIN la hebra principal del sistema RAZIEL, encargada de instanciar el resto de módulos de sistema así como de la comprobación de errores de primer nivel (Página 204).

CONFIGURATIONMODULE encargado de proveer las variables de configuración y entorno del software de monitorización (Página 206).

CAPTUREBTMODULE encargado de la captura, procesamiento y envío de los pasos capturados mediante BT (Página 208).

SENSOR (Página 209)

MONITOR (Página 210)

NOTIFICADOR (Página 212)

CAPTUREWIFIMODULE encargado de la captura, procesamiento y envío de los pasos capturados mediante WiFi (Página 213).

SENSOR (Página 215)

MONITOR (Página 217)

NOTIFICADOR (Página 221)

3GCONNECTIONMODULE encargado de la gestión del establecimiento de la conexión 3G para comunicaciones externas (Página 226).

WIFICONNECTIONMODULE encargado de la gestión del establecimiento de la conexión WiFi para comunicaciones externas (Página 225).

SYSTEMUPDATEMODULE encargado de la autogestión de las versiones del software RAZIEL (Página 228).

SCRIPTSMODULE encargado de la ejecución de scripts en remoto en el sistema HOREB (Página 230).

EMAILMODULE encargado de la emisión de mensajes de notificación y error del sistema mediante correo electrónico (Página 233).

ABDIELSERVERMODULE encargado del establecimiento y gestión de la conexión con el servidor central de procesamiento ABDIEL (Página 223).

EVERYTHINGITSFINEMODULE encargada de la gestión y comprobación y recuperación ante errores del sistema RAZIEL y sus módulos (Página 234).

A parte de los 11 módulos principales, se implementan dos módulos auxiliares que proveen de mecanismos adicionales comunes, pero que no realizan una funcionalidad del sistema. Se trata de los módulos de depuración o LOG (Página 235) y utilidades o `util` (Página 235).

5.6.5.1 *Demonio*

El demonio se instancia mediante el script `/etc/init.d/raziel` añadido a HOREB. Este script se ejecuta automáticamente en el arranque del sistema una vez que el resto de script de `/etc/init.d/` han sido ejecutados. Dicho script dota al demonio de los siguientes servicios:

`START` servicio que inicia el software de monitorización.

`STOP` servicio que detiene el software de monitorización.

`FORCE-RELOAD Y RESTART` servicio que reinicia el software.

`UPDATE` servicio empleado para instalar una actualización del de monitorización **RAZIEL**. Más detalles en la Sección 5.6.5.9.

`UNDO-UPDATE` servicio empleado para revertir una actualización del software a la última versión estable empleada. Más detalles en la Sección 5.6.5.9.

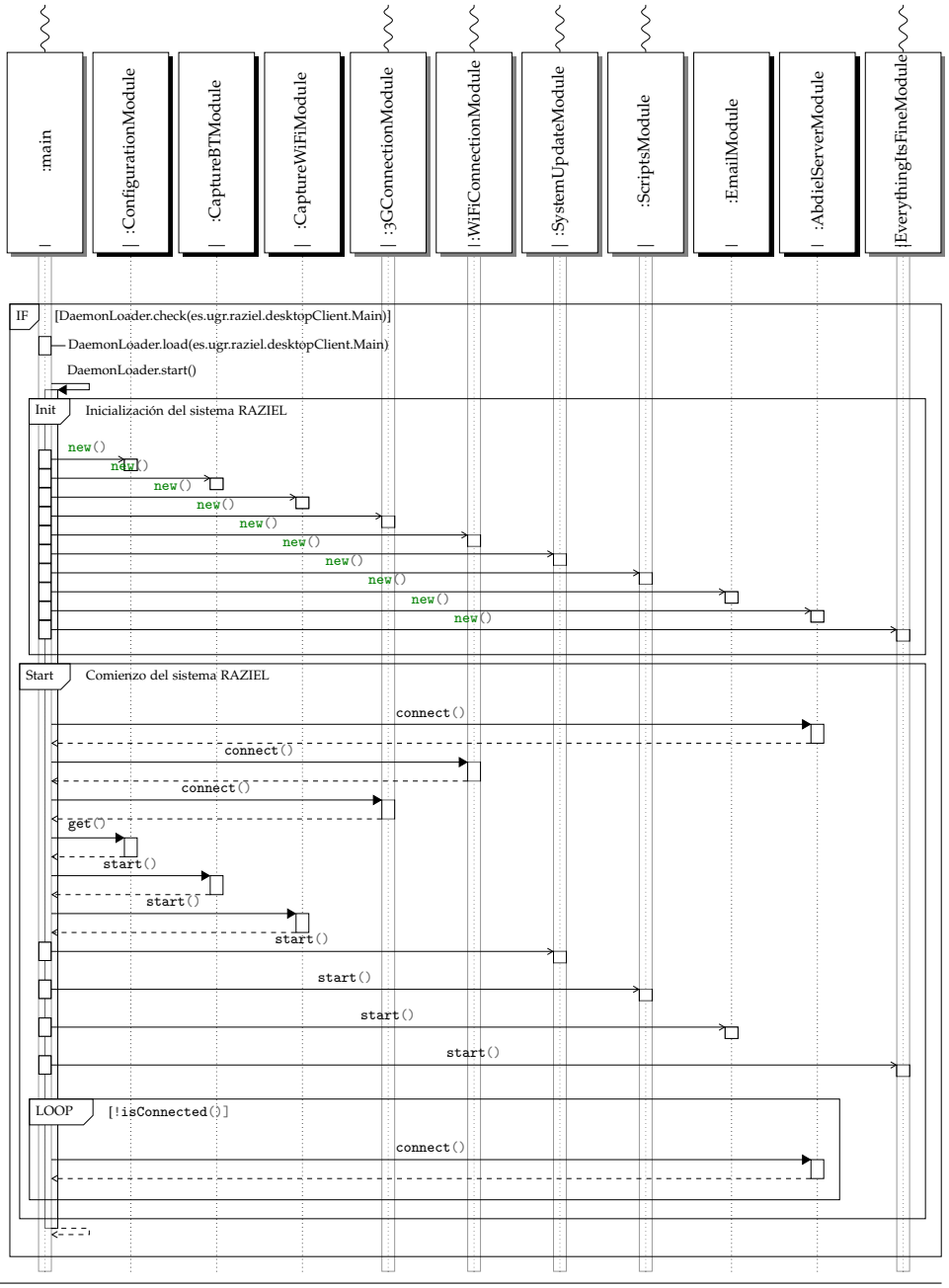
`STATUS` Servicio que proporciona detalles sobre la ejecución del software.

5.6.5.2 *Main*

Una vez el demonio comienza la ejecución del software de monitorización **RAZIEL**, se declara el demonio de nombre completo `es.ugr.raziel.desktopClient.Main` y se comienza la instancia de los módulos del sistema (Código 5.27).

La declaración de dicho demonio permite garantizar que sólamente una instancia de **RAZIEL** se encuentra en ejecución.

Código 5.27
RAZIEL: Módulo MAIN encargado de la iniciación del resto de módulos del sistema



En primer lugar se carga la configuración, que entre otros, contiene los módulos que se han definido serán necesarios para esta ejecución. El sistema ofrece versatilidad en cuanto a los módulos que carga, sin embargo existen dependencias funcionales entre muchos de ellos, siendo por ejemplo el módulo `ConfigurationModule` imprescindible, así como al menos uno de los módulos de captura (`CaptureBTModule` o `CaptureWiFiModule`).

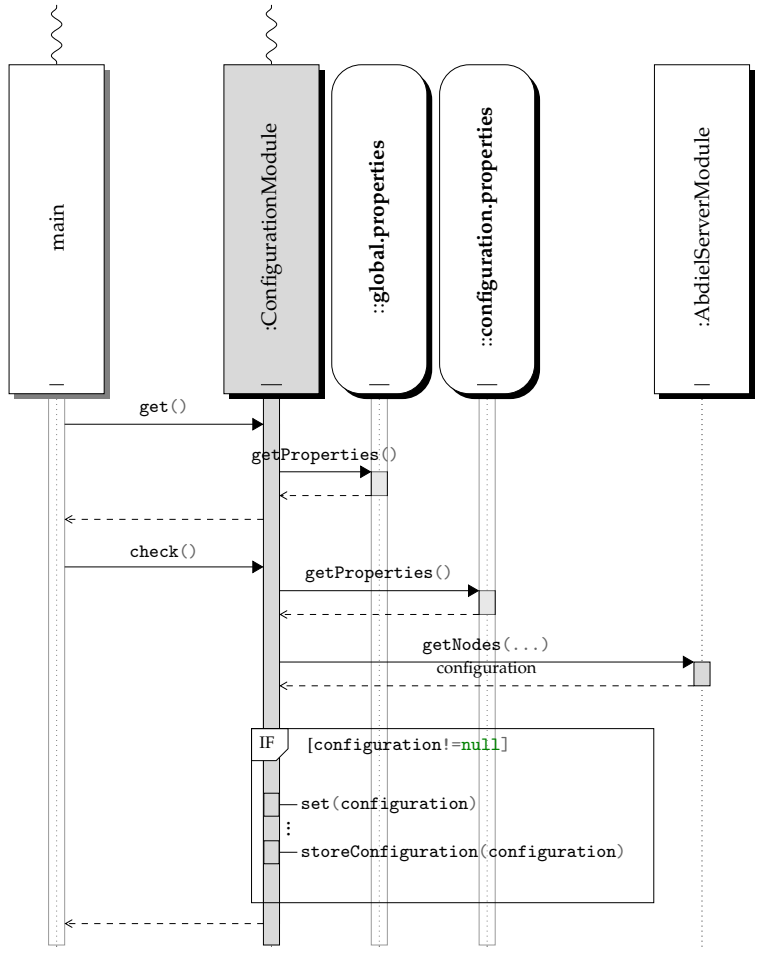
Una vez el módulo `MAIN` ha inicializado el resto de módulos, queda a la espera de alguna invocación de los servicios del demonio definidos en la Sección 5.6.5.1.

5.6.5.3 *Configuration Module*

Este módulo provee de las variables configurables y de entorno que permiten modificar el comportamiento del sistema **RAZIEL**. También es el encargado, frente al sistema operativo, de las tareas de configuración de las nuevas versiones del software y configuraciones directas del sistema operativo (para más detalles ver las secciones 5.6.5.9 y 5.6.5.10).

El sistema provee de tres entornos desde puede obtenerse las variables de configuración, interactuando entre ellas según se refleja en el Código 5.28. De esta forma, la información más prioritaria es la que propaga el servidor y la menos prioritaria es la que se aloja en la configuración interna del software empaquetado.

Código 5.28
RAZIEL: Módulo de configuración encargado de cargar las variables de entorno y configuración que definen el comportamiento variable del sistema.



5.6.5.3.1 Entorno `global.properties`

El contenedor `global.properties` es un fichero empaquetado e integrado en el software. Entre otras variables, contiene el número de versión del software, los valores por defecto de conexión externa y los tiempos de escaneo. La única manera de modificar las variables contenidas en este contenedor es mediante la modificación del ejecutable del software. Este contenedor, adicionalmente, posee tres campos adicionales que no pueden ser sobrescritos por las otras fuentes de comunicación.

El primero de esos campos indica el número de versión del software empaquetado. Se emplea en el sistema de actualización (Sección 5.6.5.9).

Los otros dos campos, son empleados para la securización del acceso al servidor, aportando una capa de adicional al canal de comunicaciones empleando autenticación por HTTP, además del propio mecanismo de autenticación del servidor. El empleo de estas variables es descrito en la Sección 5.8.2 .

5.6.5.3.2 Entorno `configuration.properties`

El contenedor `configuration.properties` es un fichero alojado en `/usr/share/raziel/` que contiene las variables de identificación del nodo (idSensor, idNodoBluetooth), información sobre el nodo (nombreNodo, latitud y longitud) las credenciales de acceso al servidor (url, usuario, credenciales), los módulos requeridos para la ejecución del sistema, los tiempos y modos de funcionamientos implicados directamente en la captación y monitorización del sistema. Estos parámetros han sido descritos al detalle en la Sección 5.6.3.

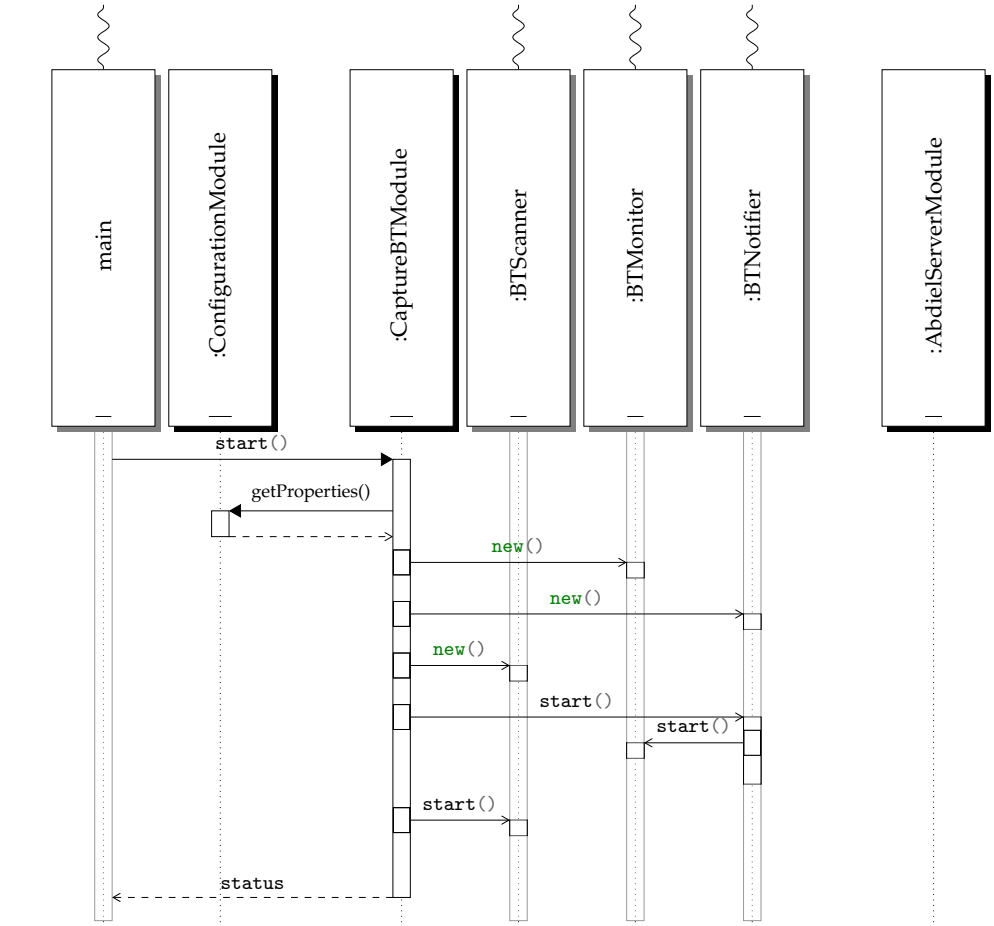
5.6.5.3.3 Entorno `AbdielServerModule`

Por último, el módulo le solicita al módulo `AbdielServerModule` que le facilite la configuración que el servidor tiene establecida para dicho nodo identificado. Tras obtener las variables de configuración, el sistema no sólo emplea esas variables en la ejecución, sino que también actualiza el fichero `configuration.properties` almacenando la nueva configuración establecida.

5.6.5.4 *Capture Bluetooth Module*

El módulo de captura Bluetooth se encarga, como indica su nombre, de la monitorización de dispositivos detectado mediante la captación de comunicaciones Bluetooth. Este módulo incorpora los tres componentes (Código 5.29) definidos en la arquitectura (Código 5.29): un escáner (`BTScanner`), un monitor (`BTMonitor`) y un notificador (`BTNotifier`).

Código 5.29
RAZIEL: Módulo de captura de dispositivos Bluetooth, que presenta los tres componentes descritos en la Arquitectura: escáner, monitor y notificador.

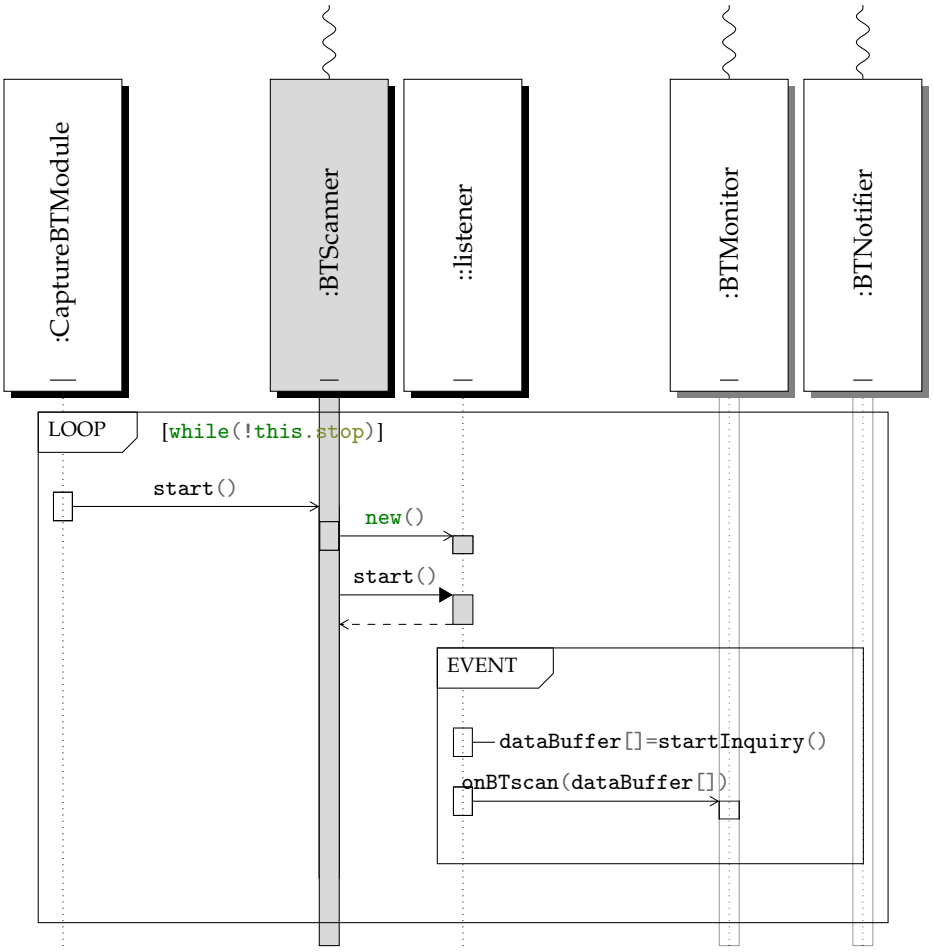


Cada uno de estos submódulos se comporta según lo descrito en la Sección 5.6.1, aunque con algunas peculiaridades propias de la implementación, que se describen a continuación brevemente.

5.6.5.4.1 Scanner

El escáner Bluetooth (`BTScanner`) está constantemente realizando una `Inquiry` tras otras siguiendo los mecanismos que se han descrito en la Sección 4.2 (Código 5.30).

Código 5.30
RAZIEL: Módulo de captura de dispositivos Bluetooth: Escaner
El escáner se encuentra constantemente realizando búsqueda de dispositivos Bluetooth en las inmediaciones. Los resultados son enviados al Monitor.

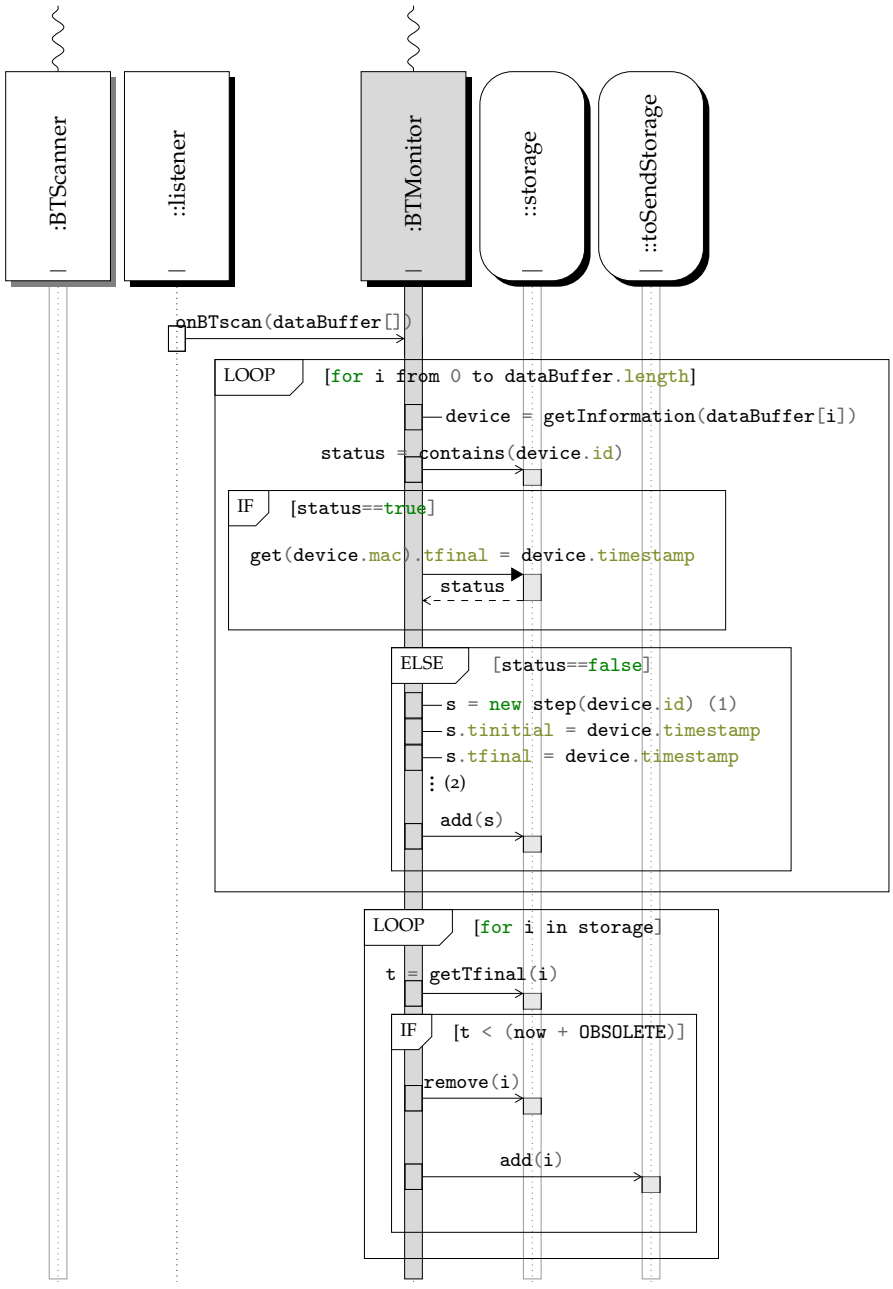


Con los resultados de cada búsqueda `BTScanner` envía a `BTMonitor` los resultados de la `Inquiry`. Estos resultados puede pertenecer tanto a un único dispositivo, como a varios. Una vez enviados los resultados, sin necesidad de obtener respuesta por parte de `BTMonitor`, empieza una nueva búsqueda.

5.6.5.4.2 Monitor

El monitor es el encargado de procesar la información enviada por el `BTScanner` para llevar el control de la monitorización de los dispositivos (Código 5.31).

Código 5.31
RAZIEL: Módulo de captura de dispositivos Bluetooth: Monitor.
El monitor recibe la información proporcionada por el escáner, que empleará para actualizar la información que mantiene en sus dos contenedores.



(1) Realiza el cifrado de la MAC y la identificación del fabricante.

(2) Añade al paso la información sobre el nodo, su identificador y coordenadas.

El monitor mantiene dos almacenamientos como se ha presentado en la Sección 5.6.1.2. Uno de ellos contiene los dispositivos que están siendo actualmente monitorizados (`storage`) y otro mantiene los dispositivos que han sido marcados como obsoletos (`toSendStorage`). Para Bluetooth, sólo se ha implementado el modo de Monitorización por Pasos, descrito en la Sección 5.6.1.3.1, por lo que al contenedor `::toSendStorage` sólo van los dispositivos que se hayan considerado obsoletos.

El contenedor `storage` se implementa en una estructura hash mantenida en memoria, empleando el identificador como clave de la estructura y el resto de información presentada en la Sección 5.6.1.2 como valor.

Para el contenedor `toSendStorage` se opta por mantener un fichero `CSV` alojado en la partición de `/usr/share/raziel/` denominado `lastScanned_BT.csv`[44]. Debido a la naturaleza del contenedor (`::toSendStorage`), este únicamente requiere escribir los nuevos **pasos** que se vayan marcando como obsoletos. Y realizar una lectura total cuando vayan a ser mandados los **pasos** al servidor. Si bien mantener el fichero en la partición puede parecer que atenta contra la vida de la tarjeta microSD como se presentó en la Sección 5.5.1, es la mejor alternativa para mantener en todo momento los **pasos** ya detectados, sin que estos sean perdidos ante fallas del software o de la integridad del nodo antes de que hayan sido transmitidos.

Una vez que el monitor (`BTMonitor`) recibe la información que ha generado el `BTScanner`, interpreta la información de cada uno de ellos. Si la información interpretada es válida, se consulta al contenedor `storage` si el dispositivo identificado se encuentra dentro, lo que implica que el dispositivo ya había sido detectado con anterioridad en cuyo caso se actualiza el instante de tiempo `tfinal` con el instante de tiempo en el que ha sido detectado el dispositivo `device.timestamp`. En caso de que el dispositivo no haya sido detectado con anterioridad, se crea un nuevo paso (`s`) empleando el identificador del dispositivo, y se actualizan su ventana de detección (`tinitial` y `tfinal`) al instante de tiempo en el que fue detectado (`device.timestamp`) y se introduce en el contenedor `storage`.

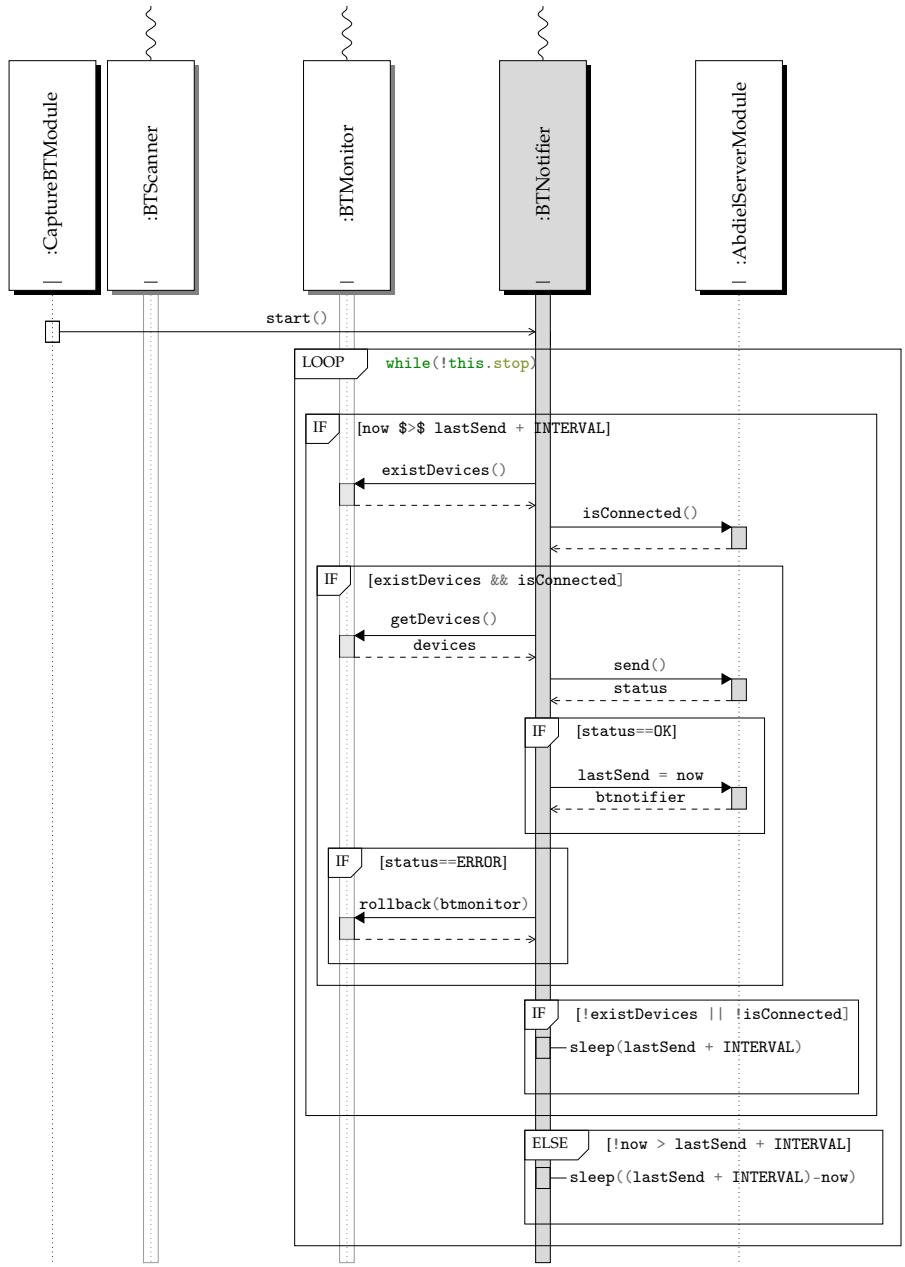
Esta acción se realiza secuencialmente para todos los dispositivos que hayan sido proporcionados por el `BTScanner`. Una vez que se ha procesado el buffer de datos, se realiza la comprobación de obsoletos. Para ello, el `BTMonitor` itera todos los **pasos** comprobando si el instante de tiempo de la última detección (`tfinal`) es menor que la fecha actual más el instante de tiempo definido en la configuración pasa ser configurado obsoleto. Cuando el paso del dispositivo se determina que es obsoleto, se elimina del contenedor en memoria `storage` y es copiado en el contenedor `toSendStorage`, lo que repercute en una escritura en el fichero `CSV` `/usr/share/raziel/lastScanned\_BT.csv`.

44 †Aunque incorpora Scanned en el nombre, no ha de confundirse, ya que este fichero contiene únicamente los **pasos** que han sido determinado por el monitor como obsoletos.

5.6.5.4.3 Notifier

El notificador (`BTNotifier`) solamente entra en juego con el **servidor de cómputo** se encuentra presente, y su cometido es el de enviar los pasos que el `BTMonitor` ha considerado como obsoletos (Código 5.32). Para ello mantiene una marca de tiempo de en que momento realizó él último envío con éxito al servidor.

Código 5.32
RAZIEL: Módulo de captura de dispositivos Bluetooth: Notificador
El notificador es el encargado de periódicamente notificar los pasos que el monitor ha considerado obsoletos y enviarlos al servidor de cómputo.



El `BTNotifier` de forma periódica le pregunta al `BTMonitor` si tiene dispositivos marcados como obsoletos listos para el envío y pregunta al módulo

del servidor `AbdielServerModule` (que será descrito en la Sección 5.6.5.6) si existe conexión con el **servidor de cómputo**. En caso afirmativo, le solicita al `BTMonitor` que le envíe esos pasos. En ese instante, el `BTMonitor` vuelca el contenido del contenedor `toSendStorage` dejándolo vacío.

`BTNotifier` procede al envío mediante `AbdielServerModule` los pasos a enviar. En caso de que no exista error por parte del `AbdielServerModule`, se actualiza la fecha del último envío exitoso. En caso contrario, hay que proporcionales nuevamente los pasos que no han sido enviados al `BTMonitor` para que los ponga a buen recaudo nuevamente.

Finalmente, y en cualquier caso, el `BTNotifier` se duerme hasta que le toque el próximo intento de envío con el servidor.

En caso de desconexión continuada con el **servidor de cómputo**, el módulo `AbdielServerModule` no devolverá nunca el estado de conexión correcto, por lo que no se le solicitarán al `BTMonitor` en ningún momento los pasos almacenados en `toSendStorage`. De esta forma, los pasos recogidos a lo largo de todo el tiempo se encontrarán seguros en la tarjeta microSD hasta la recogida del **nodo** o se restablezca la conexión.

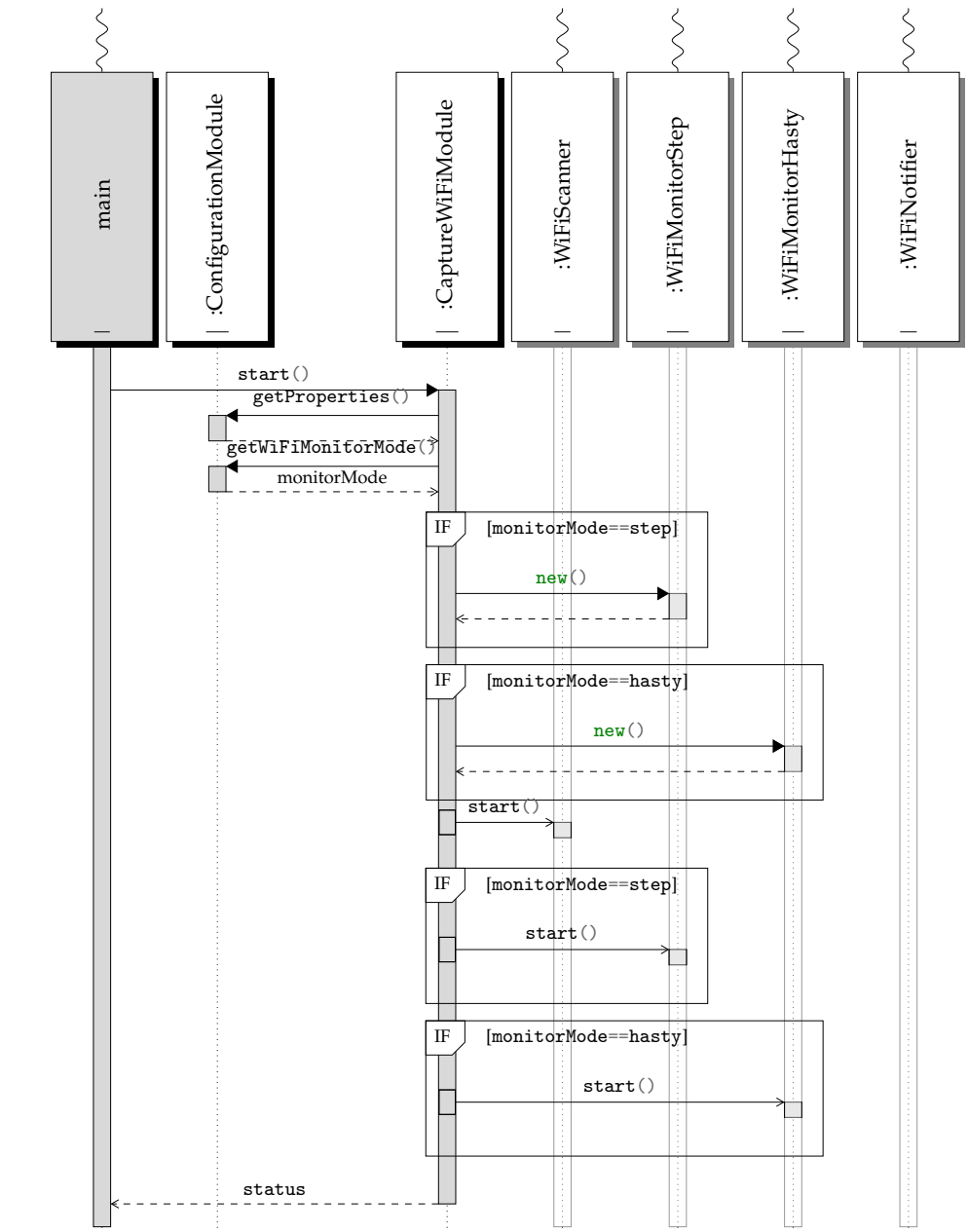
5.6.5.5 *Capture WiFi Module*

El módulo de captura WiFi se encarga, como indica su nombre, de la monitorización de dispositivos detectado mediante la captación de comunicaciones WiFi.

Este módulo incorpora los tres componentes definidos en la arquitectura (Código 5.33) un escáner (`WiFiScanner`), un monitor (`WiFiMonitor`) y un notificador (`WiFiNotifier`).

Para este módulo se han empleado los dos tipos de monitores implementados: modo paso (Sección 5.6.1.3.1) y modo precoz (Sección 5.6.1.3.2). Ambos modos serán descritos primero sus componentes comunes, y posteriormente su funcionalidad concreta.

Código 5.33
RAZIEL: Módulo de captura de dispositivos WiFi, que presenta los tres componentes descritos en la Arquitectura: escáner, monitor y notificador.



Cada uno de estos submódulos se comporta según lo descrito en la Sección 5.48, aunque con algunas peculiaridades propias de la implementación. Para el módulo de captura WiFI se han implementado los dos modos de monitorización: el modo de monitorización de pasos presentado en la Sección 5.6.1.3.1 y el modo precoz presentado en la Sección 5.6.1.3.2. La elección entre un modo u otro se establece según una variable de configuración (Código 5.26).

5.6.5.5.1 Scanner

El escáner WiFI (`WiFiScanner`) es más complicado que el `BTScanner` porque, como se presentó en la Sección 4.3, el protocolo WiFi no permite de forma nativa la búsqueda de dispositivos cercanos. En su lugar, se capturan tramas que los dispositivos están emitiendo, gracias a la interfaz de red configurada en modo monitor (Sección 4.3.4).

En primer lugar se activa dicho modo monitor mediante los scripts de `aircrack` (Sección 73). Esto crea una interfaz de red `mon0` en el sistema operativo. Entre las tareas del `WiFiScanner` está de gestionar que esta interfaz se encuentre siempre habilitada, pues la que se encarga de la recepción de paquetes en el modo monitor.

Un `listener` es el encargado de recibir la información RAW de cada trama capturada por la interfaz `mon0`. Al contrario que en `BTScanner` que se sigue siempre el formato `FHS`, cada trama WiFi capturada tiene un formato distinto, que el `WiFiScanner` ha de saber interpretar, ya que la información recibida es binaria. Para facilitar la interpretación, se convierte esta información binaria en una cadena de texto hexadecimal (Figura 5.57).

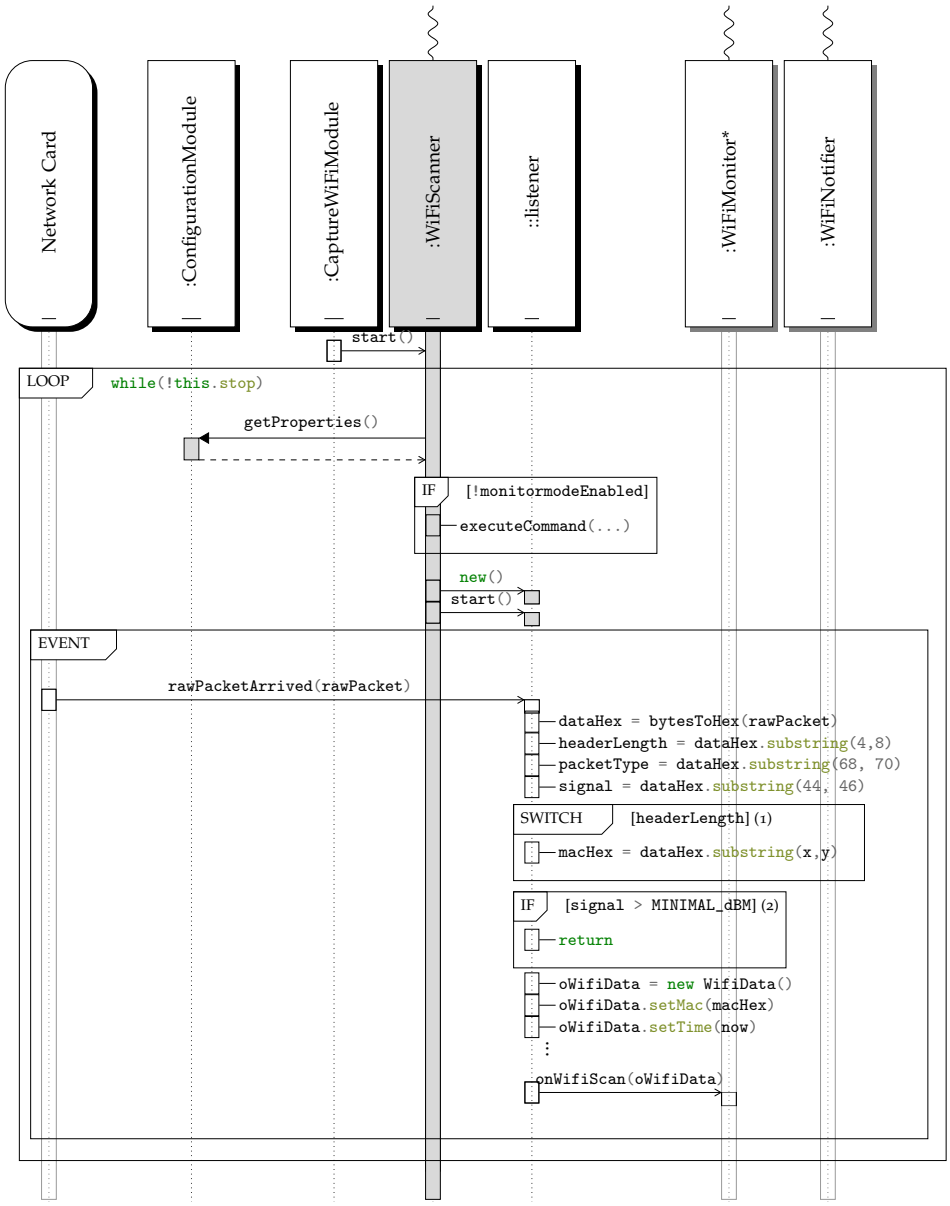
```
15-02-2016 13:48:01,992 TRACE [es.ugr.raziel.threads.scanners.WifiScannerThread]
Encontrado 1 paso wifi a -76db de 4C7403E3B418
000024002F4000A02008000000000000AE1268E34D00000010306C09C000B4000000B40094000C
00A45630A397804C7403E3B4180400F089FFFFFFFFFFFFFFFF4EB3DCC3
15-02-2016 13:48:01,995 TRACE [es.ugr.raziel.threads.scanners.WifiScannerThread]
Encontrado 1 paso wifi a -76db de 4C7403E3B418
000024002F4000A02008000000000000871868E34D00000010306C09C000B4000000B400940020
00A45630A397804C7403E3B4180400008AFFFFFFFFFFFFFFFFC108751A
15-02-2016 13:48:02,007 TRACE [es.ugr.raziel.threads.scanners.WifiScannerThread]
Encontrado 1 paso wifi a -70db de A456308F7CD0
000024002F4000A02008000000000000036986BE34D000000100C6C09C000BA000000BA00800000
00FFFFFFFFFFFFA456308F7CD0A456308F7CD0A0BB34D8C0D5EE0800006600311400076564757 2
6F616D01088C1218243048606C03
```

Figura 5.57
Ejemplo de tramas WiFi RAW capturadas.

Para discernir el tipo de paquete (Anexo A.2) se extraen los campos que definen la longitud de la cabecera (`headerLength`) y tipo de paquete (`packeType`). Con esa información el `BTScanner` es capaz de determinar en que posición se encuentra la dirección MAC del equipo que ha enviado la trama capturada (Código 5.34).

Además, consultando la cabecera `Radiotap` (Sección 5.1.2) se extrae la intensidad `RSSI` (Sección 5.1.2) con la que ha sido recibida la trama. Esto permite que se pueda configurar el `WiFiScanner` (Sección 5.26) para que descarte las tramas que tengan un `RSSI` inferior al umbral establecido. De igual manera, se descartan las tramas que el `WiFIScanner` no sea capaz de interpretar.

Código 5.34
RAZIEL: Módulo de captura de dispositivos WiFi: Escáner
El dispositivo se encuentra fuera del radio de acción del sensor, por lo que no es detectable.



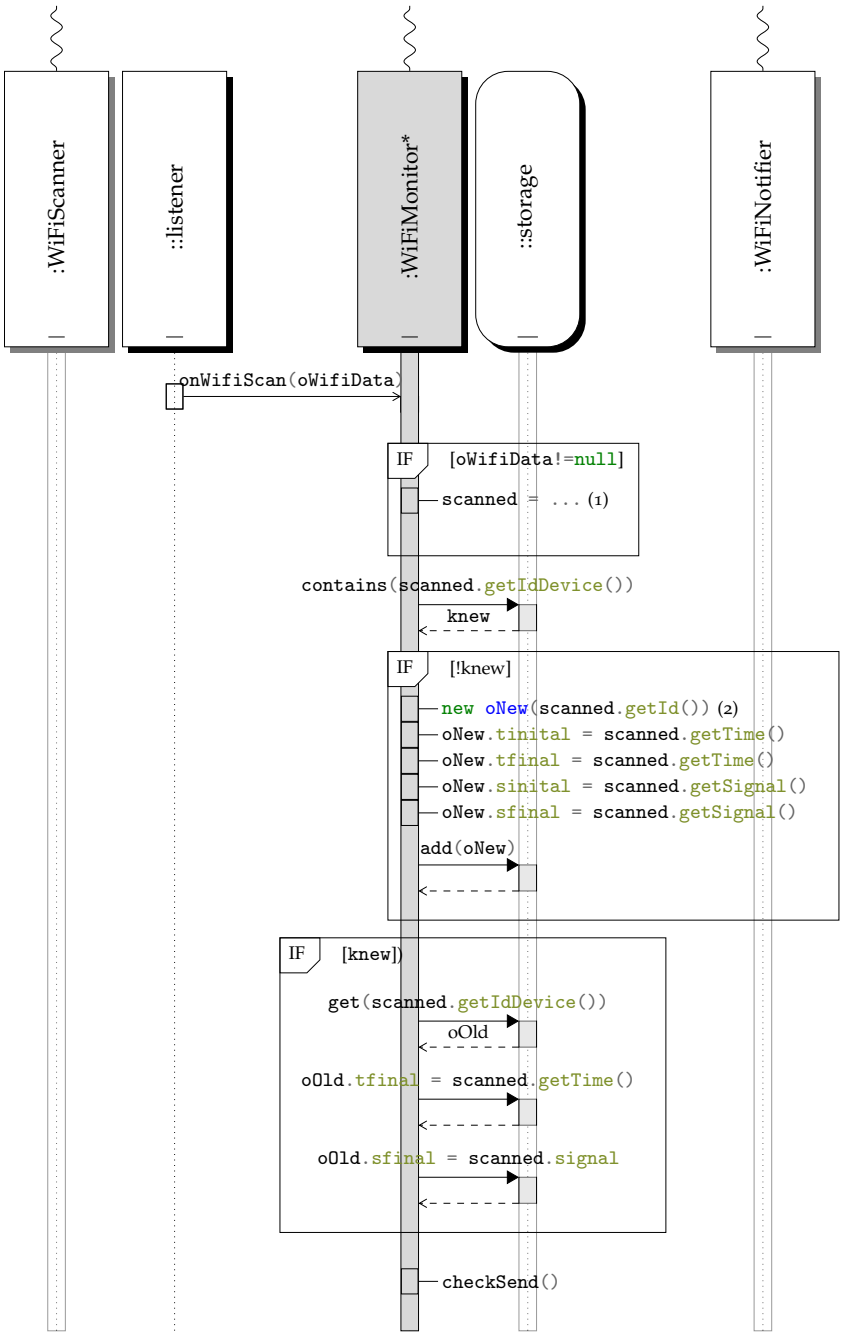
(1) Establece las posiciones donde están la dirección del emisor en la trama en función del tipo y subtipo de trama.
(2) Permite filtrar los paquetes en base al RSSI (Sección 5.1.2) definida en el fichero de configuración (Sección 5.26).

Una vez que se ha extraído la dirección MAC del emisor de la trama, se establece la marca del tiempo y se almacena la intensidad `RSSI` y se envía esta información al `WiFiMonitor`.

5.6.5.5.2 Monitor

El monitor es el encargado de procesar la información enviada por el `WiFiScanner` para llevar el control de la monitorización de los dispositivos (Código 5.35).

Código 5.35
RAZIEL: Módulo de captura de dispositivos WiFi: Monitor. El monitor recibe la información proporcionada por el escáner, y la emplea para actualizar la información que mantiene en sus dos contenedores.



(1) Realiza el cifrado de la MAC y la identificación del fabricante.

(2) Añade al paso la información sobre el nodo, su identificador y coordenadas.

Al igual que en el caso de `BTMonitor` y como se ha presentado en la Arquitectura (Sección 5.6.1), `WiFiMonitor` mantiene dos contenedores para los pases: `storage` para los dispositivos actualmente monitorizados y `toSendStorage` para los dispositivos que han de ser notificados al **servidor de cómputo**.

Igualmente a `BTMonitor`, para `storage` se opta por una estructura hash en memoria y para el contenedor `toSendStorage` un fichero `CSV`[45].

Cuando una **detección** es enviada del `WiFiScanner` al `WiFiMonitor`, este obtiene la información sobre el dispositivo detectado (Sección **??**). Si la información interpretada es válida, se consulta el contenedor `storage` si el dispositivo ya estaba siendo monitorizado en ese momento. En caso de que el dispositivo no haya sido monitorizado con anterioridad, se anota el paso del dispositivo, anotando los instante de tiempo de primera (`tinitial`) y última (`tfinal`) detección al instante de tiempo en el que ha sido detectado el dispositivo. Igualmente, se hace con la intensidad de detección inicial (`sinitial`) y final (`sfinal`), que se establece ambas a la de la primera detección.

Si el dispositivo ya estaba siendo monitorizado, es decir ya se encontraba su paso dentro del contenedor `storage`, se actualizan tanto la marca de tiempo final (`tfinal`) como la intensidad final (`sfinal`) a la de la última detección respectivamente.

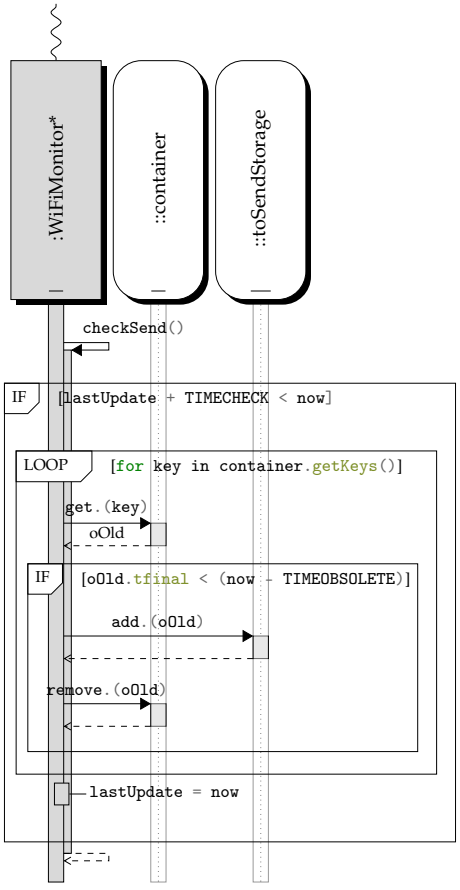
Una vez realizada la insercción o actualización, se procede con la función `checkSend`, que al igual que en el caso del `BTMonitor` se encarga de decidir que paquetes son los que se deben de notificar al `WiFiNotifier`. En este caso se han implementado dos modos presentado en las arquitectura (Sección 5.6.1.2).

45 ↑Para más información, puede leerse la Sección 5.6.5.4.2 donde se explica en más detalles estos contenedores en el caso del `BTMonitor`.

Modo paso

El modo paso o `step mode` (Sección 5.6.1.3.1) funciona de manera similar al único modo implementado para el `BTMonitor` (Código 5.36). De forma periódica (cada `TIMECHECK` segundos) se comprueban todos los pasos de dispositivos que están almacenados en `storage`. Aquellos que lleven más de un tiempo (`TIMEOBSOLETE`) son copiados al contenedor `toSendStorage` y eliminados del contenedor `storage`. Tanto el tiempo que determina el periodo de comprobación (`TIMECHECK`) como el que determina el tiempo mínimo para considerar el paso del dispositivo obsoleto (`TIMEOBSOLETE`) son determinados en la configuración del sotfware (Sección 5.6.5.3). Una vez se ha determinado que pasos son obsoletos, se actualiza la marca de tiempo de la última actualización (`lastUpdate`).

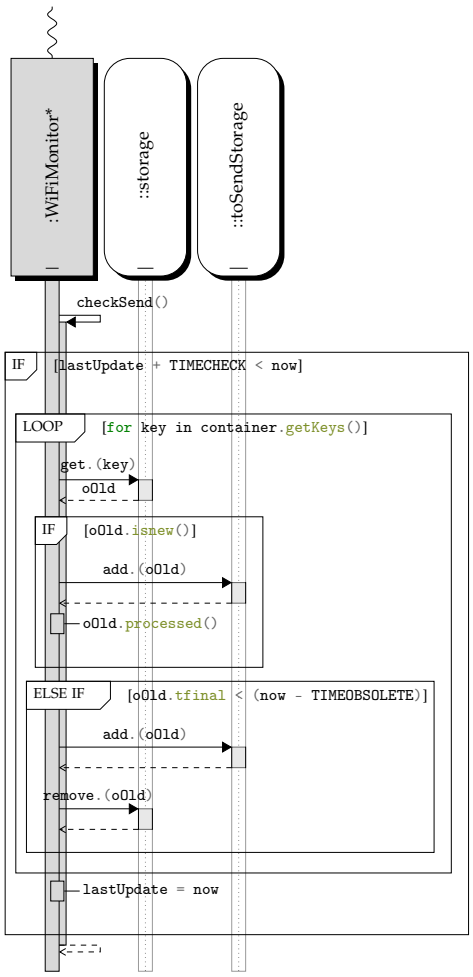
Código 5.36
RAZIEL: Módulo de captura de dispositivos WiFi: Monitor - Modo paso
El monitor determina que pasos de dispositivos se han quedado obsoletos, los elimina del contenedor en memoria y lo pasa al contenedor almacenado en disco. Esta operación, se realiza de forma periódica por parte del monitor.



Modo precoz

El modo precoz o `hasty mode` (Sección 5.6.1.3.2) notifica también los pasos de los dispositivos que están siendo actualmente monitorizados. Para ello, se añade una bandera a los pasos cuando son creados, indicando que esos pasos son considerados nuevos. De forma periódica (cada `TIMECHECK` segundos) se comprueba que los pasos de los dispositivos monitorizados sean nuevos, añadiendo dichos pasos al contenedor `toSendStorage` pero sin removerlos del contenedor `Storage`. En su lugar, la bandera que indica que el paso es nuevo se baja, de forma que no sea añadido en la siguiente comprobación. Para los pasos dispositivos cuya bandera de nuevo paso está bajada, se comprueba de forma análoga al anterior modo los pasos que se consideran obsoletos. Para los pasos considerados obsoletos, se copian al contenedor `toSendStorage` y se eliminan del contenedor `Storage`.

Código 5.37
RAZIEL: Módulo de captura de dispositivos WiFi: Monitor - Modo precoz.
El monitor,a demás de enviar los pasos que ha determinado obsoletos, también envía información sobre los pasos la primera vez que se realiza la comprobación.



5.6.5.5.3 Notifier

El notificador (`WiFiNotifier`) solamente es empleado cuando el **servidor de computo** se encuentra presente. Su cometido es el de enviar los pasos que el `WiFIMonitor` ha considero que tienen que ser enviados (Código 5.38). Para ello, mantiene una marca de tiempo con el instante en el que se realizó con éxito el último envío al servidor.

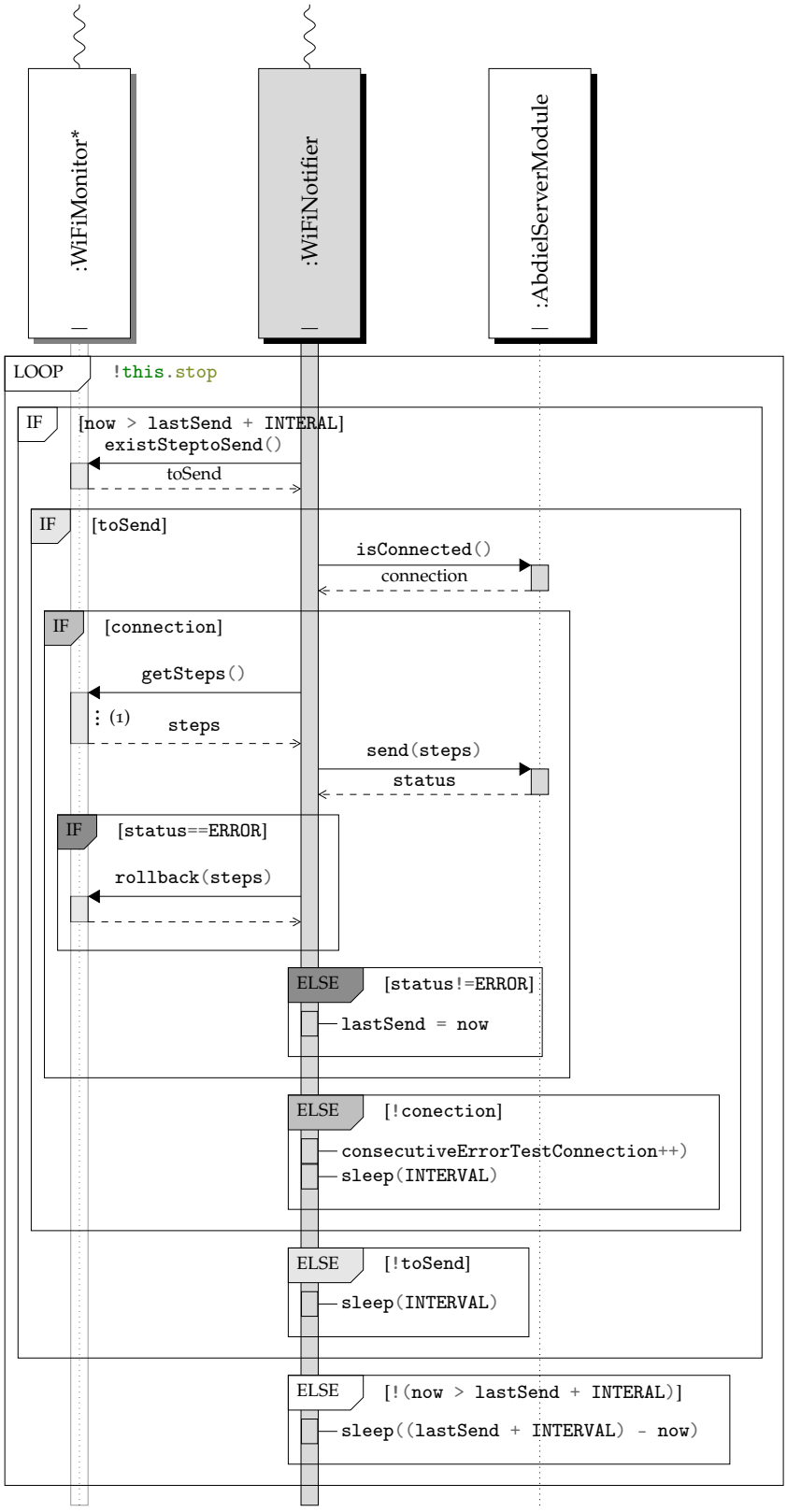
De forma periódica, `WiFiNotifier` le pregunta a `WiFiMonitor` si tiene pasos pendientes de enviar y al módulo de servidor `AbdielServerModule` si existe conexión con el **servidor de cómputo**. En caso de recibir respuesta afirmativa por parte de ambos, le solicita a `WiFiMonitor` que le envié los pasos que ha determinado han de ser enviados y los borra del contenedor `toSendStorage`.

`WiFiNotifier` envía los pasos al servidor mediante el módulo `AbdielServerModule`. En caso de no existir ningún error en el envío, se actualiza la fecha de último envío exitoso. En caso de error, `WiFiNotifier` devuelve los pasos a `WiFiMonitor` para que los almacene nuevamente.

En cualquier caso, `WiFiNotifier` se duerme hasta el próximo intervalo de tiempo, donde volverá a comprobar la conexión con el servidor y a solicitarle pasos al `WiFiMonitor`.

Al igual que en `BTNotifier` en caso de error con la conexión con el servidor, los pasos históricos registrados por el sistema, son almacenados por el monitor hasta que se restablezca la conexión o se acceda manualmente a la información del nodo.

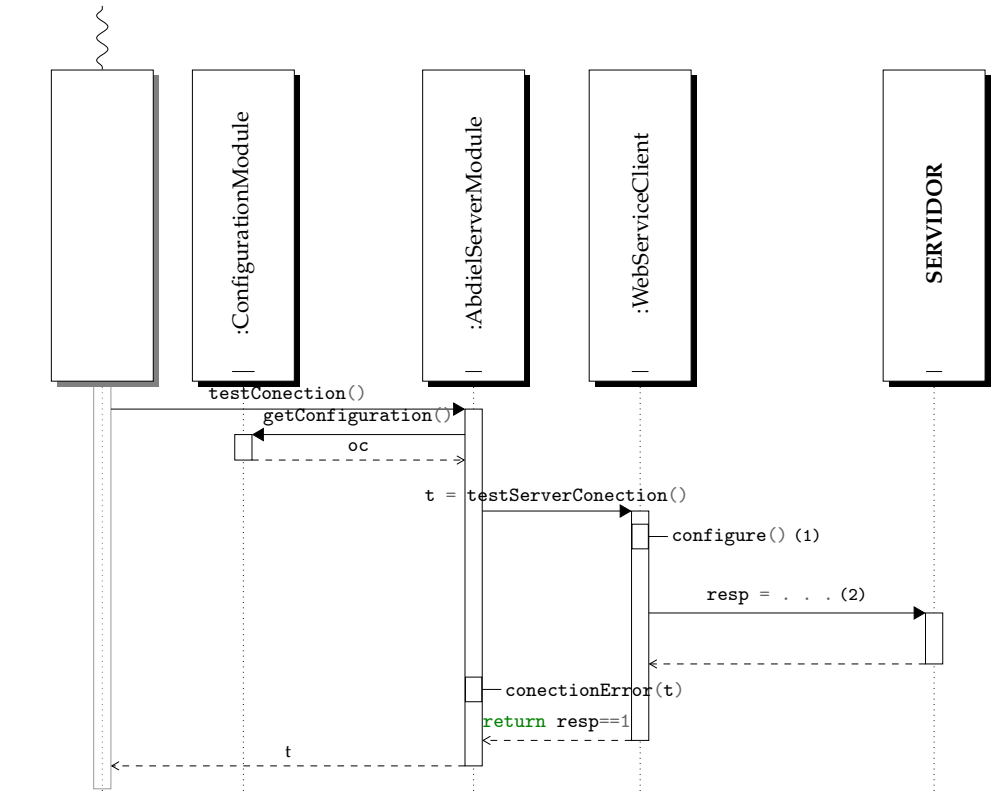
Código 5.38
RAZIEL: Módulo de captura de dispositivos WiFi: Notificador.
El notificador es el encargado de enviar periódicamente la información registrada por el monitor al servidor de cómputo.



5.6.5.6 *AbdielServer Module*

Este módulo es el encargado de proveer de la interfaz de comunicación con el **servidor de cómputo**. En la implementación consta de dos elementos (Código 5.39). El primero de ellos es la que funciona de interfaz con el resto de clases, denominada `AbdielServerModule`. Este se comunica con la clase denominada `WebServiceClient` que es la encargada de materializar las conexiones con cualquier servicio web, empleando en el caso de las invocaciones desde `AbdielServerModule` con el servidor definido en la configuración (Sección 5.6.5.3).

Código 5.39
RAZIEL: Módulo de comunicación con el servidor: Ejemplo de funcionamiento.
(1) Entre otros, crea y prepara el cliente de conexión con la `URL` almacenada en la configuración.
(2) Genera la petición al servidor y devuelve el contenido recibido en respuesta. Ver Sección ?? para más detalles.



Ambas partes son intercambiables, de esta forma, se podrían implementar distintas plataformas de comunicación de comunicación.

La clase `AbdielServerModule` consta de un mecanismo de control de errores, almacenando la marca de tiempo de la última conexión con éxito con el servidor (`lastContact`), así como una bandera en caso de error de conexión (`conectionError`) con el contador de los errores sucesivos. Estos mecanismos permiten ahora intentos de conexión en caso de situación de error con el servidor empleando la función `isConectd` vista en los notificadores (Figuras 5.32 y 5.38).

En esta sección, sólo se documentará el módulo desde la vista del cliente. La parte del servidor será abordada en la Sección 5.8). A continuación se presentan brevemente los métodos que implementa la clase `AbdielServerModule`.

testConection

Función encargada de comprobar el estado de conexión con el servidor, análogo a un "ping". La inclusión de este método obedece a la necesidad de comprobar que la conexión con el **servidor de cómputo** existe antes de intentar un envío de datos o ante cualquier otra operación. Devuelve un Booleano indicando si se ha podido establecer la conexión con el servidor en el ping enviado.

isConnected

Esta función se emplea conjuntamente con `testConection` para comprobar el estado del servidor. Se basa tanto en el envío del ping, como en la detección de errores anteriores o el tiempo de la última conexión exitosa, para determinar si la conexión con el servidor se encuentra disponible. Devuelve un booleano indicando si existe conexión con el servidor.

getNodes

Esta función solicita el listado de nodos para los que el usuario definido en el fichero de configuración (Sección 5.6.5.3) tiene permisos para enviar información. Las credenciales del servicio web pueden ser comunes para un grupo de nodos, o individuales para cada nodo[46]. Devuelve el listado de nodos para los que el usuario del servicio web dispone de permisos para enviar, así como la configuración actualizada asociada a cada uno de ellos.

startSession

Inicializa la sesión en el servidor para el nodo y los sensores indicados. El servidor sólo acepta información entrante en caso de que haya una sesión iniciada.

endSession

Finaliza la sesión en el servidor para el nodo y los sensores indicados. El servidor sólo acepta información entrante en caso de que haya una sesión iniciada.

46 ↑Decisión de diseño debido a que en una fase temprana del desarrollo se planteó la posibilidad de que sólo un nodo tuviese conexión directa al servidor e hiciese de intermediario para el envío de la información del resto de nodos.

createSteps

Realiza el envío de los pasos por parte del notificador. En primer lugar, divide en lotes de 500 pasos[47] y los empaqueta en un fichero `CSV`[48] que posteriormente es comprimido `CSV` empleando GZIP [70].

[70] GZIP file format specification version 4.3

getCurrentVersion

Obtiene el número de versión del software que según el servidor tiene que ejecutar el nodo.

getApp

Solicita la versión indicada del software RAZIEL .

getScript

Solicita al servidor si el nodo ha de ejecutar algún script de mantenimiento, de ser así, el servidor devuelve el script a ejecutar.

sendStatus

Envía mensajes de información genérica al servidor. Es empleado tanto para comunicar estados de error, consumos de red, resultados de ejecuciones de scripts solicitados o cualquier otro tipo de información sobre el nodo que deba ser conocida por el servidor.

5.6.5.7 *WiFi conection Module*

El módulo de conexión de red wifi o `WiFiConnectionModule` se encarga de gestionar la conexión de red empleando una red WiFi disponible. En el prototipo se ha implementado una conexión empleando el protocolo WPA, por ser el más extendido. Puede hacer uso de la misma tarjeta de red que la empleada para la monitorización WiFi, siempre y cuando el hardware lo soporte, o bien emplear una interfaz distinta. Esta interfaz es determinada en el software mediante su nombre en el sistema operativo y ha de ser indicado en el archivo de configuración.

El funcionamiento del módulo es simple, encargándose de comprobar periódicamente si la interfaz de red está levantada y de si la red inalámbrica está conectada. El funcionamiento del módulo se describe al detalle en el Código 5.40.

47 ↑Estos pasos serán cacheados en la inserción en la base de datos y se ha demostrado que este valor es el más optimo para ello. Ver experimento 5.6.2.

48 ↑Formato que menor tamaño genera el fichero resultante. Ver experimento 5.6.1.

Código 5.40
RAZIEL: Módulo de conexión a red por WiFi: WiFiConnectionModule
El módulo es el encargado de intentar la conexión a red mediante WiFI.

(1) Levanta la interfaz de red determinada.
(2) Emplea las credenciales de autenticación alojadas en la ruta de `this.wpa_file` para el intento de conexión WPA.
(3) Solicita mediante el DHCP credenciales de red al AP.
(4) Solicita al sistema operativo la información disponible sobre las interfaces de red activas.
(5) Comprobación de que la interfaz de red determinada está activa.
(6) Comprobación de que la interfaz de red determinada tiene dirección IP asignada.

Debido a que la conexión mediante red WiFi interfiere perjudicialmente con el funcionamiento de la monitorización WiFi, su empleo no ha pasado del entorno de pruebas en laboratorio. Es por ello que mecanismos de gestión del módulo más avanzados, como la contabilización de errores, no han sido implementados.

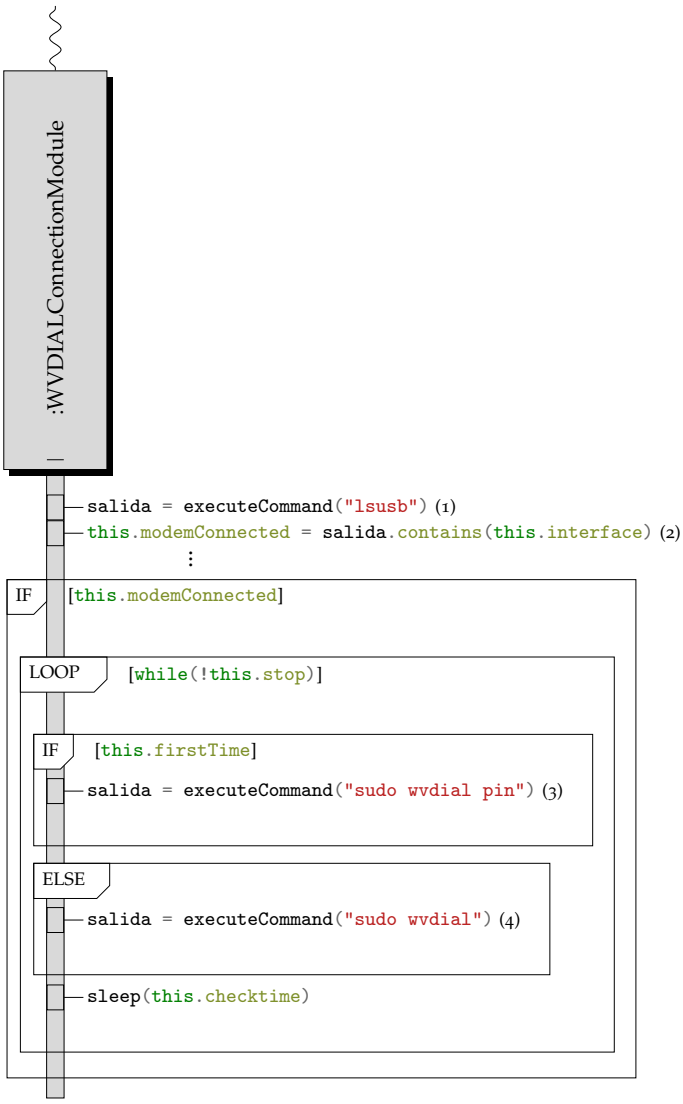
5.6.5.8 *3G Module*

Este módulo es en encargado de la gestión de la conexión 3G. En el prototipo se han implementado tanto módulos para SAKIS (Sección 5.5.3.3.1) como para WVDIAL (Sección 5.5.3.3.2). Ambos módulos a efectos prácticos son muy similares, aunque sólo el módulo con WVDIAL ha sido empleado fuera del laboratorio, por lo que será el que se presente.

El sistema requiere que en el archivo de configuración se indique el identificador de dispositivo del modem 3G a emplear. Este identificador es el

par fabricante-producto del modem 3G en su modo de funcionamiento de modem, por lo que es requerido que el script `usb_modeswitch` (Sección 76) haya funcionado correctamente para cambiar el modo de funcionamiento de almacenamiento masivo a modem. Si el modem se encuentra detectado se sigue la secuencia de pasos simplificada en el Código 5.41

Código 5.41
RAZIEL: Módulo de conexión a red por WiFi: 3GConnectionModule
El módulo es el encargado de intentar la conexión a red mediante 3G.



(1) Solicita información al sistema operativo sobre los dispositivos conectados mediante puertos USB.
(2) Comprueba si el modem identificado por el par fabricante-producto e indicado en la configuración del sistema, se encuentra conectado
(3) Inicia la marcación mediante la configuración de `wvdial`, añadido el paso adicional para proporcionar el pin de desbloqueo de la tarjeta SIM.
(4) Inicia la marcación mediante la configuración de `wvdial`.

Si la marcación empleando `WVDIAL` es exitosa, el proceso que la ha generado se quedará en ejecución hasta que la comunicación se interrumpa. Esto puede suceder porque el `ISP` haga que la conexión expire, porque se produzca un

corte de cobertura o por un error durante la transmisión. En cualquier caso, el módulo esperará un tiempo determinado en la configuración, antes de reintentar de nuevo la marcación.

Dado a que en los escenarios en los que se ha empleado este módulo, este es la única vía de comunicación con el servidor, es considerado crítico. Es por ello que se le ha dotado de mecanismos de recuperación ante errores que no han sido especificados en la Figura 5.41 para facilitar su interpretación.

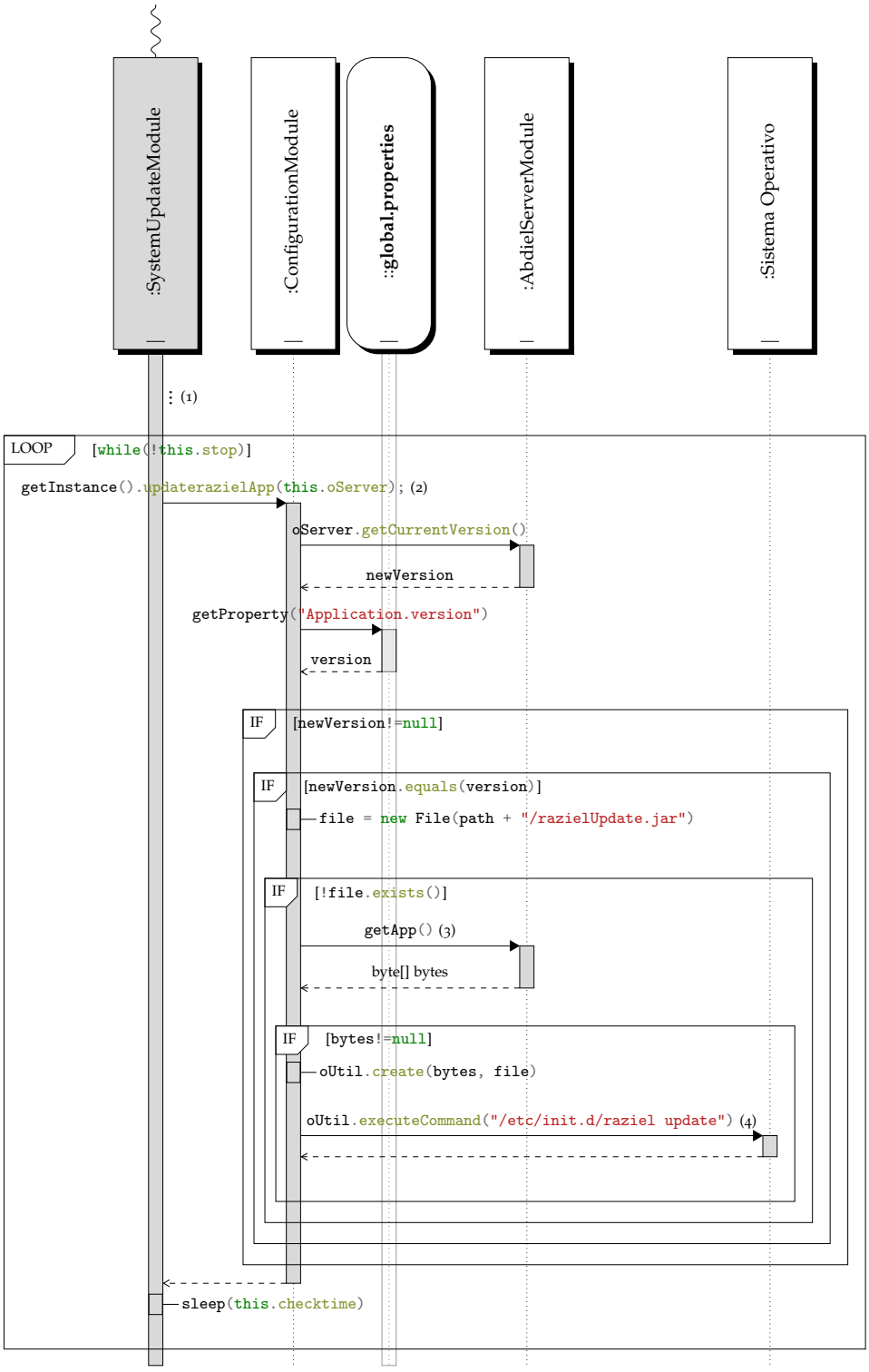
Debido a que las comunicaciones 3G disponibles comercialmente suelen facturar en función del tráfico consumido, ya sea tarificando en base al tráfico generado o por tarifas reguladas por consumo máximo, el control del tráfico de red generado es importante para no sobrepasar el tráfico presupuestado. Para ello se ha implementado un mecanismo de notificación al servidor del consumo de red generado. Esta información es enviada periódicamente al servidor por el módulo `EverythingItsFine` que será presentado más adelante. La decisión de emplear ese módulo, en lugar del propio módulo `3gConnectionModule`, es debido a que ese módulo es el encargado de notificar al servidor sobre todos los estados y aspectos relativos al funcionamiento del sistema.

5.6.5.9 *System Update Module*

El desarrollo de software es un proceso incremental y en constante crecimiento. La subsanación de errores, la implementación de nuevas funcionalidades o las mejoras en la eficiencia, pueden hacer necesario que el software del sistema RAZIEL necesite ser actualizado. Sin embargo, dado que los nodos están implantados en las zonas de monitorización y que su acceso tanto físico como remoto no está garantizado, se hace necesario que el propio sistema sea capaz de actualizarse a si mismo. Como se indico en la Sección 5.6.5.1, este provee de funcionalidad para realizar actualizaciones (y cancelación de actualizaciones).

Para ello se implementa el módulo `SystemUpdateModule` en cargado de realizar las comprobaciones de la versión en ejecución, con la que el servidor establece que es la necesaria. La parte del servidor, será concretada en la Sección 5.8 de este capítulo. La versión del software que está ejecutando el nodo se indica en el fichero de propiedades interno `global.properties` como se presentó en la Sección 5.6.5.3. Si el número de versión difiere con la establecida en el servidor, se descarga la nueva versión del software y se invoca la acción `update` del demonio (Código 5.42).

Código 5.42
RAZIEL: Módulo de actualización del sistema: SystemUpdateModule
El módulo es el encargado de la comprobación de versiones y las actualizaciones del sistema.



(1) Se omite la inicialización por no resultar relevante para la lógica del sistema.
(2) La respuesta a esta petición por parte del servidor se detalla en la Sección **??**.
(3) La respuesta a esta petición por parte del servidor se detalla en la Sección **??**.
(4) La parte del demonio se presenta en la Sección **??**.

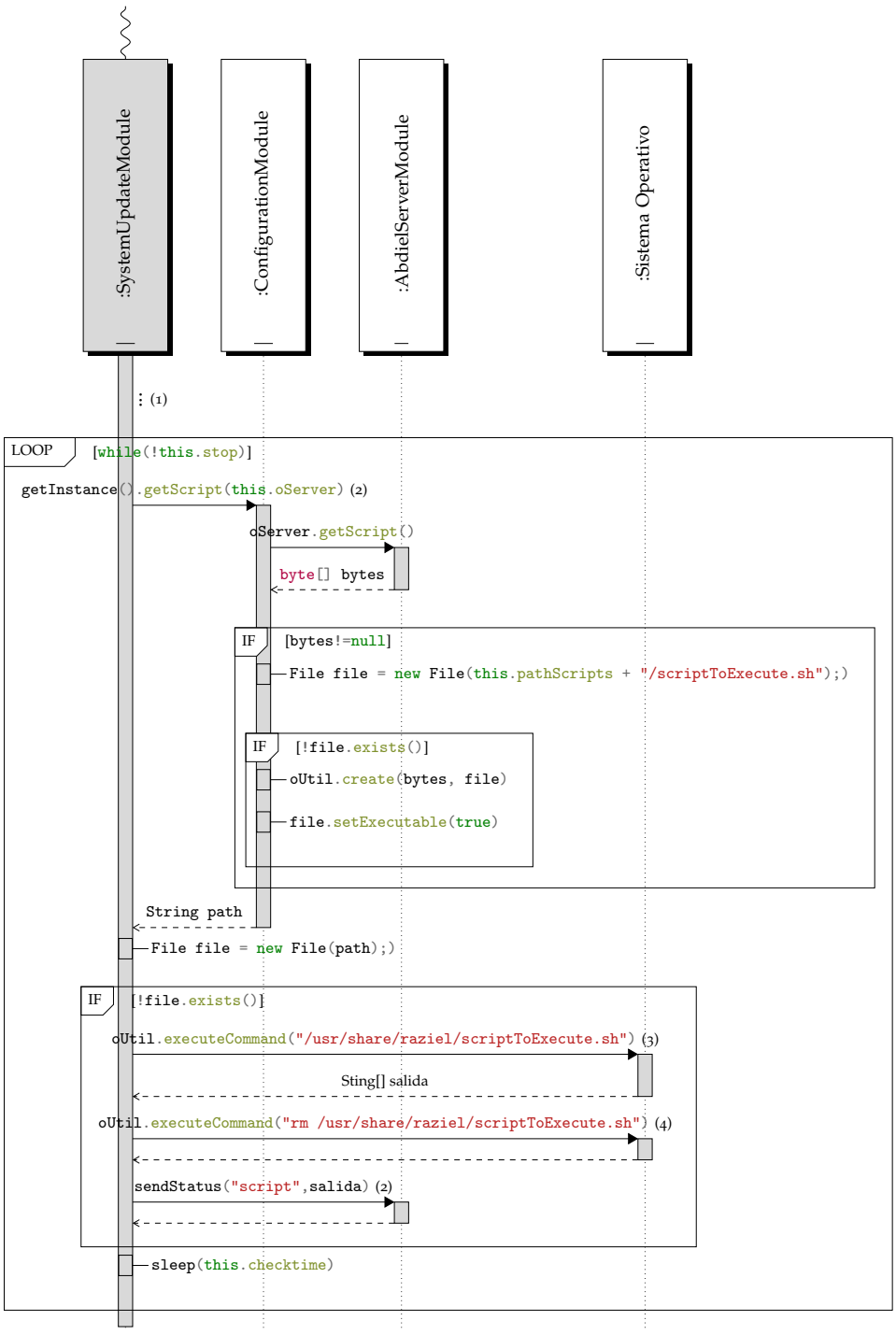
El sistema permite tanto actualizaciones a nuevas versiones, como a versiones antiguas, y permite establecer versiones del software distintas para cada nodo individualmente. Para garantizar la recuperación ante errores, el demonio realiza una copia de seguridad de la versión en ejecución del software antes de proceder a la instalación de la nueva versión. En caso de error, el demonio puede emprender la acción `undo-update` para restablecer la versión copia de seguridad. De esta forma, el se garantiza que existe siempre posibilidad de que el software de monitorización pueda ejecutarse.

5.6.5.10 *Scripts Module*

De igual manera que el software de monitorización, el sistema operativo puede requerir realizar modificaciones sobre el mismo, ya sea por tareas de mantenimiento o depuración. De esta tarea se encarga el módulo `ScriptsModule`.

Para ello, de forma periódica, consulta al servidor si el nodo debe de realizar la ejecución de algún script. De ser así, el servidor le envía el script a ejecutar, el módulo lo ejecuta y si este devuelve alguna salida, envía la parte final del mensaje de esta al servidor (Código 5.43). El envío de únicamente una parte de la salida del script al servidor, se impuso para evitar que por error se enviasen mensajes demasiado largos desde el nodo al servidor, consumiendo demasiado ancho de banda.

Código 5.43
RAZIEL: Módulo de scripts remotos: Scripts Module
El módulo es el encargado de la ejecución de scripts de forma remota en el sistema operativo.



(1) Se omite la inicialización por no resultar relevante para la lógica del sistema.
(2) La respuesta a esta petición por parte del servidor se detalla en la Sección **??**.
(3) Se ejecuta en el sistema el script descargado
(4) Una vez el script se ha ejecutado, se borra del sistema.

Este mecanismo de ejecución de scripts en remoto es muy útil en escenarios en los que no se tiene acceso al nodo, tanto físicamente, como mediante red.

Esto suele ocurrir cuando el nodo es implantado en un emplazamiento restringido (como un semáforo o panel de señalización), la red es otorgada por una infraestructura securizada por una tercera institución o se está haciendo uso 3G para las comunicaciones.

A modo de demostración de la aplicación de este módulo, se detallan alguno de los scripts remotos empleados en producción.

Ejemplo de uso: Conservación de ficheros de log

Debido a que como se ha descrito en la Secciones 5.5.1.2 y 5.5.1.1, los ficheros de LOG generados tanto por el sistema operativo como por el Software RAZIEL, son almacenados en memoria volátil. Esto impide que se pueda recuperar ficheros de LOG de periodos concretos (como durante la ocurrencia de un error) o ante un reinicio o apagado del nodo.

Para conservar los ficheros de LOG, se puede enviar el script remoto recogido en el Código 5.44, para que los ficheros de LOG sean copiados de forma persistente en la tarjeta microSD.

Código 5.44
Script de almacenaje permanente de LOGs

```
#!/bin/sh
folder_name = "/usr/share/raziel/logs/$date '+%Y-%m-%d_%H:%M:%S'$"
mkdir "$folder_name$"
cp /var/log/raziel/logs/* "$folder_name$"
```

Ejemplo de uso: Actualización de paquetes

Aunque la actualización del sistema operativo es considerada crítica, en ocasiones se ha requerido tanto la instalación de algún software adicional, o la actualización de algún paquete concreto. Para tal fin, se puede enviar el script remoto recogido en el Código 5.45.

Código 5.45
Script de actualización de paquetes

```
apt-get install --only-upgrade <nombre del paquete>
```

Ejemplo de uso: Acceso remoto por reverse shell code

Dado que en muchos escenarios no se dispone de acceso remoto al nodo y en ocasiones es necesaria una interoperabilidad más directa, sobre todo para tareas de depuración, se puede enviar el código (5.46) para perpetrar un acceso remoto mediante `Reverse Shell Code`.

Código 5.46
Script de autodestrucción del sistema

Servidor	Nodo
`sudo nc -l <puerto>`	`sudo bash > & /dev/tc/<ip servidor>/<puerto> 0>&1`

Ejemplo de uso: Autodestrucción

Dado que el sistema se emplaza en sitios públicos, son suceptibles a hurtos y manipulaciones por terceros. Para paliar esta situación, la de un robo de un nodo, o la intrusión de un nodo no identificado en el sistema, se puede emplear el Código 5.47 para eliminar totalmente la información de la tarjeta microSD.

Código 5.47
Script de autodestrucción del sistema

```
1          sudo apt-get install --only-upgrade <nombre del paquete>
```

De esta forma, si el nodo empieza a operar de forma anómala o comprometida, se puede fulminar tanto el sistema operativo, como el sistema de monitorización RAZIEL o los datos capturados no enviados al servidor.

5.6.5.11 *Email Module*

Desarrollado como una herramienta de depuración durante las fases tempranas del desarrollo del prototipo, su cometido es el de informar al servidor de estado del sistema, tanto durante la iniciación, en caso de error y de forma periódica. Además, permite adjuntar los ficheros de LOG generados durante la ejecución, lo que permitía la conservación de estos antes de que el módulo de ejecución de scripts en remoto estuviese implementado (Sección 5.6.5.10).

Su uso ha quedado descontinuado debido al consumo adicional de red que supone, que resultaba elevado en los casos en los que el nodo funcionase mediante 3G. Además, la cuenta de correo receptora añadía un tercer elemento ajeno al sistema, por lo que su uso se vio delegado en las versiones más avanzadas del prototipo, integrándose estas funcionalidades dentro de la plataforma de comunicación con el servidor (Secciones 5.6.5.6 y 5.8).

La notificación de estados de inicio (Sección **??**) y fin del sistema (Sección **??**) permiten controlar el estado de los nodos sin depender de consultar mensajes de correo.

La notificación de mensajes de errores se integró en el sistema dentro del sistema de mensajería provisto por el método `sendStatus` de la plataforma de comunicación (Sección **??**).

La comprobación de errores y consistencia del resto de sistemas, encargada originariamente a este módulo, dio lugar al subsistema denominado **EverythingItsFine**.

Sin embargo, se considera de recibo nombrar este sistema aunque actualmente se encuentre descontinuado o `deprecated`, ya que su funcionalidad como mecanismo de comunicación del nodo con el servidor durante las fases tempranas de desarrollo permitió la solución de innumerables errores y facilitó enormemente la tarea de depuración del sistema. Y aunque se encuentre descontinuado, su funcionalidad sigue vigente y podría ser activado nuevamente en los nodos en producción si así se necesitase.

5.6.5.12 *EverythingItsFine Module*

Tal y como se ha presentado en esta sección, el sistema **RAZIEL** es un engranaje de múltiples módulos y componentes interactuando entre ellos y con elementos externos al software, como el hardware de las interfaces de red o las comunicaciones con el servidor. Se implementa el módulo **EverythingItsFine** para gobernarlos a todos, encontrar errores y traer a estados consistentes el sistema, permitiendo operar de forma independiente y ciega a todos los demás módulos.

Este módulo implementa algunos mecanismos de detección de errores, y los mecanismos u operaciones que permiten salvaguardar la situación de error. La mayoría de los parches que se implementan en el sistema, son probados en primer lugar en este sistema. Muchos de estas soluciones, una vez probada su eficacia son integradas en la lógica interna del módulo correspondiente.

De forma periódica, el módulo comunica al servidor el estado o `STATUS` (Sección **??**) del sistema. Durante la escritura de esta tesis, la principal carga de desarrollo del prototipo se encuentra centrada en la integración de mayor cantidad de mensajes de estado, y su interpretación por parte del sistema. En un futuro, donde se deseé descentralizar tanto el procesamiento como almacenamiento de los datos, se podría emplear este mecanismo de comunicación únicamente para la cuantificación de dispositivos monitorizados, desligando al servidor de estas tareas.

De igual manera, la ejecución del módulo realiza una serie de comprobaciones periódicas sobre el buen funcionamiento del resto de módulos del sistema, realizando tareas de mantenimiento en caso de que algún módulo no esté operando con forme a lo esperado. Estas tareas pasan desde comprobar el estado de las interfaces de red monitorizadas o la consistencia de las fechas y horas. En caso de detectar alguna inconsistencia, el sistema puede intentar levantar las interfaces, reiniciar el módulo o incluso volver a reiniciar todo el sistema **RAZIEL**.

Este mecanismos de control de errores se suma al mecanismo interno de `try-catch` que se implementa en todos los módulos [49].

49 ↑Y que al igual que otros muchos otros elementos no constituyentes de la lógica de los módulos han sido omitidos en la descripción detalla de estos.

5.6.5.13 *Logger*

Debido a la gran cantidad de módulos interoperando y la dependencia del sistema **RAZIEL** con el hardware y eventos generados por estos, la tarea de la generación de ficheros de LOGs para la depuración resultan críticos para la detección y corrección de errores. Aunque mecanismos de comprobación de la calidad del software, como test unitarios y de integración han sido empleados para el desarrollo de cada módulo, la naturaleza hardware dependiente del sistema, dificulta sobremanera la realización de test de cobertura en todo el sistema.

Si bien, esta afirmación puede ser cuestionada, el esfuerzo necesario para el desarrollo de todos los simuladores requeridos[50] para este tipo de pruebas, sobrepasa en gran medida al esfuerzo de desarrollo de todo el prototipo funcional, y además, no garantizan que el software validado contra los test basados en simuladores sea funcional en el prototipo físico.

Escapa del ámbito de esta tesis definir al detalle el módulo, sin embargo se hace necesario nombrar que para la tarea de depuración ha sido empleada la librería `Apache log4j`[51], que permite realizar una depuración independiente para cada módulo del sistema y jerarquizada por niveles de prioridad de notificación.

5.6.5.14 *Utilidades*

Como se ha descrito al principio de la Sección 5.6.5, existe un último módulo del sistema no funcional que provee de una series de primitivas y herramientas funcionales no clase-dependientes que permiten garantizar una buena legibilidad de los módulos del sistema, al abstraer y encapsular extractos de código rutinarios requeridos en varios ámbitos, pero que no son constituyentes de la lógica interna de ningún módulo concreto.

Forman parte de este módulos funciones para la manipulación del sistema operativo tales como la gestión de gestión de las interfaces de red y sus modos o la ejecución de procesos y scripts, la conversión entre contenedores de objetos no primitivos, la manipulación de fechas, la gestión de ficheros, la compresión de conjuntos de datos, la realización de operaciones HASH, la conversión entre datos `RAW` binarios a cadenas de texto hexadecimal.

50 ↑Dispositivo hardware, sistema operativo, interfaces de red, eventos sobre estos,...
51 ↑https://logging.apache.org/log4j/2.x/

5.6.6 *Consideraciones de eficiencia sobre el Software RAZIEL*

Estudio 5.6.1: Escalabilidad del fichero de pasos a enviar. Comparativa entre CSV y JSON

Código 5.48
Consumo de memoria RAM del sistema operativo

```
-rw-rw-r--  1 antares antares 1200539 dic 20 14:21 pasos_10000.csv
-rw-rw-r--  1 antares antares 3722962 dic 20 14:44 pasos_10000.json
-rw-rw-r--  1 antares antares  127082 dic 20 14:21 pasos_1000.csv
-rw-rw-r--  1 antares antares  380104 dic 20 14:32 pasos_1000.json
-rw-rw-r--  1 antares antares   11508 dic 20 14:20 pasos_100.csv
-rw-rw-r--  1 antares antares   36811 dic 20 14:30 pasos_100.json
-rw-rw-r--  1 antares antares    1169 dic 20 14:20 pasos_10.csv
-rw-rw-r--  1 antares antares    3702 dic 20 14:29 pasos_10.json
-rw-rw-r--  1 antares antares     113 dic 20 14:20 pasos_1.csv
-rw-rw-r--  1 antares antares     369 dic 20 14:29 pasos_1.json
-rw-rw-r--  1 antares antares   59928 dic 20 14:21 pasos_500.csv
-rw-rw-r--  1 antares antares  186452 dic 20 14:30 pasos_500.json


-rw-rw-r--  1 antares antares 201219 dic 20 14:21 pasos_10000.csv.gz
-rw-rw-r--  1 antares antares 239513 dic 20 14:44 pasos_10000.json.gz
-rw-rw-r--  1 antares antares  18380 dic 20 14:21 pasos_1000.csv.gz
-rw-rw-r--  1 antares antares  21873 dic 20 14:32 pasos_1000.json.gz
-rw-rw-r--  1 antares antares   1742 dic 20 14:20 pasos_100.csv.gz
-rw-rw-r--  1 antares antares   2237 dic 20 14:30 pasos_100.json.gz
-rw-rw-r--  1 antares antares    316 dic 20 14:20 pasos_10.csv.gz
-rw-rw-r--  1 antares antares    546 dic 20 14:29 pasos_10.json.gz
-rw-rw-r--  1 antares antares    105 dic 20 14:20 pasos_1.csv.gz
-rw-rw-r--  1 antares antares    276 dic 20 14:29 pasos_1.json.gz
-rw-rw-r--  1 antares antares   7527 dic 20 14:21 pasos_500.csv.gz
-rw-rw-r--  1 antares antares   9093 dic 20 14:30 pasos_500.json.gz
```

Aquí habrá que poner algo, digo yo.

Estudio 5.6.2: Inserción de pasos en la base de datos. Eficiencia del mecanismo de caché

Código 5.49
Consumo de memoria RAM del sistema operativo

```
Inserción de 25164 pasos

Cache 1    -> 18.728733 s
Cache 10   -> 2.03695 s
Cache 50   -> 0.651283 s
Cache 100  -> 0.44023 s
Cache 500  -> 0.16746 s
Cache 1000 -> 0.22833 s
Cache 2000 -> 0.45601 s
```

Estudio 5.6.3: Eficiencia del Sistema Raziel

En el Estudio 5.5.1 se ha presentado el consumo de recursos del sistema operativo. Se completa dicho estudio con el siguiente, donde se mide el impacto del software de monitorización *RAZIEL*.

Para ello se someterá al software de monitorización a distintas cargas de trabajo. Se han realizado las mediciones en una Raspberry Pi 2, como la presentada en la Sección 5.4.1. En todos los casos, se ha dejado el sistema en funcionamiento durante 10 minutos antes de realizar la medición. No se ha configurado ningún sistema de filtrado de las tramas capturadas, con el fin de someter al mayor stress posible al sistema.

En carga baja el software lidia con un flujo de tráfico de red wifi de 100 paquetes detectables por segundo. En carga media, el flujo se indica a 400, carga alta 800 y carga muy alta 1000$fbps$. El porque la elección de estos valores se discutirá en el Estudio 6.1.6.

El consumo de CPU y MEMORIA del software se detalla en la Tabla 6.3.

Tabla 5.13
Consumo de recursos del Software RAZIEL ante 4 niveles de carga.

		BAJA		MEDIA		ALTA		MUY ALTA	
PID		% CPU	% MEM	% CPU	% MEM	% CPU	% MEM	% CPU	% MEM
4150	R	32.2	7.6	59.5	7.6	95.7	7.7	99.6	7.7
2366	S	1.0	7.6	2.7	7.6	3.7	7.7	4.0	7.7
2325	S	0.0	7.6	0.0	7.6	0.3	7.7	0.3	7.7
2385	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
2389	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
2526	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
2531	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
2533	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
2541	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
3584	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
3938	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4145	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4146	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4147	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4149	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4151	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4152	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4154	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
4424	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7
5572	S	0.0	7.6	0.0	7.6	0.0	7.7	0.0	7.7

La hebra cuyo identificador de proceso o `PID` es 4150 se corresponde con el sensor WiFi (Sección 5.6.5.5.1) que es el elemento crítico del sistema debido a que es el elemento software encargado del procesamiento de las tramas WiFi capturadas. Con una carga de trabajo alta la CPU se encuentra a niveles cercanos al 100 % de la ocupacuón de uso.

Sin embargo, el resto de elementos software del sistema de monitorización hacen un uso poco intensivo de los recursos, ya que únicamente son activados bajo demanada o de forma periódica.

La ejecución tanto del sistema operativo como del software de monitorización no supone una carga excesiva para la placa, lo cual otorga un gran margen de computabilidad para la escalabilidad del software ya sea mediante procesado de mayor volúmenes de datos o la inclusión de nuevas capacidades y procesamientos.

Sin embargo, el sensor wifi es un elemento crítico del sistema, debido a que es el que realiza la mayor parte del procesamiento. Cualquier mejora en la eficiencia de dicho método es bien recibida. Es por ello que se hacen mandatarios los mecanismos de filtrado que se han indicado a lo largo de la Sección 5.6.5.5.

5.7 SERVIDOR DE CÓMPUTO

En esta sección se describe brevemente la configuración y prestaciones del servidor de cómputo local empleado en el prototipo del sistema de monitorización propuesto. Se presentan algunas configuraciones que han sido relevantes para garantizar la eficiencia, alta disponibilidad y consistencia tanto del servidor como de los servicios ofertados por él. Estos servicios ofertados son presentados brevemente.

Aunque la tesis doctoral se enmarca en el ámbito de la arquitectura de computación, la arquitectura del servidor de cómputo será descrita de forma muy superficial en esta memoria de tesis, siendo abordada en mayor profundidad en el trabajo fin de máster denominado *Big Data en la predicción del tráfico: Optimización y Difusión en la Nube*.

5.7.1 *Big Data con un único servidor de computo*

La necesidad de establecer un servidor de cómputo local hardware frente a alternativas en la nube o virtuales, como ha sido presentado anteriormente, obedece a la imposición de las restricciones de las administraciones y entidades colaboradoras, para la incorporación de los nodos de monitorización a sus arquitecturas de red existentes. Las configuraciones pertinentes para la incorporación de un servidor físico determinado a una arquitectura de red son más sostenibles y fácilmente ejecutables que la concesión de esos mismos privilegios a un servicio en la nube. Además, para el personal habitual de la administración conceptos como la nube resultan muy difusos y es interpretado como .ªlgo"poco seguro y que ajeno a mecanismos de control estándar. Esta imposición, obliga a tener que disponer de una máquina física, concreta y identificable[52] que se encargue de recoger la información recopilada por los nodos de monitorización.

Además, la imposición de ser un sistema de bajo coste que se presentó en la Sección 5.2.1 influye directamente en el presupuesto disponible para el servidor de cómputo local, lo cual repercute en las prestaciones costeables del servidor. En la Sección 5.7.2 se recogen las modestas prestaciones computacionales del servidor de cómputo empleado en el prototipo. La no disponilidad de grandes granjas de computo, ha agudizado las prestaciones del propio sistema, con una premura por la eficiencia y escalabilidad de la arquitectura y procesamiento, cuyas limitaciones y crecimiento no podían ser subsanados arrojando los datos frente a cada vez mayor cantidad de recursos computacionales.

Aunque ha sido fruto de debate en numerosas ocasiones, sobre todo por los defensores más canónicos que argumentan que no se puede hacer Big Data con un único servidor de cómputo, todos los aspectos relacionados con los datos han sido implementados siguiendo los paradigmas y metodologías

52 ↑Y de ser requerido, desenchufable o incluso destruible.

empleadas, y comúnmente adoptados, en los escenarios de Big Data; con el fin de garantizar la mayor escalabilidad de los datos dentro de las prestaciones computaciones disponibles, así como, la viabilidad de la incorporación de mayores recursos computaciones en el futuro de forma sencilla.

Además, abordando una definición menos canónica del concepto de Big data, cada nodo de monitorización es a su vez un nodo de procesamiento del propio sistema, como ha sido presentado en la Sección 5.6. Es por ello que aunque el servidor de cómputo final sea el principal elemento de cómputo, cada nodo de monitorización realiza una ardua tarea de adquisición y de reducción de la información capturada, siguiendo la arquitectura presentada en la Sección 5.6.1.

Es necesario notar, como se presentará en los experimentos relacionados con el volumen de datos, que en muchas ocasiones varios GiB de datos RAW capturados por los nodos son almacenados resumidos en el servidor empleando escasos KiB de memoria conteniendo únicamente aquello que es potencialmente útil. Y aunque escapa del ámbito de esta tesis la discusión sobre la inclusión o no de un sistema con procesamiento de datos distribuido, como el propuesto, en el paradigma Big Data se considera necesario al menos presentar las argumentaciones pertinentes para al menos evitar que de forma tajante tanto el sistema, su arquitectura y las investigaciones derivadas sean infravaloradas y desconsideradas dentro del ámbito del Big Data. Es opinión del autor, que el Big Data no consiste en el procesamiento masivo de la mayor cantidad de datos ingentes posibles, sino del procesamiento ordenado, unificado, inteligente, reductivo y escalable de cantidades potencialmente inabordables de datos. Opinión que comparten cada más autores en los últimos años [95, 104, 210, 232, 303].

[95, 104, 210, 232, 303] Beyond the hype: Big data concepts, methods, and analytics, Big Data: It's Not the Size That Matters, Big data is not about size: When data transform scholarship, You may not need big data after all, It is not about size: a further thought on big data

5.7.2 *Características del servidor de cómputo local*

Se recogen de forma testimonial las características del servidor de cómputo en la Tabla 5.14.

Tabla 5.14
Características del Servidor de cómputo local

PROCESADOR	Intel(R) Core(TM) i5-4430 CPU @ 3.00GHz
MEMORIA	16 GiB DDR3 1333MHz
TARJETA GRÁFICA	NVIDIA GT 630
DISCO DURO 1	Westerm Digital 500GB 7200RPM
DISCO DURO 2	Westerm Digital 500GB 7200RPM

Como sistema operativo se instala inicialmente Ubuntu Server 12.04.03 LTS con el kernel GNU/Linux 3.5.0-40-generic x86_64. La versión actual del servidor corre Ubuntu Server 14.04.1 LTS con el kernel 3.13.0-43-generic x86_64. La migración de una versión a otra fue un proceso comprometido del servidor. Se opta por este sistema operativo frente a las alternativas planteadas (Debian,

CentOS, Ubuntu 13.10) debido a predilecciones personales y familiaridad con el entorno, además de suponer un entorno Libre y disponible para todos los investigadores del proyecto.

Se configura un entorno LAMP [160] para el servidor para la implementación de la API Rest descrita en la Sección 5.8 así como para alguno de los servicios descritos en la Sección **??** que además del procesado de datos servirá algunos servicios y plataformas WEB.

[160] *Open Source Development with LAMP: Using Linux, Apache, MySQL and PHP*

5.7.3 *Configuraciones destacables en el servidor*

Si bien las configuraciones realizadas en el servidor han sido numerosas, dos de ellas son suficientemente relevantes para el ámbito de esta tesis como para ser descritas brevemente debido a su impacto en la consistencia y eficiencia de los datos. La última,

5.7.3.1 *RAID*

La seguridad e integridad de los datos debe estar garantizada en el servidor de cómputo local, además de optimizar las operaciones de E/S. Para ello se establece una configuración de discos duros en RAID1 [207].

[207] *A case for redundant arrays of inexpensive disks (RAID)*

El RAID 1 crea una copia exacta (o espejo) de los datos en dos o más disco. Esta configuración premia en rendimiento a la lectura de los datos, incrementando el rendimiento como múltiplo lineal del número de copias. Esto implica que un sistema con un RAID1 constituido con 2 discos duros puede estar leyendo simultáneamente dos datos diferentes en los dos discos, consiguiendo en teoría duplicar el rendimiento efectivo. Gracias a esto el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo la tasa de trasferencia, con el único límite de velocidad soportada por la controladora.

En el servidor, se adquirió una placa base con soporte para RAID1 de forma física (no simulada por software) mediante dos controladoras de disco independiente, una para cada disco (spliting o duplexing).

En las operaciones de escritura, el conjunto RAID1 se comporta como un único disco duro, dado que los datos deben ser escritos en todos los disco que conforman el RAID. Esto implica que el rendimiento de las operaciones de escritura no se ve influenciado por el RAID. Se considera esta configuración como óptima debido a que la naturaleza de los datos operables. Se estima que cada dato del sistema sea escrito una única vez, cuando es enviado del nodo al servidor de cómputo, permaneciendo inalterable a lo largo del tiempo. La lectura de los datos, en cambio, ha de ser realizada en múltiples ocasiones, en función de los algoritmos de procesamiento que se realicen, siendo susceptible cada dato de ser leído una o más veces veces por cada algoritmo ejecutado. Por tanto resulta óptimo disponer de una configuración

que premie la lectura de los datos frente a la escritura, como la que otorga la disposición de los discos en RAID1.

Además, un sistema RAID1 tiene muchas ventajas a la hora de la administración del sistema. Por ejemplo, en entornos 24/7 como el esperable, es posible marcar un disco como "inactivo" para la realización de una copia de seguridad de dicho disco, reconstruyéndose el RAID de forma automática al volver a marcar el disco como activo. Esto supone una gran baza a la hora de obtener instantáneas de los datos. Este mismo sistema de "reconstrucción" funciona igualmente en caso de fallo físico de uno de los discos, con lo cual se mantiene siempre un respaldo del disco duro, lo cual supone un mecanismo de seguridad adicional al estar los datos almacenados por duplicado.

5.7.3.2 *Sistema noatime*

Dado que la eficiencia de las operaciones de lectura frente a las de escritura, la habilitación del sistema NoAtime de las tablas de ficheros resulta favorecedora. La principal peculiaridad de esta configuración, es que en los sistemas de archivos convencionales, cada operación de lectura de un fichero conlleva una escritura, para almacenar la fecha de último acceso. Esta información es prescindible frente a la eficiencia conseguida, que es puesta a prueba en el siguiente estudio (Estudio 5.7.1).

Estudio 5.7.1: Velocidad de lectura en particiones NOATIME

Se ejecutará a los 10 minutos del inicio del sistema operativo el comando:

```
time find /etc -name ".*exec cat '' ";» /dev/null 2>/dev/null
```

Dicho comando mide el tiempo del sistema necesario para acceder a todos los ficheros alojado en `/etc` "imprimirlos por pantalla" (aunque se vuelva a salida nula).

Ejecutamos el comando tanto en nuestro sistema por defecto, como en nuestra configuración no-atime:

	Sistema por defecto	Sistema no-atime
Tiempo	2.632*s*	0.037*s*
Ganacia	1	2 049 729

Tabla 5.15
Tiempos de lectura en disco en un sistema por defecto y en un sistema no-atime

Como se puede observar, la mejora es increíblemente significativa. Es por ello que en nuestro entorno preferimos sacrificar la existencia del campo atime, frente a unos accesos a disco muchos más rápidos, que es el fin máximo de este capítulo.

5.7.4 *Entornos de trabajo implementados*

En esta sección se presentan los distintos entornos, funcionalidades y servicios que son ofertados en el servidor directamente relacionados con la lógica funcional del sistema de monitorización y la difusión de resultados. Muchas de estas funcionalidades son descritas con más detalle en secciones futuras de esta memoria, por lo que sirva esta sección como pequeña introducción y puesta en común de los distintos enfoques de uso del servidor.

5.7.4.1 *Almacenamiento local: Servidor MySQL*

Como se ha comentado en la arquitectura del sistema (Sección 5.3) el principal cometido del servidor es el de servir de almacenamiento para la información capturada y enviada por los nodos de monitorización. Los pasos enviados por los distintos nodos funcionando de forma simultánea, son recopilados y almacenados por el servidor de cómputo. Es por ello que el principal escenario o servicio ofertado del servidor es el de ser un almacén consistente y perdurable.

Para ello, se implementa un sistema de base de datos que permita el almacenamiento eficiente de los datos recibidos por el servidor. Se implementa una versión de MySQL optimizada para favorecer las operaciones requeridas del sistema, concretamente y en la actualidad, la versión 5.6.40. Dado el carácter trascendente del almacenamiento de los datos en el sistema, la Sección 5.9 de este capítulo se centra en las optimizaciones y arquitectura del modelo para el almacenamiento eficiente y escalable de los datos.

5.7.4.2 *Almacenamiento nube: EZEQUIEL*

Dado que los datos son almacenados en base de datos internas, el servidor es el origen y entorno de ejecución del software `EZEQUIEL`, encargo de procesar los datos obtenidos, generar resúmenes y difundir los resultados obtenidos. Estos resultados son publicados en la nube para facilitar su exportación e interpretación por agentes externos al sistema. La parte del almacenamiento en la nube y la peculiaridades del software `EZEQUIEL` son descritos en la sección 5.13 de este mismo capítulo.

5.7.4.3 *Comunicación con los nodos de monitorización*

Para que los datos capturados por los nodos de monitorización lleguen al servidor, se hace necesario implementar una arquitectura de comunicación entre ambos elementos. Esta comunicación se realiza por medio de una `API REST` implementada en el servidor, accesible por los nodos mediante conexión a internet provista por alguno de los métodos presentados en la Sección 5.5.3.

Esta `API` sirve de enlace entre los nodos y la base de datos del servidor, ofreciendo todas las funcionalidades requeridas para la comunicación del

software RAZIEL presentadas en las Sección 5.6.5.6 con la base de datos. La arquitectura de comunicación se presenta en la Sección 5.8 (Página 246) de este mismo capítulo.

5.7.4.4 *Panel de control*

Para la gestión de la información de los nodos registrados en el sistema, se implementa un panel de control de los mismos accesible mediante la web para los usuarios registrados del sistema. Este servicio es ofertado por el servidor debido a que la información sobre los nodos es generada y almacenada en el mismo servidor. Este panel de control, se describe brevemente en la Sección 5.14.3.

5.7.4.5 *Plataforma de difusión WEB*

Mucha de la información publicitaria, de difusión y divulgación sobre los distintos proyectos relacionados con Mobywit son alojados en el propio servidor web del servidor mediante el dominio http://mobility.ugr.es. Esta y otras herramientas de publicacion son presentadas en la Sección 5.14.

5.7.4.6 *Sistema propio de control de versiones: GitLab*

Para la gestión eficiente del código desarrollado se instancia un servidor de GitLab[53], que ofrece un servicio web de control de versiones y desarrollo de software colaborativo basado en git[54] (Figura **??**).

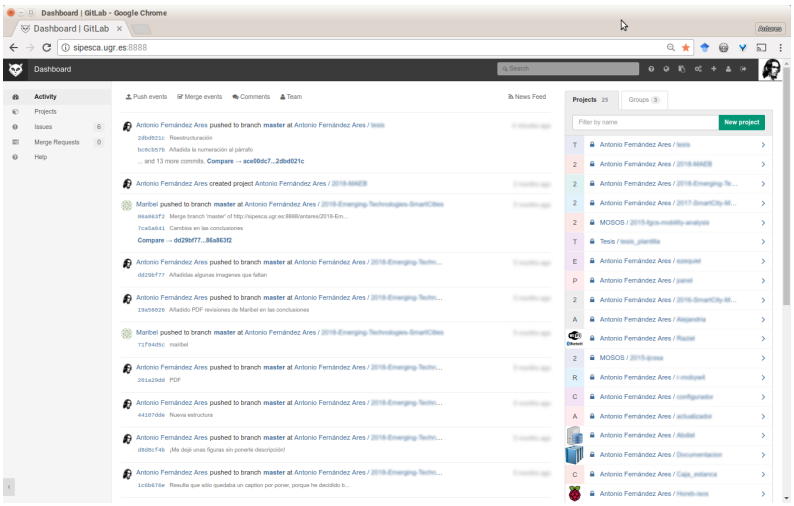


Figura 5.58
Portal de Gitlab implantando en el servidor para el control de versiones.
(*) Alguna información sensible o no relevante ha sido emborronada en la captura.

53 ↑https://gitlab.com/gitlab-org/gitlab-ce
54 ↑https://git-scm.com/

De esta forma tanto el código fuente del software desarrollado, las modificaciones del sistema operativo y cualquier otro aspecto del sistema de monitorización se encuentra centralizado en el servidor.

5.7.4.7 *Administración remota del servidor con Webmin*

Para la gestión del servidor en remoto, se instancia un administrador remoto por interfaz web denominado `WEBMIN`[55], configurado para que sea accesible solamente desde un par de equipos determinados mediante cortafuego.

5.7.4.8 *Procesamiento y difusión de resultados: RStudio*

`RStudio`[56] es en entorno en el servidor que ha terminado siendo indispensable para el análisis y procesamientos de datos en el ámbito de esta tesis. En el servidor se implanta una versión de `RStudio Server`[57], que permite disponer de un entorno de desarrollo integrado para el lenguaje de programación `R`.

Para ello se dispone de un `IDE`[58] accesible mediante un navegador web, que es ejecutado en el propio servidor de cómputo. Esto permite mover la computación aplicada de los datos al mismo entorno donde están alojados, prescindiendo de innecesarias trasferencias de datos entre distintas máquinas, pero permitiendo que el entorno sea accesible para múltiples usuarios de forma simultánea desde cualquier lugar.

El empleo aplicado de `R` se presentá en más detalle en la Sección 5.10, donde se describe la biblioteca de R *Mobywit* desarrollada para el procesamiento eficiente de la información. Adicionalmente, en la Sección 5.14.5 se aborda la generación de informes procedurales empleando `RMarkdown`.

La práctica totalidad de experimentos desarrollados en la tesis doctoral, han sido ejecutados en el servidor de computo empleando `RStudio` tanto para el desarrollo de la librería y sus procedimientos, como la extracción de resultados y su representación. Es por ello, que dedicar una lineas en la memoria de tesis a una herramienta que ha aportado tanto a la metodología eficiente de trabajo, resulta de recibo.

55 ↑`http://www.webmin.com/`
56 ↑`https://www.rstudio.com/`
57 ↑`https://www.rstudio.com/products/rstudio-server/`
58 ↑Integrated Development Environment o entorno de desarrollo integrado

5.8 ARQUITECTURA DE COMUNICACIÓN

En la Sección 5.6.5.6 se ha presentado la parte del cliente de la plataforma desarrollada para la comunicación entre los nodos de monitorización y el servidor de cómputo. En esta sección se detalla la parte del servidor, presentando brevemente en primer lugar las herramientas empleadas para su desarrollo, los mecanismos de seguridad desarrollados y la funcionalidad ofrecida por la API. Esta información se complementa con el Anexo A.4, donde se recoge una descripción detallada de cada función.

5.8.1 *Entorno de desarrollo de la API de Comunicaciones*

Se ha desarrollado una `API RESTfull` empleando `PHP`[59] `5.6`[60] potenciado por `Laravel Framework`[61] `5.2.23`[60].

[82] Architectural styles and the design of network-based software architectures

[83] RFC 2616 - Hypertext transfer protocol–HTTP/1.1

Una `API RESTfull` [82] es un servicio web que sigue los principios arquitectónicos `REST` transmitiendo mediante `HTTP` [83]. Estos principios se basan en una serie diseños funcionales clave, siendo el más destacable de ellos, que las comunicaciones entre cliente y servidor carecen de estados, contiendo el mensaje toda la información requerida para la comunicación.

En lugar de una `API RESTfull` podría haberse empleado otro entorno de comunicaciones, como por ejemplo `SOAP` que en la práctica resulta un protocolo más robusto, seguro y con un fuerte tipado. Sin embargo las `API RESTfull` se han convertido en los últimos años en la tecnología de comunicación predominante en el ámbito del Internet de las Cosas [146] debido a que requiere menos recursos tanto en el cliente como en el servidor, y que resulta mucho más flexible y escalable.

[146] RESTful Design for Internet of Things Systems

El único punto vulnerable de las `API RESTfull` es la seguridad de las mismas, tema que es abordado en la Sección 5.8.2.

5.8.1.1 *PHP*

A pesar de que PHP tiene una gran cantidad de detractores, sigue siendo uno de los principales lenguajes de programación empleados para el desarrollo web actualmente [134]. Dispone de una gran cantidad de frameworks que lo potencian, una gran base de librerías desarrolladas para él, una documentación online muy detallada, gran soporte de la comunidad y un bagaje histórico de más de 20 años.

[134] Interactive: The Top Programming Languages 2018

Si bien existen numerosas alternativas para el desarrollo de la `API RESTfull` como `Java` (`JSF`, `Spark` o `Struts`), `Ruby` (`Ruby on Rails`), `Python` (`Django`) o

59 ↑`http://www.php.net/`

60 ↑Las versiones indicadas son las empleadas en el desarrollo inicial, no las versiones actuales en producción.

61 ↑`https://laravel.com/`

Javascript (Node.js) no es potestad de esta tesis determinar las ventajas y desventajas de las distintas alternativas para el desarrollo web, ya que existen innumerables estudios y comparativas sobre la temática y sus resultados nunca son concluyentes.

La elección de este lenguaje obedece principalmente a la predilección del desarrollador y autor de la tesis, en base a su experiencia previa con el mismo.

Sin embargo, dada la modularidad del diseño del sistema, las herramientas empleadas para el desarrollo de la plataforma de comunicación podrían ser fácilmente sustituible por cualquier otro lenguaje o plataforma, bien para la realización de comparativas o la solucionar futuros problemas de escalabilidad.

Sin embargo, debido al óptimo desempeño de la API desarrollada durante el prototipo, este tipo de estudios no ha sido requerido.

5.8.1.2 *Laravel*

Laravel ofrece un soporte completo para aplicaciones basadas en la arquitectura modelo-vista-controlador o MVC así como mecanismos nativos para el ruteo RESTful.

Está basado en Composer[62] para la gestión de las librerías y módulos empleados en el proyecto tanto nativos a Laravel como de terceros, lo cual facilita en gran medida la gestión de versiones de las dependencias del software.

Emplea Blade[63] como motor de plantillas para las vistas, lo que permite unificar y modularizar de forma sencilla los formatos de salida tanto de la API, así como cualquier otra vista de salida del sistema, como gráficas, widgets, tablas o webs.

Para el acceso y gestión de las bases de datos ofrece dos mecanismos principales Eloquent ORM[64] para la gestión interactiva de los modelos y Query Builder[65] para la creación y ejecución de consultas SQL genéricas directamente en la base de datos, de forma abstracta al servidor de base de datos empleado.

Finalmente, destacar que incorpora la librería Carbon[66] que ofrece una gran cantidad de operaciones comunes para el tratamiento de fechas. Estas operaciones incluyen desde diferentes conversores de formatos, operaciones para la adición y sustracción de unidades de tiempo o la comprobación de pertenencia a intervalos entre otros.

62 ↑https://getcomposer.org/
63 ↑https://laravel.com/docs/5.6/blade
64 ↑https://laravel.com/docs/5.0/eloquent
65 ↑https://laravel.com/docs/5.6/queries
66 ↑https://carbon.nesbot.com/docs/

5.8.2 *Securización*

La seguridad de las comunicaciones entre nodos y el servidor de cómputo es de vital importancia, sobre todo debido a que la `API RESTfull` por su naturaleza, contiene en el mensaje toda la información requerida para la interpretación y aceptación del mensaje. Esto implica que cualquier tercero que capture el mensaje transmitido, puede emitir nuevos mensajes falsos, que serán interpretados y aceptados por el servidor, como si hubiesen sido emitidos por el nodo.

Es por ello que se hace necesario implementar elementos de seguridad adicionales a la `API RESTfull`. Estos elementos son descritos a continuación.

5.8.2.1 *Comunicación segura mediante Https*

[225] RFC 2660 - The secure hypertext transfer protocol

[224] RFC 2818 - Http over tls

Para securizar las comunicaciones se emplea `HTTPS` [225] que permite cifrar la transferencia mediante SSL/TLS [224], empleando en el prototipo un certificado auto-firmado o `Self-signed certificate`. La seguridad de las transacción se basa en el principio de intercambio de claves entre cliente y servidor y en el concepto criptográfico de claves públicas y privadas.

Sin entrar en demasiado detalle, la clave pública es empleada para cifrar la transmisión y sólo es descifrable empleando la clave privada. Al iniciar una comunicación entre el cliente (nodo) y el servidor, el cliente genera una clave temporal que cifra con la clave pública del servidor y se la envía. El servidor, descifra la clave recibida mediante el empleo de su clave privada. La clave temporal es empleada entonces para cifrar y descifrar los mensajes de la comunicación.

Se ha optado por emplear una clave pública autofirmada por comodidad en el prototipo, aunque lo idóneo sería que esta estuviese firmada por una autoridad de certificación.

Para generar las claves se emplea el Código 5.50, donde es necesario definir las rutas donde se almacenarán las claves pública y privada generadas. Adicionalmente se nos solicitará información durante el proceso, relativa al lugar y organización que hará uso del certificado.

Código 5.50
`Script` para la generación del par clave pública y clave privada para el cifrado HTTPS. El fichero `crt/mobywit.key` contiene la clave privada y el fichero `key/mobywit.crt` contiene la clave pública.

```
1 mkdir crt
2 mkdir key
3 openssl req -new -x509 -days 365 -keyout key/mobywit.key -out crt/mobywit.crt -nodes
```

A continuación, se debe editar el fichero del VirtualHost correspondiente de APACHE, y añadir las líneas indicadas en el código 5.51

Código 5.51
Lineas a añadir a la configuración del Host virtual en APACHE para habilitar HTTPS.

```
1  ....
2  SSLEngine On
3  SSLCertificateFile /etc/apache2/ssl/crt/mobywit.crt
4  SSLCertificateKeyFile /etc/apache2/ssl/key/mobywit.key
5  ...
```

Siempre que no se comprometa la integridad de la clave privada alojada en el servidor, la comunicación entre los nodos y el servidor estará securizada mediante el cifrado por medio de `HTTPS`. Si bien implementar esta capa de seguridad implica un volumen de tráfico mayor y una relentización de las comunicaciones por el cifrado y descifrado, su uso permite que ningún elemento intermedio enrutador entre el nodo y el servidor pueda interpretar las información transmitida. Sin embargo, el empleo de HTTPS sólo securiza las transmisiones entre clientes y servidores, sin ofrecer ningún mecanismo que permita restringir el acceso y empleo de la `API RESTfull`.

5.8.2.2 *Identificación del nodo en el sistema*

Como se ha visto en la sección 5.6.3 cada nodo de monitorización se asocia con un identificador dentro del sistema y unas credenciales de conexión mediante un par usuario-contraseña.

Dentro del sistema, un usuario definido puede tener varios nodos de monitorización asociados, de forma que varios nodos compartan las credenciales de conexión. En el prototipo, por comodidad, para cada proyecto donde se ha hecho uso del sistema, se ha creado un par usuario-contraseña distinto.

Debido a la naturaleza de la `API-RESTfull` para cualquier tarea de comunicación del nodo con el servidor, es necesario proveer dichas credenciales en el mensaje. Estas credenciales son enviadas mediante variables POST, dependiendo del cifrado HTTPS para evitar que las credenciales sea interceptadas. Si las credenciales provistas son válidas, el servidor acepta la petición.

Las credenciales de cada nodo se almacenan en texto plano dentro del fichero de configuración del software RAZIEL, como se presentó en la sección 5.6.3. Esto supone una vulnerabilidad ya que cualquier intromisión dentro de los archivos del nodo que permita la adquisición de las credenciales, permite total acceso a la `API-RESTfull` a terceros haciéndose pasar por el nodo en cuestión.

5.8.2.3 *Identificación de la aplicación RAZIEL mediante HTTP Basic Auth*

Para evitar usos no autorizados de la `API-RESTfull`, se implementa un método de identificación de la aplicación empleadora. De esta forma, solamente las aplicaciones desarrolladas autorizadas pueden hacer uso de la plataforma de comunicación. Para ello se emplea el mecanismo de autenticación `Basic Auth` de `HTTP` [223].

[223] RFC 7617 - The'Basic'HTTP Authentication Scheme

Cada versión del software RAZIEL incorpora en su código fuente de forma `hardcode` unas credenciales determinadas que firman la versión concreta del software desarrollado. En cada comunicación, además de las credenciales del sistema enviadas mediante variables `POST`, se envían dichas credenciales empleando los campos `HTTP` estándar para tal fin. Las transmisiones que se realicen sin dichas credenciales válidas, son descartadas por el servidor, incluso aunque las variables de identificación del sistema sean correctas.

Al encontrarse las credenciales que firman la aplicación dentro del propio código fuente, estas credenciales no son accesibles aunque un tercero tenga acceso a los archivos del nodo. De igual manera, se puede bloquear el acceso a la `API-RESTfull` a versiones antiguas del software desarrollado, sencillamente revocando las credenciales de dicha versión de la lista de versiones autorizadas.

5.8.3 *Descripción ampliada de las funciones de la API*

En la Sección 5.6.5.6 se han presentado las funciones del software RAZIEL que hacen uso de la API REST. En la Tabla 5.16 se presentan la correspondencia entre las funciones del software y las funciones provistas por la API. Una definición detalla de todas y cada una de estas funciones se recoge en el Anexo A.4.

Tabla 5.16
Correspondencia entre las funciones del módulo ServerModule del software RAZIEL y las funciones de la API REST

Función en el nodo	Función en la API	Descripción
`testConnection`	`/connection/test`	Comprueba conexión con el servidor
`getNodes`	`/nodes`	Devuelve la lista de nodos asociados al usuario
`startsession`	`/session/start`	Inicia la sesión del nodo indicado
`endsession`	`/session/end`	Finaliza la sesión del nodo indicado
`createSteps`	`/steps`	Envía los pasos a notificar al servidor
`getApp`	`/version/app`	Descarga la versión establecida del software
`getCurrentVersion`	`/version/current`	Obtiene el número de versión del software establecida
`getScript`	`/version/script`	Descarga, de existir, el script a ejecutar de forma remota
`sendStatus`	`/status`	Envía mensajes de estado al servidor

5.9 ALMACENAMIENTO LOCAL

En esta sección se describe el sistema de almacenamiento local del servidor de cómputo, encargado de guardar los *pasos* transmisitidos por los nodos de monitorización y la información relativa al funcionamiento de estos. Se presenta el motor de base de datos empleado para tal fin y las cuestiones de diseño que justifican su empleo. Se recogen las configuraciones más relevantes al ámbito de la tesis realizadas en el motor, el esquema relacional diseñado para el almacenamiento de la información así como las optimizaciones que garantizan una alta disponibilidad y eficiencia del acceso a la información.

Esta sección se complementa con la Sección 5.13 donde se presenta la parte del almacenamiento en la nube. Si bien el servidor de almacenamiento local trata con la información en crudo de los pasos recibidos por cada nodo de monitorización, la información almacenada en la nube es información procesada y resumida, resultado de la ejecución de los métodos analíticos (Sección 5.11) y predictivos (Sección **??**).

5.9.1 *Descripción de la información almacenada*

Cómo se ha descrito en la Sección 5.1, la unidad mínima de información notificada al servidor es el paso de un dispositivo. Esta información es transmitida por el software de monitorización `RAZIEL` (Sección 5.6) por medio de la plataforma de comunicación (Sección 5.8).

El servidor almacena todos los pasos recibidos por la plataforma de comunicación de cada uno de los sensores del nodo de monitorización. Cómo se ha definido anteriormente en las secciones indicadas, el paso de un dispositivo contiene el identificador del dispositivo (Secciones 4.3.5 y 4.2.2), del sensor del nodo que lo ha detectado (Sección 5.6.3.1), la ventana de tiempo de la primera y última detección, así cómo la información adicional dependiente del sensor empleado (Secciones 5.1.2 y 5.1.2).

Estos elementos de información se recogen en la Figura 5.59.

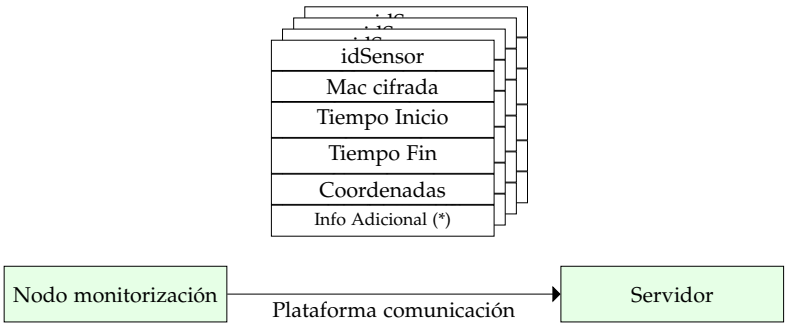


Figura 5.59
Información suministrada por el nodo al servidor.
(*) La información adicional varía en función de la tecnología inalámbrica empleada y se encuentran descritas en la Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi.

La información adicional es descrita en la Sección 5.1.2 para Bluetooth y Sección 5.1.2 para WiFi, pero en resumen, en Bluetooth provee información adicional sobre el dispositivo detectado y en el WiFi provee información sobre las calidad o intensidad de las detecciones que del paso.

Adicionalmente, y por los motivos descritos en la Sección 5.6.3.2, los pasos enviado por el nodo de monitorización incorporan también las coordenadas latitud y longitud donde se realizó la detección. Sin embargo, en la versión prototipo del sistema de almacenamiento desarrollado, esta información aunque es recibida por el servidor no es almacenada, debido a que en el estado de desarrollo actual, no se ha implementado que los nodos puedan desplazarse. Sin embargo, el sistema de monitorización se encuentra preparado, para el supuesto en el que los nodos pudiesen moverse, el servidor estaría recibiendo ya dicha información.

Hay por tanto cuatro elementos de información principales en el almacenamiento: los nodos, sensores, pasos y dispositivos. Así como otra información secundaria relativa al control de versiones, mantenimiento y autenticación.

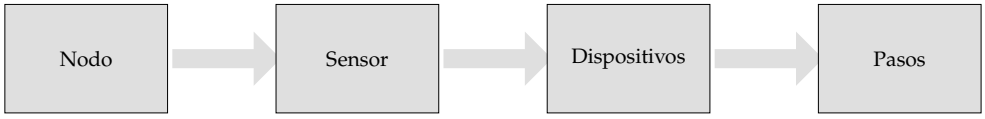


Figura 5.60
Información a ser almacenada en el servidor, relativa a los nodos, sensores, pasos y dispositivos

Los elementos de información principales están muy relacionados entre ellos. Los nodos disponen de varios sensores, los cuales detectan varias veces a un mismo dispositivo en distintos instantes de tiempo. Estas detecciones de cada dispositivo por un sensor, constituyen un paso de dicho dispositivo por el nodo. Además, un mismo dispositivo puede pasar por los sensores de igual naturaleza de varios nodos, por lo tanto los pasos de un mismo dispositivo en una ventana cercana de tiempo en varios nodos distintos se encuentran también fuertemente relacionados.

Todas estas relaciones entre las unididades atómicas de información con la que tiene que trabajar el sistema, son constituyentes de las restricciones y requerimientos del motor empleado para el almacenamiento de los datos.

5.9.2 *Elección del motor de base de datos.*

Debido a las restricciones y relaciones presentadas en las sección anterior, se opta por un motor de base de datos relacional o SQL, más concretamente MySQL 5.7 [243]. Aunque se ha contemplado la viabilidad de motores NoSQL, estos han sido descartados para el almacenamiento local debido al fuerte caracter relacional de los datos a tratar. Sin embargo, estos motores han sido considerados y empleados para el almacenamiento procesado en la nube, que se describe en la Sección 5.13.

[243] High Performance MySQL: Optimization, Backups, and Replication

No es potestad de esta tesis debatir sobre los aspectos distintivos de las bases de datos SQL contra las NoSQL. Sin embargo, la falta de estandarización en las bases de datos NoSQL y la mayor dificultad para la gestión de consultas o `queries` elaboradas, decantan el empleo para el almacenamiento de los pasos del prototipo de base de datos SQL. El tratamiento de relaciones entre unidades de información no es nativo en las bases de datos NoSQL, por lo que no existen mecanismo eficientes para unir la información entre distintas entidades. Además, dada que la estructura de datos empleada está bien definida y los datos van a ser procesados de forma intenvisa, la flexibilidad de los modelos que ofrece NoSQL no es aprovechada.

Finalmente, los motores NoSQL son mucho más escalables que los motores SQL. Los motores SQL escalan de forma vertical, es decir, mejoran su eficiencia a medida que mejora las prestaciones del servidor. Los modelos NoSQL, en cambio, escalan de forma horizontal, lo que implica que se puede mejorar la eficiencia añadiendo más servidores de cómputo. Sin embargo, concretamente MySQL permite la escalabiidad horizontal mediante el empleo de `NDB Cluster`[67], que permite establacer particiones (`shards` o `partitions`) de los datos.

Si bien, en el prototipo no se han empleado varios nodos de almacenamiento, la configuración actual se encuentra preparada para, fácilmente en el futuro, añadir cuantos nodos de almacenamiento adicionales se deseen al sistema.

Por tanto, las principales ventajas del empleo de motores NoSQL, no aportan, para el problema del almacenamiento local de los pasos, ninguna funcionalidad de peso para decantar su empleo, frente a la estabilidad y madurez que ofrecen los sistemas SQL, en este caso, MySQL.

67 ↑`https://dev.mysql.com/doc/refman/5.7/en/mysql-cluster.html`

5.9.3 *Optimizaciones al motor de base de datos.*

A pensar de considerar que MySQL es la elección más adecuada para los requerimientos establecidos, se hace necesario una serie de configuraciónes para optimizar el comportamiento y eficiencia del motor de base datos para el objetivo propuesto.

En concreto se presentan brevemente tanto aquellas configuraciones intimamente relacionadas con el ámbito de esta tesis o aquellas que no suelen ser empleadas frecuentemente, pero para el prototipo resultan relevantes.

5.9.3.1 *Deshabilitado de la `Query-Caché`*

La caché de consultas o `query-cache` es un mecanismo nativo de MySQL que se encarga de almacenar el texto de las consulta realizas junto con el resultado que se envió al cliente. Al almacenar tanto el texto de las consultas como el resultado, este mecanismo permite responder de forma rápida a una petición de una misma consulta antes de expire el tempo de caché.

Este tipo de mecanismos de caché son muy empleados y útiles en servidores WEB y demás entornos en los que la probabilidad de volver a solicitar unos datos anteriormente solicitados es muy alta. Por ejemplo, una página web que ofrezca resultados obtenidos de una única consulta en cada una de sus visualizaciones. También resultan muy útiles en entornos donde las tablas no sufren muchos cambios a lo largo del tiempo.

Debido a que MySQL es comúnmente empleada en el desarrollo web y a la eficiencia en resolución de consultas sucesivas, es el mecanismo de `query-cache` se encuentra habilitado por defecto. Sin embargo, el sistema de almacenamiento local de los pasos empleado en el prototipo es de distinta naturaleza. Los datos son creados prácticamente en tiempo real, y debido a que los almacenados son procesador por el propio sistema y almacenados en la nube, no se suelen recibir multiples peticiones de una misma consulta concreta.

Aunque este mecanismo ofrece una alta eficiencia en entornos donde las consultas son reiterativas, en el sistema propuesto supone un cuello de botella a la paralelización de los acceso a la base de datos, al tener que consultarse la caché antes de procesar cualquier consulta recibida. De esta forma, aunque el motor de la base de datos reciba múltiples peticiones en paralelo, un único elemento (la `query-cache`) debe comprobar de forma secuencial que la petición realizada no haya sido resulta anteriormente.

Debido al empleo no reiterativo del sistema de almacenamiento local, se opta por dehabilitar la caché de consultas. Para ello es necesario cambiar la variable Global `query_cache_type` al valor `OFF` de la configuración del motor.

5.9.3.2 *Aumentado cuotas de tiempo procesamiento*

Para evitar tipos de ataques o bloquear demasiado tiempo las tablas de las bases de datos, MySQL incluye por defecto un tiempo máximo de procesamiento de las consultas realizadas. Si se excede ese tiempo máximo de ejecución, la ejecución de la consulta es cancelada por el motor.

Este comportamiento es determinado por la variable de la configuración global `max_execution_time`. En el motor empleado, se establece al valor `0` para indicar que el motor no tiene que cortar, por su propia decisión, ninguna consulta a pesar de la duración de esta.

Esta configuración, permite realizar análisis computacionalmente más complejos, implicando una gran cantidad de tiempo de procesamiento de los datos.

5.9.3.3 *Aumentado tamaño de tablas temporales y en memoria*

MySQL emplea tablas temporales para las consultas que requieren o generan tablas intermedias, como por ejemplo como resultado de emplear cualquier mecanismo de unión u ordenación sobre una o varias tablas.

Por defecto, cualquier tabla temporal es generada en memoria hasta que supera el tamaño indicado por la variable `max_heap_table_size`, momento en el que se emplea una tabla en disco. Si la tabla temporal supera el tamaño indicado por la variable `tmp_table_size`, la consulta que ha generado dicha tabla es cancelada en el mejor de los casos, o emplea únicamente los datos que han cabido en la tabla temporal en el peor de los cosas. Por defecto, MySQL limita el tamaño de las tablas temporales a $16MB$.

Debido a que el mecanismo de tablas temporales es muy empleado en el procesamiento que se describe en la Sección 5.10, a que el empleo de tablas temporales en memoria es más eficiente que el empleo de tablas en disco y a que el tamaño de $16MB$ por defecto resulta insuficiente para muchas tareas con volumen considerable de datos, se amplia el tamaño de estas tablas a$512MB$ las tablas en memoria y a $2GB$ las tablas temporales. Esta configuración se realiza mediante las instrucciones del código 5.61.

```
1 SET GLOBAL tmp_table_size = 1024 * 1024 * 2048;
2 SET GLOBAL max_heap_table_size = 1024 * 1024 * 512;
```

Figura 5.61
Definición del tamaño de las tablas temporales empleadas por el motor de base de datos.

Estas cantidades han resultado suficientes para los algoritmos empleados en los experimentos realizado, pero si fuese requerido, podrían ser ampliados fácilmente para preparar el escenario para la ejecución de nuevos experimentos con mayor volumen de datos.

5.9.3.4 *Habilitado de `performance schema` para la monitorización del rendimiento*

`Performance schema`[68] es una funcionalidad de MySQL que permite monitorizar el servidor a bajo nivel, centrándose en los eventos a los que el servidor responde. Sin entrar en demasiado detalle, provee un esquema con tablas almacenadas en memoria[69] que almacena contadores sobre la utilización de cada uno de los recursos, como tablas o índices. Por medio de este esquema adicional, por ejemplo, puede medirse la eficiencia general del servidor, la eficiencia de empleo de los índices desarrollados, localizar cuellos de botella o funcionalidades críticas.

Un ejemplo de informe generado mediante `Performance schema` se puede ver en la figura Figura 5.62, donde se recogen las estadísticas de uso de las tablas del servidor[70].

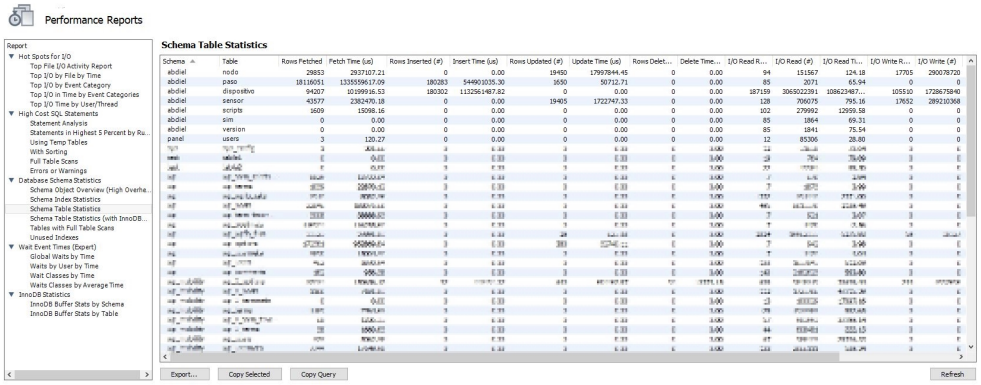


Figura 5.62
Informe generado mediante la información almacenada en `Performance schema`.

El habilitado del `Performance schema` provee una valiosa y poderosa herramienta para la depuración y optimización del servidor de almacenamietno local. Para habilitarlo, es necesario cambiar la variable Global `performance_schema` al valor `ON` de la configuración del motor.

5.9.3.5 *Gestión de datos geográficos*

Siguiendo la especificación `OGC`[71], `MySQL` a partir de la versión 5.7 incluye una extensión para trabajar con datos geométricos lo que le permite generar, almacenar y analizar datos geográficos. En el ámbito de la tesis, esto permite relacionar y geoposicionar los nodos y realizar consultas empleando operadores geográficos, como por ejemplo, localizar los nodos más cercanos a un nodo determinado, o calcular las distancias en línea recta entre los nodos. En la Sección 5.10.3 se presentan algunos procedimientos que hacen uso de estas funciones geográficas.

68 ↑`https://dev.mysql.com/doc/mysql-perfschema-excerpt/5.7/en/`
69 ↑De esta forma, el impacto en la eficiencia del servidor es mínima.
70 ↑Los datos mostrados corresponden a 15 días de ejecución del servidor
71 ↑Open Geospatial Consortium `http://www.opengeospatial.org/standards/sfs`

5.9.4 *Elección del motor de almacenamiento*

Una vez elegido el motor de base de datos a emplear, la siguiente cuestión de diseño es la del motor de almacenaiento o `storage-engine`. El motor de almacenamiento es el mecanismo del servidor de base de datos encargado del almacenamiento de las tablas de la base de datos. Anteriormente, MySQL emplea por defecto `MyISAM`, pero desde la versión 5.5 emplea `InnoDB`, aunque muchos otros `storage-engine` son empleables[72]. Debido a que la elección del `storage-engine` implica distinta eficiencia y funcionalidades, es un aspecto que debe de ser considerado.

Se elige `InnoDB` como `storage-engine` debido a que se trata de un motor de almacenamiento transaccional `ACID` [112] o `ACID Compliant`. `ACID` es el acrónimo de `Atomicity`, `Consistency`, `Isolation` y `Durability`; o en castellano: Atomicidad, Consistencia, Aislamiento y Durabilidad. Sin entrar en demasiado detalle, que escapa del ámbito de esta tesis, al ser `InnoDB ACID Compliant` se garantiza la integridad de las tablas y su información. `InnoDB` además provee el mecanismo de claves foráneas y distintos niveles de bloqueos. Finalmente, `InnoDB` es el único `storage-engine` de MySQL que ofrece el mecanismo de particionado, cuyo empleo se describe en la Sección 5.9.6.3.

5.9.5 *Esquema de base de datos*

En esta sección se presenta el esquema de base de datos empleada en el sistema de almacenamietno local del prototipo de sistema de monitorización. Se presenta cada tabla empleada para almacenar cada unidad de información presentada en la Sección 5.9.1. En la Figura 5.63 se muestra a modo de resumen las tablas y sus relaciones. Además, se recogen las vistas implementadas sobre las tablas físicas.

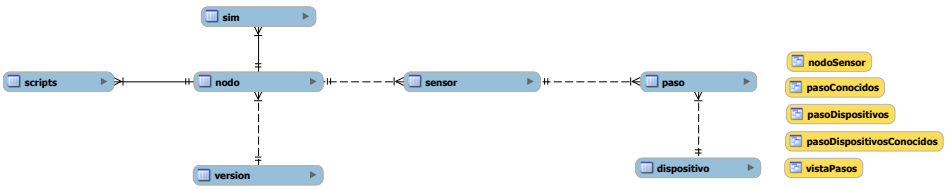


Figura 5.63
Relaciones entre las tablas del almacenamiento local. En azul las tablas físicas, en amarillo, las vistas creadas sobre las distintas tablas, que funcionan como tablas lógicas o virtuales. Gráfico generado mediante `MySQL Workbench`: `https://www.mysql.com/products/workbench/`.

A continuación, se recogen las definiciones de las distintas tablas, con las consideraciones de diseño más relevantes para el ámbito de la tesis[73]..

72 ↑`https://dev.mysql.com/doc/refman/5.7/en/storage-engines.html`

73 ↑Aunque en esta sección se indicarán los índices implementados en cada tabla, este tema será abordado en mayor profundidad en la Sección 5.9.6 donde se explicarán la naturaleza y funcionamiento de los índices establecidos para cada tabla y su impacto en la resolución de consultas para el análisis que se presenta en la Sección 5.11

5.9.5.1 *Tabla nodo*

Esta tabla es la encargada de almacenar la información de los nodos de monitorización del sistema. Consta de los campos recogidos en la Tabla 5.17.

Tabla 5.17
Tabla `nodo` que almacena la información del nodo de monitorización.

`nodo`		
`COLUMNAS`		
`idNodo`	`(int(10) unsigned)`	Identificador del nodo. Ver Sección 5.6.3.1.
`latitud`	`(double)`	Empleado para la localización geográfica del nodo. Ver Sección 5.6.3.2
`longitud`	`(double)`	Empleado para la localización geográfica del nodo. Ver Sección 5.6.3.2
`nombre`	`(varchar(200))`	Nombre largo descriptivo del nodo y su proyecto. Por ejemplo `DGT-PETRA-MOMOFES-A-44pk118.3-creciente`
`etiqueta`	`(varchar(45))`	Nombre más coloquial del nodo, de uso interno.
`poligono`	`(varchar(1400))`	Polígono `kml` de representación del nodo.
`inforAdicional`	`(varchar(140))`	Información adicional sobre el nodo. Averías, emplazamiento, modificaciones.
`ultimaIP`	`(int(10) unsigned)`	Última IP desde la que ha comunicado el nodo.
`ultimaConexion`	`(timestamp)`	Marca de tiempo de la última comunicación del nodo con el servidor.
`intervaloActual`	`(int(10) unsigned)`	Intervalo de tiempo de sincronización del envío de datos con el servidor. Ver Sección 5.6.3.
`usuarioWS`	`(varchar(40))`	Credenciales de la plataforma de comunicación. Ver Sección 5.8.
`passwordWS`	`(binary(16))`	Credenciales de la plataforma de comunicación. Ver Sección 5.8.
`usuarioSSH`	`(varchar(40))`	Credenciales de acceso remoto por SSH. Ver Sección 5.6.5.10.
`passwordSSH`	`(binary(16))`	Credenciales de acceso remoto por SSH. Ver Sección 5.6.5.10.
`modoDebug`	`(tinyint(1))`	Bandera que indica si el nodo el servidor debe descartar los pasos recibidos.
`net_ip`	`(int(10) unsigned)`	Credenciales de red de la conexión fija en la infraestructura de terceros.
`net_mask`	`(int(10) unsigned)`	Credenciales de red de la conexión fija en la infraestructura de terceros.
`net_gateway`	`(int(10) unsigned)`	Credenciales de red de la conexión fija en la infraestructura de terceros.
`idVersion`	`(varchar(200))`	Número de versión del software de monitorización que el servidor establace que tiene que ejecutar el nodo de monitorización. Ver Sección 5.6.5.9
`tarjetaSIM`	`(varchar(20))`	Número de referencia de la tarjeta SIM emplazada en el módem 3G del nodo de monitorización.
`ÍNDICES`		
`PRIMARY`	`UNIQUE-BTREE`	(idNodo)

Sobre ella se implementa un único índice, por defecto, en la columna idNodo.

Para facilitar la creación de nodos en el sistema, se implementan varias funciones `crearNodo`, una de ellas descrita en el Código 5.52 que codifica las contraseñas de forma automática.

Código 5.52
Función crearNodo

```
CREATE DEFINER=`root`@`localhost` FUNCTION `crearNodo`(_idNodo INT(10),_nombreNodo
↪ VARCHAR(200),_latitud FLOAT,_longitud FLOAT,_usuarioWS
↪ VARCHAR(40),_passwordWSTextoPlano VARCHAR(40)) RETURNS int(10)

BEGIN

INSERT INTO `abdiel`.`nodo` (`idNodo`,`latitud`, `longitud`, `nombre`, `usuarioWS`,
↪ `passwordWS`,`modoDebug`)
VALUES (_idNodo, _latitud, _longitud, _nombreNodo, _usuarioWS,
↪ unhex(md5(_passwordWSTextoPlano)),1);

RETURN LAST_INSERT_ID();
END
```

De igual manera, como las credenciales de red se almacenan como dígitos, para facilitar la introducción de las credenciales de red, se implementa la función `setNodoNetwork` recogida en el Código 5.53.

Código 5.53
Función setNodoNetwork

```
CREATE DEFINER=`root`@`localhost` FUNCTION `setNodoNetwork`(_idNodo INT(10), _ip
↪ VARCHAR(200), _mask VARCHAR(200), _gateway VARCHAR(200)) RETURNS int(11)
BEGIN
        UPDATE `abdiel`.`nodo` SET `net_ip`=inet_aton(_ip) ,
        ↪ `net_mask`=inet_aton(_mask), `net_gateway`=inet_aton(_gateway) WHERE
        ↪ `idNodo`=_idNodo;
    RETURN ROW_COUNT();
END
```

Estas funciones permite crear desde el propio intérprete de MySQL los elementos funcionales garantizando la consistencia de los datos, funcionalidad ha resultado bastante útil antes del desarrollo del panel de control web que se describe brevemente en la Sección 5.14.3.

5.9.5.2 *Tabla sensor*

Esta tabla mantiene la información sobre los distintos sensores que emplea cada nodo de monitorización. Los campos de la tabla se recogen en la Tabla 5.18.

Tabla 5.18
Tabla sensor que almacena la información de los sensonres del nodo de monitorización.

nodo		
COLUMNAS		
idSensor	(int(10) unsigned)	Identificador del sensor. Ver Sección 5.6.3.1.
idNodo	(int(10) unsigned)	Identificador del nodo. Ver Sección 5.6.3.1.
tipo	(enum)	Tipo de sensor. En el prototipo, sólo existen dos tipos de sensores: Bluetooth y WiFi.
ultimoEnvioFecha	(timestamp)	Marca de tiempo del último envío de pasos detectados por el sensor.
ultimoEnvioPaso	(int(11))	Número de pasos enviados en el último envío.
ÍNDICES		
PRIMARY	UNIQUE	(idSensor)
sensor_nodo	BTREE	(idNodo)

En el prototipo únicamente se han desarrollado dos tipos de sensores (BLUETOOH y WiFi), pero añadir más tipos sensores a cada nodo sería trivial actualmente en el sistema de almacenamiento local, requiriendo únicamente definir un nuevo tipo de naturaleza de sensor[74].

Aunque no ha sido implementado en los nodos de monitorización, el sistema de almacenamiento permite que un nodo disponga de varios sensores de igual naturaleza, definiendo únicamente tantos sensores como se desee en esta tabla. Para facilitar la creación de los Sensores habituales de un nodo, se desarrolla la función `crearSensor` que se presenta en el Código 5.54.

Código 5.54
Función crearSensor

```
1 CREATE DEFINER=`root`@`localhost` FUNCTION `crearSensor`( _idNodo INT(10), _tipo
  ↪  ENUM("Bluetooth","Wifi")) RETURNS int(10)
2 BEGIN
3 INSERT INTO `abdiel`.`sensor` (`idSensor`, `tipo`, `idNodo`)
4 VALUES (_idNodo+_tipo, _tipo, _idNodo);
5
6 RETURN LAST_INSERT_ID();
```

Los campos `ultimoEnvioFecha` y `ultimoEnvioPasso` sirven de mecanismo de control de errores y son empleados, entre otros, por el panel de control web (Sección 5.14.3).

74 ↑Aunque puede ser discutida la decisión de emplear tipos ENUM, su uso está más que justificado en entornos donde la variable a almacenar no es dinámica, garantizando mayor eficiencia, legibilidad y flexibilidad sacrificando ligeramente la creación de nuevos tipos de naturalezas, que ha de ser realizada moficando las definiciones de las columnas. Sin embargo, en el empleo de tipos ENUM ofrece una gran eficiencia, y más en una tarea tan empleada como la definición de tipos de dispositivos y sensores.

5.9.5.3 *Tabla dispositivo*

Dado que es esperable que un dispositivo sea detectado en múltiples ocasiones, se implementa una tabla que contenga la información común del dispositivo a todos los pasos que se haya detectado del mismo. Esta tabla es la consta de los siguientes campos, cuya relación puede observarse en la Figura 5.63:

Tabla 5.19
Tabla `sensor` que almacena la información de los dispositivos detectados en una o más ocasiones.

`dispositivo`		
`COLUMNAS`		
`idDispositivo`	`(binary(20) unsigned)`	Identificador del dispositivo. Ver Secciones 4.2.2 y **??**.
`majorDevClass`	`(char(10))`	Identificador tipo principal de dispositivo. Ver Sección 4.2.3.5.
`minorDevClass`	`(char(5))`	Identificador suptipo de dispositivo. Ver Sección 4.2.3.5.
`serviceDevClass`	`(char(10))`	Identificado los servicios ofrecidos. Ver Sección 4.2.3.5.
`fabricante`	`(varchar(100))`	Nombre del fabricante del dispositivo. Ver Sección 5.1.2.
`naturaleza`	`(enum)`	Naturaleza del dispositivo. En el prototipo, sólo existen dos tipos de naturaleza: Bluetooth y WiFi.
`etiqueta`	`(varchar(40))`	Etiqueta del dispositivo, para tareas de depuración identificar de forma simple un dispositivo de pruebas.
`blacklist`	`(int(11))`	Bandera para omitir el dispositivo en los recuentos de pasos de dispositivos.
`ÍNDICES`		
`PRIMARY`	`UNIQUE`	(idDispositivo)

La naturaleza de los campos `majordeviceclass`,`minordeviceclass` y `serviceclass` se describe en las Secciones 4.2.3.5 y 5.1.2 y son empleados por los dispositivos Bluetooth para conocer la naturaleza de los dispositivos cercanos. Como son pertenecientes a la naturaleza del dispositivo, son comunes a todas las detecciones y pasos del mismo.

El campo etiqueta es empleado para identificar y localizar de forma natural los dispositivos empleados en la fase de depuración del prototipo, asignando nombres comunes a dispositivos concretos. De esta manera, es relativamente sencillo reconocer un dispositivo empleado para pruebas sin tener que memorizar el identificador del dispositivo.

Debido a la posible controversia del sistema de monitorización, es posible excluir de los estudios a dispositivos concretos activando la bandera `blacklist` del dispostivo. Aunque el sistema seguirá recibiendo los pasos realizados por dicho dispositivo, no será contabilizado en los análisis que se presentan en la Sección 5.11. Esta bandera es también empleada para descartar dispositivos de infraestructura o de individuos que no sea necesario monitorizar expresamente. Por ejemplo, si el sistema de monitorización se implantase en un negocio, sería posible excluir de los análisis a los empleados de dicho negocio.

5.9.5.4 *Tabla paso*

Los pasos de dispositivos son la información primordial a almacenar en el sistema de almacenamiento. Esta tabla es crítica dentro del sistema y ha sido fruto de numerosos estudios y optimizaciones que se recogen en la Sección 5.9.6. La estructura de la tabla se recoge en la Tabla **??**:

Tabla 5.20
Tabla paso que almacena la información de los pasos de dispositivos.

paso		
COLUMNAS		
idSensor	(int(10) unsigned)	Identificador del sensor. Ver Sección 5.6.3.1.
idDispositivo	(binary(20) unsigned)	Identificador del dispositivo. Ver Secciones 4.2.2 y **??**.
tinicio	(timestampt(3))	Marca de tiempo del instante inicial en el que ha sido detectado el dispositivo.
tfin	(timestampt(3))	Marca de tiempo del último instante de tiempo en el que ha sido detectado el dispositivo.
tdb	(timestampt(3))	Marca de tiempo del instante de tiempo en el que el paso ha sido insertado en la base de datos.
sinitial	(tinyint(1))	El indicador de fuerza de la señal de la primera detección del dispositivo. Ver Sección 5.1.2.
sfinal	(tinyint(1))	El indicador de fuerza de la señal de la última detección del dispositivo. Ver Sección 5.1.2.
obsoleto	(bit(1))	Bandera que indica si el paso ha quedado obsoleto. Ver Sección 5.6.1.3.
ÍNDICES		
noDuplicados	UNIQUE	(idSensor,idDispositivo,tinicio)
ventanaTemporal	BTREE	(tinicio)
busquedaDispositivo	BTREE	(idDispositivo)
idSensor	BTREE	(idSensor)

Como los campos de información adicional WiFi (Sección 5.1.2) son relativos a las detecciones del dispositivo, son almacenados en la tabla de los pasos, pues varian entre varias detecciones de un mismo dispositivo.

Para facilitar tareas de depuración, se añade también una marca de tiempo `tdb` con el instante en el que el paso ha sido insertado en la base de datos. Debido a que el nodo de monitorización dispone de varias fuentes para determinar su fecha y hora (Sección 5.5.4) es factible que el nodo tenga desajustes horarios, por ejemplo, ante fallos eléctricos o de red, que puede ser detectable ante grandes discrepaciancias entre las fecha de detección e inserción.

La bandera `obsoleto` es empleado en el sistema de monitorización HASTY presentada en la Sección 5.6.1.3.2, e indica si la información relativa al cierre de ventana de tiempo es certera. Si el dispositivo no está marcado como obsoleto, es probable que aún se encuentre en las inmediaciones del nodo, y que el tiempo relativo a `tfin` se vea actualizado por mediación del nodo de monitorización en el futuro.

Cómo se puede observar, no se dispone de una única clave primaria para esta tabla, sino que cada tupla es representada de forma inequívoca por la

convicación del instante de tiempo de primera detección y los identificadores del dispositivo y el sensor. Esto garantiza que no puede aparecer dos veces detectado el mismo dispositivo en el mismo instante de tiempo en el mismo sensor.

Cada uno de los elementos que componen dicha clave primaria, disponen de un índice en la tabla para su rápida localización. El tipo de tecnología empleada en estos índices, su eficacia y relevancia en el ámbito de la tesis es abordado en la Sección 5.9.6.

5.9.5.5 *Tablas auxiliares: version, scripts y sim*

Estas tablas son menos relevantes en el sistema de monitorización para el ámbito de la tesis, por lo que no requieren una descripción tan exhaustivas como las anteriores.

La tabla `version` contiene la información sobre el número de versión referenciado en la tabla `nodo`, el empaquetado `JAR` de dicha versión, así como las notas o consideraciones de dicha versión. Es empleado principalmente por el software de monitorización del nodo para actualizar las versiones en ejecución (Sección 5.6.5.9).

La tabla `scripts` mantiene el listado de scripts de mantenimiento que tiene que ejecutar cada nodo concreto del sistema. Es empleado por el sistema de ejecución remota del software de monitorización (Sección 5.6.5.10).

La tabla `sim` contiene la información relativa a la tarjeta sim y la conexión 3G que hace uso el nodo de monitorización determinado por la relación entre esta tabla y la tabla `nodo`.

5.9.5.6 *Vistas auxiliares*

El mecanismo de vistas de MySQL permite disponer de tablas virtuales que pueden ser resultado de consultas complejas o de la unión de una o más tablas. Para evitar realizar constantemente las consultas que generan dichas tablas virtuales, las vistas pueden ser almacenadas en el gestor de base de datos y ser empleadas de igual manera que las tablas como origen en cualquier otra consulta realizada. Las vistas se enumeran en la Figura 5.63, y su por su poca relevancia con la temática de la tesis y su poco impacto, no requieren más consideración.

5.9.6 *Optimizaciones en la base de datos*

El esquema de base de datos presentado es bastante simple, con la tabla `paso` como principal elemento crítico, pues es el contedor donde serán almacenado la mayor cantidad de información generada por el sistema de monitorización. La optimización del empleo de esta tabla para las consultas relacionadas con la misma es una pieza crítica del sistema, pues es una tabla con un alto crecimiento esperable.

En este apartado se presentan las optimizaciones llevadas a a cabo sobre la estructura de base de datos presentada en la Sección 5.9.4. Estas optimizaciones suponen en muchos mecanismos avanzados de alto rendimiento o `High Performance` [243], que no suelen ser habilitados ni configurados por defecto, pero que suponen una mejora significativa en eficiencia.

5.9.6.1 *Optimizado de consultas con ventanas temporales*

El establecimiento de un marco temporal determiando se encuentra presente en la mayoría de las peticiones que se realizan a la tabla `paso`. Esto es debido, a que el sistema de monitorización, aunque almacene información histórica, los análisis y estudios se van a ceñir a un intervalo de tiempo concreto y acotado, por ejemplo, requiriendo los datos de la última semana, un día concreto o unas horas.

El principal mecanismmo empleado en las base de datos para optimizar las consultas, es el empleo de índices en aquellas columnas que determinan los campos de la búsqueda. O dicho de otra manera, los condiciones `WHERE` de las consultas. Un índice no es más que un puntero a una determina la tupla (o tuplas) de una tabla en base al valor (o valores) de la columna (o columnas) que han sido empleadas en la constitución del índice.

Por defecto, MySQL emplea en la mayoría de escenarios índices basado en funciones `HASH` para determinar la dirección a la que tiene que apuntar el puntero. Esta función permite limitar el rango de tuplas en las que el motor de base de datos tiene que localizar la tupla o tuplas requeridas. Y si el índice dispone de una alta cardinalidad, reducir al mínimo el tiempo requerido. Debido a que un índice basado en `HASH` determina mediante una función matemática la posición de la posición de la tupla deseada, en órdenes de eficiencia, podría decirse que operan en orden constante u $O(1)$. Es decir, el tiempo requerido para localizar cualquier tupla es constante en tiempo e independiente del tamaño del conjunto de búsqueda, siendo constituvo del tiempo de cómputo requerido para la ejecución de la función `HASH`.

La ventaja de los índices basados en funciones `HASH` es que son fácilmente computables, por lo que sus resultados no necesitan ser almacenados en contenedores auxiliares. Sin embargo, únicamente son útiles cuando las condiciones son exactas, es decir, cuando el criterio de búsqueda o columna require un único valor, que es el empleado en la función `HASH`. En la figura 5.64, se puede observar este mecanismo.

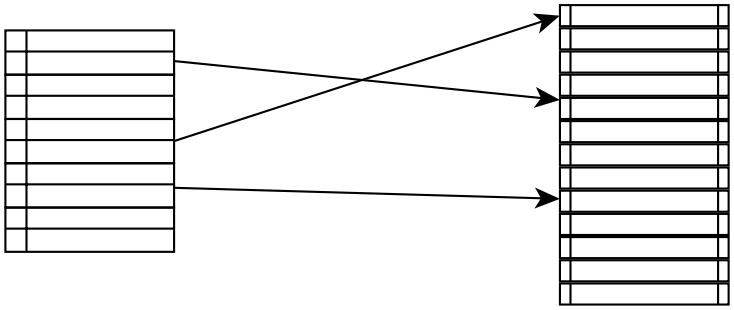


Figura 5.64
Funcionamiento de un índice HASH en la localización de tuplas

Sin embargo, para devolver tuplas que se encuentrne entre rangos de valores de la columna constituida por el índice, se necesita explorar todas las tuplas existentes en la tabla. Esto supone orden de eficiencia es $O(n)$, e implica que a medida que la tabla se haga más grande, el tiempo requerido para la resolución de consultas que requieran rango de valores, crecerá en igual medida al crecimiento de la tabla. Debido al empleo masivo de búsquedas mediante ventanas temporales que se prevee, la solución del empleo de índices HASH no es satisfactorio.

Se propone por tanto el empleo de índices basados en árboles binarios balanceados, Arboles-B o BTREE. Los índices almacenados por esta esta estructura están soportados de forma nativa por MySQL 5.7, pero no son empleados por defecto debido a que no son igual de efientes que los índices HASH en la resolución de consultas exactas, y debido a que requieren de estructuras de datos almacenadas en disco para su gestión. Esto implica, que el tamaño en disco de tabla contendrá tanto los datos almacenados por la tabla, como las estructuras de datos adicionales creadas por los índices BTREE empleados.

Un índice basado en BTREE resuelve la posición de un tupla concreta recorriendo los nodos de un árbol binario balanceado, constituido respecto al valor de la clave definida en el índice ordenada según sus valores. Este criterio de búsqueda se recoge en la Figura 5.65.

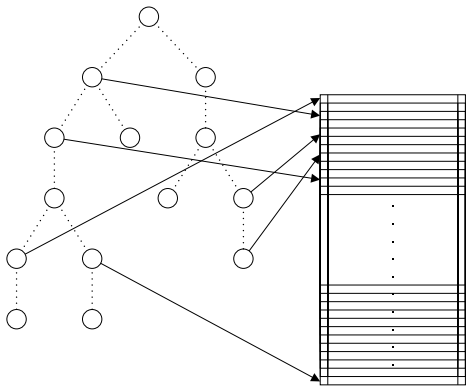


Figura 5.65
Funcionamiento de un índice basado en BTREE en la localización de tuplas

Para la localización de una tupla concreta en base a un valor de índice exacto, se recogen los nodos del árbol hasta localizar el nodo hijo que contiene el valor exacto de la clave buscada. Esto supone un órden de eficiencia de $O(logn)$, que resulta mucha más ineficiente que el empleo del índice `HASH`. Sin embargo, para recuperar las tuplas comprendidas entre el rango de valores consituido por dos valores concretos de la clave del índice, sólo se tienen que recorrer el árbol hasta localizar los nodos hijos que identifican las tuplas con dichos valores. Las tuplas con valores intermedios a los definidos en las búsqueda, son obtenibles realizando un recorrido enorden en el Árbol-B. Un ejemplo de este mecanismos de búsqueda se recoge en la Figura 5.66.

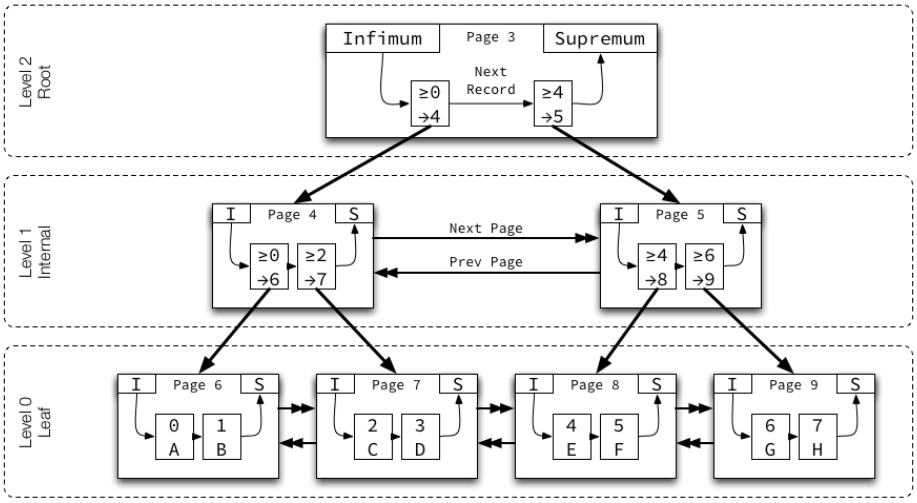


Figura 5.66
Organización de un índice basado en BTREE
Fuente: Jeremy Cole - B+Tree index structures in InnoDB [57]

El órden de eficiencia requerido para localizar las tuplas pertenecientes a un ventana de valores, empleado índices basados en `BTREE` es de orden $O(2 \times log\ n)$, como siempre, en el peor de los casos. Ya que el motor de base de datos, está preparado para realizar la búsqueda de ambos valores de forma simultánea en el árbol en las etapas iniciales del recorrido.

Estudio 5.9.1: Estudio de eficiencia de los índice BTREE

Para cuantificar el impacto de los índices basados en BTREE frente a los basados en HASH se crean varios subconjuntos de n pasos, alojados en diferentes tablas cuya única diferencia es la tipología de los índices empleados.

Se ejecuta en ambas tablas la consulta recogida en el código 5.55 que solicita los pasos recogidos en una ventana de tiempo determinada.

Código 5.55
Consulta a ejecutar en la experimentación de índices.
Dicha consulta solicita los dispositivos que han pasado entre dos fechas concretas, expresadas en fortado UNIX_TIMESTAMP con precisión de milisegundos

```
1  SELECT * FROM paso WHERE tinicio BETWEEN "2017/09/01 09:00:00" AND "2017/09/01
   ↪  10:00:00"
```

En la Tabla 5.21 se recogen los tiempos resolución de dicha consulta en ambos conjuntos de datos, descontado el tiempo de transferencia de los resultados.

Tabla 5.21
Tiempos en la comparativa entre índice por defecto HASH e índice basado en BTREE

TAMAÑO	HASH	BTREE
10000	1.93	0.01
100000	19.6	0.01
1000000	211.4	0.01
Tiempo (segundos)		

La diferencia de tiempo implicado en la resolución de las consultas es considerable. Sin embargo, hay un detalle adicional de los índices basados en BTREE y es que la tabla generada por la resolución de la consulta, en el caso del BTREE se encuentra ordenada según el campo `tinicio`, lo cual permite ahorrar ese paso en procesamientos o análisis posteriores.

El tiempo requerido en el índice BTREE se encuentra por debajo de la precisión máxima de las herramientas de medición empleadas. A fecha de escritura de esta tesis y con más de 28 millones de pasos registrados por el sistema de almacenamiento local, el tiempo de resolución sigue permaneciendo por debajo de 0.01 segundos. Un índice HASH de seguir la progresión lineal que deterina su eficiencia, necesitaría unos 6000 segundos para resolver cualquier consulta que implique el uso de una ventana temporal.

5.9.6.2 *Optimización de consultas de recurrencias*

Además del uso de las ventanas temporales en la mayoría de las consultas realizadas a nuestra base de datos, parte de nuestro cómputo residía en la búsqueda de recurrencias de un dispositivo capturado en distintos nodos. Supongamos así que un vehículo pasa por los nodos A y B en un corto periodo de tiempo y nos interesa conocer el tiempo que ha tardado en ir desde un nodo al otro. A esta ruta de un dispositivo entre dos nodos, lo denominamos trazas.

Las trazas resultan muy interesantes de estudiar, pues nos permiten obtener estadísticos interesantes sobre el flujo de movimiento de los vehículos dentro del mapa donde se hayan ubicados los nodos. Sim embargo, computacionalmente raelizar dicha búsqueda en nuestro conjunto de datos supone un orden de eficiencia $O(n^2)$ pues para cada tupla se debe buscar entre las otras tuplas si el `idDispositivo` es el mismo. Más adelante, en la Sección 5.10 se abordará un procedimiento SQL que optimiza este tipo de consultas para la búsqueda de todas las trazas.

Estudio 5.9.2: Eficiencia de consultas de recurrencias

Se ejecuta la siguiente consulta SQL para localizar las dos primeras trazas existentes en todo el conjunto de datos de pruebas, que recordemos es de 100 000 registros.

```
mysql> SELECT t1.idDispositivo, t1.idNodo as Origen,t1.tinicio as
↪  Origen_inicio, t1.tfin as Origen_fin, t1.tfin-t1.tinicio as Origen_dif,
↪  t2.idNodo as Destino, t2.tinicio as Destino_Inicio, t2.tfin as
↪  Destino_fin, t2.tfin-t2.tinicio as Destino_dif,
↪  from_unixtime(t1.tinicio/1000) as Origen_fecha,
↪  from_unixtime(t2.tinicio/1000) as Destino_fecha, t2.tinicio -
↪  t1.tinicio as Diferencia FROM paso as t1 INNER JOIN paso as t2 ON
↪  t1.idDispositivo = t2.idDispositivo and t1.idNodo <> t2.idNodo and
↪  t2.tinicio - t1.tinicio BETWEEN 0 AND 3600000 LIMIT 2;
```

Figura 5.67
Consulta a ejecutar en la experimentación de índices III. Dicha consulta busca en la base de datos dispositivos que hayan sido detectados por dos nodos distintos en una diferencia de tiempo de no más de una hora, devolviendo las 2 primeras ocurrencias.

Observar que en la consulta hacemos uso del mecanismo INNER JOIN para cotejar la tabla sobre si misma. Mecanismos de consulta como este supusieron un punto de inflexión a favor de los sistemas SQL frente a las alternativas NoSQL disponibles en el mercado, como se ha presentado en la Sección 5.9.2.

En un sistema sin optimizar se obteniene un tiempo de 58.69*s*, que resulta demasiado elevado como para poder acercar al tiempo real este tipo de procesamiento, tal y como se pretente.

Se hace necesario por tanto estudiar la manera de optimizar la ejecución de dichas consultas, considerando que toda optimización realizada mediante la creación de índices tiene un consumo de espacio en memoria para alojar y mantenerlo. Este tema es abordado en la Sección 5.9.6.4.

Se decide por tanto realizar un índice para el `idDispositivo` de forma que la recuperación de las tuplas con igual `idDispositivo` sea mucho más eficiente. De esta forma se le proporciona al gestor de base de datos de una estructura donde para cada idDispostivo puede localizar las tuplas que hacen referencia a este dispositivo en concreto.

Ejecutamos nuevamente la consulta reflejada en la Figura 5.67 habiendo realizado la optimización mediante un índice para el `idDispositivo`, consiguiendo un tiempo de $0.02s$. Con la configuración de este mecanismo, se ha conseguido casi una ganancia de 3000 magnitudes en la eficiencia de la ejecución de este tipo de consultas, acercado el cómputo a prácticamente un tiempo constante, lo que nos permite realizar este cómputo prácticamente en tiempo real.

5.9.6.3 *Particionado de datos*

Como se ha presentado en las Sección 5.9.5, la mayor parte del almacenamiento se prevee en la parte del almacenamiento de pasos de dispostiviso en la tabla *paso*. En `MySql` las tablas `InnoDB` tienen una limitación de tamaño máximo de $4GB$, lo cual puede resultar insuficiente si todos los nodos emplean la misma tabla.

Para salvar esta limitación, y ofrecer otras muchas ventajas, MySQL dispone del de `partitioning`, que permite particionar una tabla hasta en 1024 contenedores físicos, limitado cada contenedor a $4GB$. Sin embargo, esta acción no es realizada por defecto por el gestor, siendo necesario indicarlo expresamente durante el proceso de creación de la tabla. Además, desde la versión 5.1 de `MySQL` es posible indicarle al gestor de base de datos como se desea que se realice el particionado de datos, pudiendo emplear expresiones basadas en los valores de las tuplas para elegir elegir la partición (o contenedor físico) donde será almacenada la tupla, permitiendo elaborar mecanismos eficientes si se aprovecha el sistema de particionado para emplear el principio de localidad espacial en disco en las consultas más frecuentes.

En la Figura 5.68 se observa como los datos son almacenados en disco según van llegando las peticiones de inserción, lo cual es la situación por defecto del gestor.

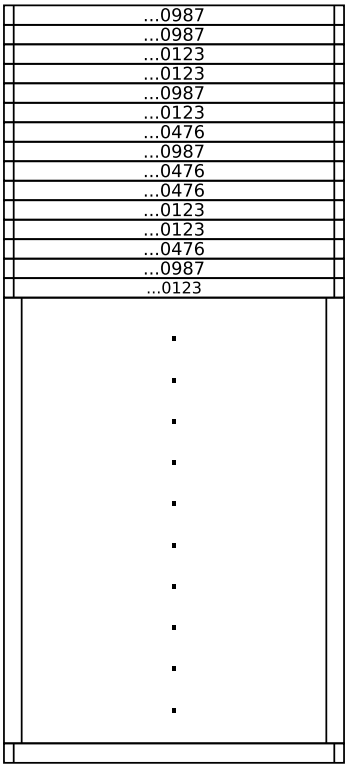


Figura 5.68
Almacenamiento de una tabla sin realizar particionado. Las tuplas se introducen de forma secuencial según se van insertando en el sistema. El dígito representa el identificador del nodo que ha

En la figura se ha suavizado para facilitar la legibilidad, pero se hace evidente observar que se realizarán rachas en las cuales se insertan tuplas consecutivas pertenezcan al mismo nodo/sensor, pues han sido introducidas y procesadas por lotes. Este comportamiento, hace que para leer los datos de una ventana temporal mayor que la ventana de inserción, sea necesario leer en el disco varias zonas no secuenciales, realizándose saltos a lo largo del contenedor de la tabla, resultando ineficiente.

Haciendo uso del mecanismo de particionado ofrecido por `MySQL`, se puede hacer corresponder un nodo o sensor concreto a cada partición de datos, de forma como se observa en la Figura 5.69. Esto permite que todas las tuplas (pasos) correspondiente al mismo nodo se almacenen en el mismo contenedor físico, y por tanto, es esperable que se encuentren en sectores consecutivos del disco duro. Además, aumentamos por 1024 el tamaño máximo de almacenamiento como se muestra en la sección **??**.

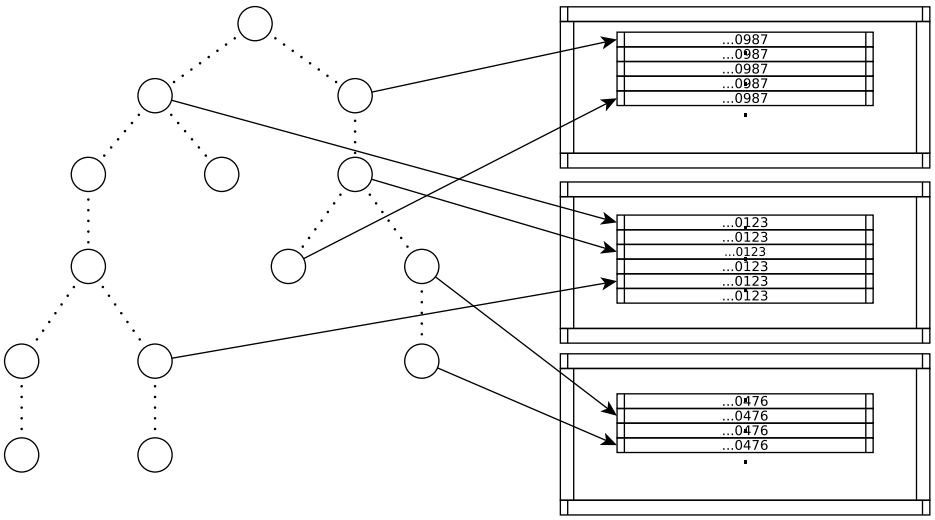


Figura 5.69
Funcionamiento de un tabla particionada en función del idNodo. Cada nodo/sensor dispone de un contenedor físico independiente, lo que permite aumentar la eficiencia de las lecturas secuenciales, y permitir un mejor mecanismo de bloqueo en la inserción.

Además, en consultas como la presentada en la Sección 5.9.6.2 se ven muy beneficiadas debido al sistema RAID1 con el que se ha dotado al servidor como se explica en la Sección 5.7.3.1 pues el gestor puede leer cada partición de un disco distinto, ya que a nivel físico son archivos distintos, y realizarlo por tanto de forma simultánea.

Por último, este es un mecanismo que se está potenciando y mejorando en las versiones de pruebas de MySQL debido a las ventajas que supone. Siendo mejorado y potenciado en cada versión, por ejemplo, en la versión de desarrollo 8.0 de `MySQL` se ha aumentado el número máximo de particiones `InnoDB` a 8192[75] frente a las 1024 de las versiones estables.

En el sistema de almacenamiento local se ha realizado un particionamiento de 96 particiones basado en el identificador del Nodo en la tabla *paso*, siendo ampliable según se requieran mayor cantidad de particiones.En la tabla *dispositivo* se ha empleado también el particionado para implementar un sistema de `blacklist`, que permite no considerar en el procesamiento los pasos de ciertos dispositivos designados.

75 ↑https://dev.mysql.com/doc/refman/8.0/en/partitioning-limitations.html

5.9.6.4 *Implicaciones de los Indices empleados*

En las tablas `InnoDB` que carecen de clave primaria establecida, como es el caso de la tabla *paso* (Tabla 5.9.5.4), MySQL determina que el primer índice `UNIQUE` será empleado como `cluster index`, o en caso de no existir ningún índice que satisfaga las necesidades, un identificador numérico autoincremental.

Los `cluster indexes` son almacenados por los índices secundarios junto con los valores de las columnas que constituyen el índice, a modo de identificador unívoco de cada tupla. Es por ello que el tamaño del `cluster index` es crítico, porque será replicado en todos y cada uno de los índices que se implementen.

Hay que lograr un equilibrio entre el número de índices implementados, y verificar si su eficiencia compensa el coste de almacenamiento que le implica. Además, hay que alcanzar una cardinalidad de posibles valores del índice, que permita el almacenamiento de una gran cantidad de tuplas.

En las tablas *nodo*, *sensor* y *dispositivo* la existencia de una clave primaria unívoca es bastante clara, contando cada tabla con un identificador (*idNodo*, *idSensor*, *idDispositivo*) con una correspondencia directa con la clave primaria.

En el caso de la tabla *paso*, la información almacenada no dispone de un identificador claro unívoco que pueda ser empleado como clave primaria, por lo que se tiene que constituir un `cluster index`. El gestor opta por tanto en el emplear el índice *noDuplicados*, constituido sobre las columnas `idSensor`, `idDispositivo` y `tinicio` (Tabla 5.9.5.4).

Como se ha comentado en las Secciones 5.9.6.2 y 5.9.6.1 donde se han comentado los mecanismos en los que se basan los índices, la tabla *paso* es la que realiza un uso intensivo de estos índices. Es por ello, que a continuación se presenta la sustentación en la que se basa la elección de este índice.

Estudio 5.9.3: Implicaciones del tamaño del `cluster index` de la tabla *paso*

Se ha tomado una partición de la tabla *paso*, cuyas cifras se presentan en la Tabla 5.22. Según esta tabla, excluyendo los índices, cada *paso* almacenado requiere 74*bytes*.

Tabla 5.22
Número de tuplas y tamaño de una partición de la tabla paso. Se presentan los tamaños tanto en Bytes como en MBytes, para una mayor interpretación.

NÚMERO DE TUPLAS	3 836 416	
TAMAÑO DATOS	287 145 984 bytes	273*MB*
TAMAÑO ÍNDICES	737 559 104 bytes	703*MB*

Aproximadamente un 70 % del espacio ocupado en disco por la partición se corresponde con los índices. En la Tabla 5.23 se presentan los tamaños de cada uno de los índices de la partición.

Tabla 5.23
De cada índice de una partición concreta, se presenta el tamaño ocupado en disco para su almacenamiento.

NODUPES	299 892 736 bytes	273*MB*	78 bytes por tupla
BUSQUEDADISPOSITIVOS	220 200 960 bytes	210*MB*	57 bytes por tupla
VENTANATEMPORAL	152 879 104 bytes	145*MB*	39 bytes por tupla

El tamaño en apariencia excesivo del campo índice es debido al tamaño del `cluster index` constituido por los campos recogidos en la Tabla 5.24, con un total de 12.5*bytes*.

Tabla 5.24
Composición del `cluster index` de la tabla paso, constituido con los campos del primer índice de componentes únicos ante la carencia de una `primary key` establecida.

COLUMNA	TIPO	TAMAÑO
idSensor	`INT(10)`	4 bytes
idDispositivo	`BINARY(20)`	2.5 bytes
tinicio	`TIMESTAMP(3)`	4 bytes + 2 bytes[a]

El requerir 12.5*bytes* para referenciar cada tupla en los índices secundarios, tiene un impacto bastante considerable en el espacio requerido en disco.

Se podría discutir sobre la viabilidad de emplear un índice artificial autoincremental usual basado en un número entero. En MySQL los enteros se definen en función del número de bytes empleado para almacenarlo, siendo los posibles valores 1,2,3,4 u 8[b].

Teniendo en cuenta que el número máximo de 1024 particiones y la limitación de 4*GB* por partición, el sistema de almacenamiento local podría trabajar con tablas de pasos de hasta 4*TB*[c]. Se necesita por tanto un índice primario con cardinalidad superior a $\frac{4TB}{74b}$ pasos distintos. Un índice basado en un `UNSIGNED INT` empleando 4*bytes* podría proporcionar únicamente un identificador primario a 7 % de los pasos totales almacenables en los 4*TB*, limitando el tamaño máximo de la tabla pasos a 286*MB*.

Se podría plantear la alternativa de emplear un clave primaria basada en un `UNSIGNED BITINT` empleando 8 bytes, que tendría capacidad más que suficiente para el almacenamiento, y supondría un ahorro respecto a los 12.5*bytes* empleados por el `cluster index`. En este caso, si se tendría cardinalidad más que suficiente para albergar identificadores para todos los pasos almacenables.

Sin embargo, empleamos un mecanismo de particionado descrito en la Sección 5.9.6.3, y este impone que la clave primaria o el `cluster index` contenga a las columnas que se empleen para definir el particionado, pues

de otra forma, no tiene manera de resolver mediante la clave primaria en que partición se encuentra.

Dado que en la tabla *paso* empleamos para el particionado el idSensor, a los 8*bytes* se le tienen que añadir los 4*bytes* que ocupa dicho identificador. Aunque los 12*bytes* de la `primary key` son menores que los 12.5*bytes* del `cluster index`, el identificador también tiene que ser almacenado, por lo que perdemos espacio de almacenamiento total, al tener que añadir a los 74*bytes* que ocupa cada tupla, los 8*bytes* adicionales del identificador artificial.

Así pues, aunque el ratio de espacio ocupado entre índices y datos se sitúe en 2.5 veces más, es un coste asumible respecto a la eficiencia lograda. Esta eficiencia adquirida por la implementación de los índices es estudiada en el Estudio 5.9.4.

a ↑Los 2 bytes adiciones son por la precisión a microsegundo.
b ↑Denominados `TINYINT`,`SMALLINT`,`MEDIUMINT`,`INT` y `BIGINT` respectivamente.
c ↑Limitación que ya se ha visto salvada a partir de la versión en desarrollo 8.0 de MySQL, como se ha indicado en la Sección 5.9.6.3

Estudio 5.9.4: Impacto de los índices en la eficiencia del almacenamiento

Como se ha presentado en la Sección anterior, la implementación de los índices presentada en la Sección 5.9.6.4 supone un consumo considerable de los recursos de almacenamiento disponibles, siendo el ratio en cuanto a tamaño de los índices respecto a los datos cercano a 2.5 veces. Esto es, por cada 100*MB* de datos de pasos de dispositivos a almacenar, se deben almacenar 250*MB* de información adicional.

Este coste es asumible debido al increíble impacto que tiene la existencia de estos índices en la eficiencia del sistema de almacenamiento propuesto. En la Tabla 5.25 se presentan los tiempos requeridos para realizar una serie de consultas concretas en la tabla paso, empleando tanto el sistema de índices, como careciendo de ellos.

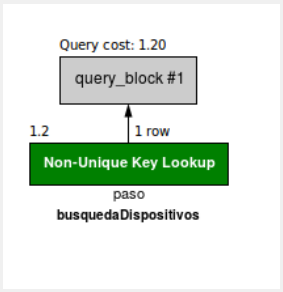
Tabla 5.25
Estudio de eficiencia de los índices, comparados con la ejecución sin los índices indicados. El tiempo indicado se refiere al tiempo de procesamiento del gestor de base de datos, siendo despreciado el tiempo de `fetch`.

TEST (ÍNDICES)	CON ÍNDICE (S)	SIN ÍNDICE (S)	TUPLAS IMPLICADAS
paso.busquedaDispositivos	0.0065	49.810	1
paso.idSensor	0.095	(*)1.648	69 393
paso.ventanaTemporal	0.320	25.503	26 662
Query compleja (*)	0.398	7152.47	69 393

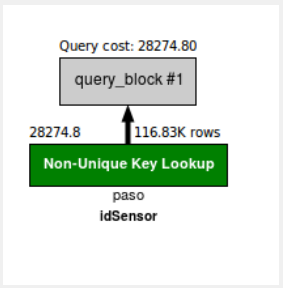
(*) La eficiencia de esta consulta, sin necesidad de emplear el índice, es debido al mecanismo de particionamiento descrito en la Sección 5.9.6.3.

(**) La query empleada implica la unión de las tablas de dispositivos con la tabla de pasos y un sensor.

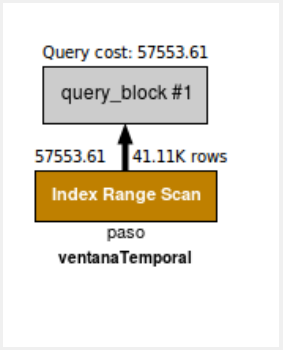
Las consultas empleadas como test, son presentadas en las Figuras 5.70 y 5.71 mediante una representación gráfica de la salida de la ejecución del comando `EXPLAIN`.



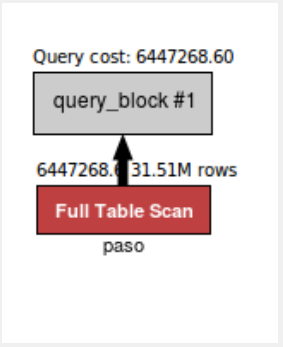
(a) busquedaDispositivos: Con Índice



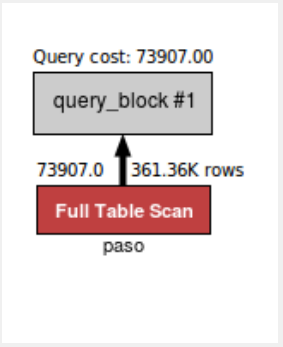
(b) idSensor: Con Índice



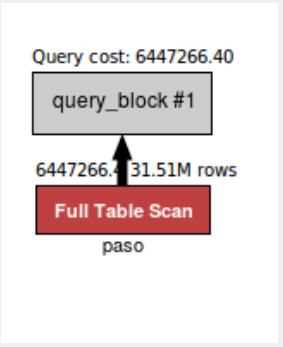
(c) ventanaTemporal: Con Índice



(d) busquedaDispositivos: Sin Índice



(e) idSensor: Sin Índice



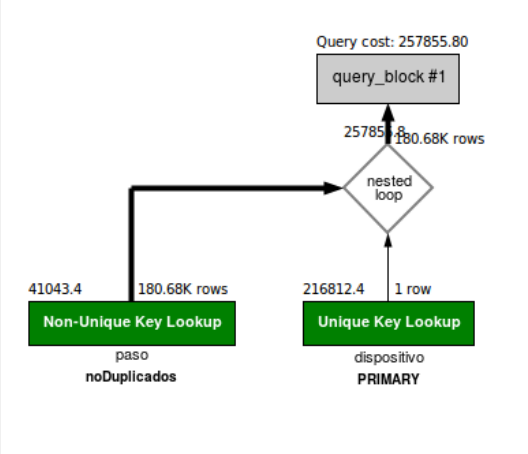
(f) ventanaTemporal: Sin Índice

Figura 5.70
`Explain` de las consultas test para el estudio del impacto en la eficiencia de los índices de la tabla paso.
Las SubFiguras (a),(b) y (c) muestran el resultado de ejecutar la sentencia `EXPLAIN` con los índices implementados en la Sección 5.9.6.4.
Las SubFiguras (d),(e) y (f) muestran el resultado de ejecutar la sentencia `EXPLAIN` sin los índices implementados.

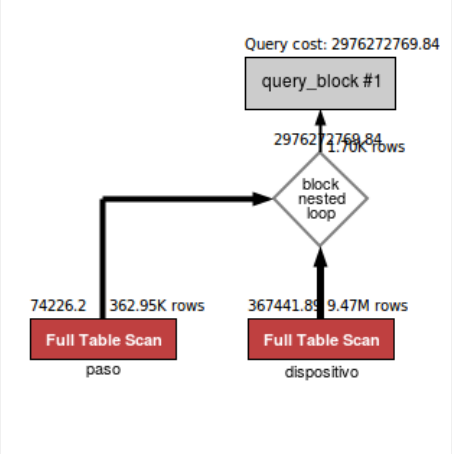
Para una mayor legibilidad de las Figuras se puede visitar la documentación oficial:
`https://dev.mysql.com/doc/workbench/en/wb-performance-explain.html`

En el caso de las consultas que implican búsqueda de dispositivos (p.e. las que implican trazabilidad), el índice propuesto resulta más de 7000 veces superior.

De todas los test realizados, la más exhaustiva es el test que emplea la query compleja, que resulta más de 500 veces más rápida que la alternativa sin emplear índices. Obtener una eficiencia 500 veces superior, pagando unicamente un coste de 2.5 veces más almacenamiento empleado, es bastante asumible.



(a) Query Compleja: Con Índice



(b) Query Compleja: Sin Índice

Figura 5.71
`Explain` detallado de la consulta combinando tablas para el estudio del impacto en la eficiencia de los índices implementados.
La consulta realiza la extracción de información de todos los dispositivos detectados por un sensor de un nodo concreto. Aparte de la mejora en la eficiencia, el empleo de los índices hace que el bucle de unión no resulte bloqueante, lo que permite la inserción de elementos en ambas tablas mientras se encuentra en ejecución la operación.
Para una mayor legibilidad de las Figuras se puede visitar la documentación oficial:
`https://dev.mysql.com/doc/workbench/en/wb-performance-explain.html`

Una advertencia que hace la función `EXPLAIN` es que el no empleo de un índice hace que el motor de base de datos tenga que buscar en toda la tabla en cada una de las peticiones, lo que implica que el tiempo empleado se verá incrementado a media que crezca la base de datos.

Un factor importante es que estas consultas son ejecutadas de forma continuada por los distintos elementos que hacen uso de la fuente de datos. En la Tabla 5.26, se recoge la información de un mes[a] obtenida por el mecanismo `Performance Scheme` (Sección 5.62) para el empleo de los índices de las tablas.

Tabla 5.26
Información del `Performance Scheme` de un mes del rendimiento de los índices implementados.

TABLA	ÍNDICE	FILAS TOTALES	TIEMPO TOTAL (μs)	TIEMPO POR TUPLA (ns)
paso	busquedaDispositivos	724 773 972	28 677 889 536.88	25.27
paso	noDuplicados	146 396 608	1 051 538 936.31	139.22
paso	idSensor	28 324 369	23 765 765.95	1191.81
paso	ventanaTemporal	5 470 347	23 389 460.89	233.88
dispositivo	PRIMARY	19 489 984	7 817 761 666.71	2.49
nodo	PRIMARY	865 826	52 717 387.45	16.42

En dicha se presenta tanto el tiempo que se ha empleado en las distintas consultas por número de tuplas devuelto. Es fácilmente calculable el tiempo de acceso al sistema de almacenamiento que ha ahorrado el empleo de los índices propuestos, en base de combinar las Tablas 5.26 y 5.25.

Tabla 5.27
Ahorro en tiempo de acceso al almacenamiento por los índices implementados, haciendo una equiparación aproximada entre el ratio de eficiencia de las distintas consultas test empleadas con y sin índices, y el tiempo de acceso empleado por el sistema en producción con índices durante un mes de computo.

TEST	TIEMPO REAL (HORAS)	RATIO	TIEMPO SIN ÍNDICES (HORAS)
paso.busquedaDispositivos	7.966	×7663.07	61 044.68
paso.idSensor	0.006	×17.34	0.11
paso.ventanaTemporal	0.006	×79.69	0.51
Query compleja (*)	0.292	×17971.03	5 249.23
TOTAL	8.27		66294.53

a ↑Durante ese mes sólo han sido ejecutadas tareas automatizadas del sistema, como las presentadas en la Sección 5.14.4, sin haberse realizado ningún estudio manual, lo cual hubiese incrementado el empleo de los índices estudiados.

Conclusiones

Gracias a las optimizaciones del sistema de almacenamiento local presentado, se consigue en un mes un ahorro en tiempo de acceso a la base de datos de más de 66 000 horas, o lo que es lo mismo, de carecer de los mecanismos de optimización presentados más de 2700 días hubiesen sido necesarios para realizar el procesamiento realizado de forma automática por el sistema durante un mes.

Aunque resulte una trivialidad defender que un mecanismo eficiente resulte más provechoso que un mecanismo no eficiente, se hace necesario notar que sin este tipo de mecanismos (pese al coste en espacio adicional de disco que lleva asociado) la explotación de un sistema de monitorización como el propuesto resultaría inviable. La eficiencia del sistema de almacenamiento local es de vital importancia para la ejecución de los procedimientos requeridos que serán presentados en la Sección 5.10 para generar las estructuras de datos que resumen el estado de la monitorización realizada. De esta forma, puede procesarse en poco más de 8 horas, lo que requeriría de más de 7 años y medio.

5.10 PROCESAMIENTO EFICIENTE DE DATOS

En esta sección se presentan los métodos y procedimientos implementados para el acceso al almacenamiento local que explotan al máximo los mecanismos de optimización presentados en la Sección 5.9.

En primer lugar se presenta una breve discusión sobre el mecanismo basado en procedimientos implementados, con el fin de justificar su uso frente a las alternativas más extendidas y frecuentes.

A continuación se presentan de modo descriptivo los métodos implementados para la obtención de las estructura de datos que son empleadas en el análisis (Sección 5.11), así como los aspectos de eficiencia más relevante de cada uno de ellos.

5.10.1 *Justificación del empleo de `MySQL Stored Programs`*

Para comprender porque es provechoso el empleo de procedimientos implementados o `Stored Programs`, es necesario tener unas nociones sobre como funcionan los plantes de ejecución de consultas en `MySQL`.

5.10.1.1 *Planes de ejecución de consultas en `MySQL`*

Un plan de ejecución de consultas es el proceso que sigue el motor de base de datos para resolver cada una de las consultas que son solicitadas al motor. Este proceso se resume de forma gráfica en la Figura 5.72 y es descrito paso a paso a continuación [243]:

[243] High Performance MySQL: Optimization, Backups, and Replication

1. El cliente envía la sentencia SQL a ejecutar al servidor.
2. El servidor comprueba su `Query Cache`[76] para comprobar si se ha ejecutado la misma `query` recientemente.
3. En caso contrario, el servidor interpreta (`parser`) la consulta en un árbol (`parse tree`) siguiendo la gramática establecida, comprobando que el orden de los `tokens` sea correcto.
4. A continuación, se preprocesa la consulta (`preprocesor`) de forma semántica, comprobando que el nombre de las tablas y columnas sean válidos, se disponga de los permisos necesarios para el acceso y no surjan ambigüedades.
5. La consulta preprocesada es optimizada mediante el `Query Optimizer`, la cual elabora un plan de ejecución óptimo de entre todas las alternativas posibles (`Query execution plan`) [77].

76 ↑Este elemento, por eficiencia, ha sido deshabilitado como se ha presentado en la Sección 5.9.3.

77 ↑El `Query Optimizar` es un mecanismo increiblemente complejo y muy estudiado dentro de los motores de base de datos, definir todo su funcionamiento escapa del ámbito de esta

6. El plan de ejecución es enviado al `Query execution engine` que lo resuelve mediante sucesivas llamadas al motor (`storage engine`). `MySQL` no transforma en `byte-code` los planes de ejecución, sino que elabora un árbol de instrucciones que debe seguir el motor para producir los resultados.
7. Finalmente, el servidor devuelve los resultados al cliente.

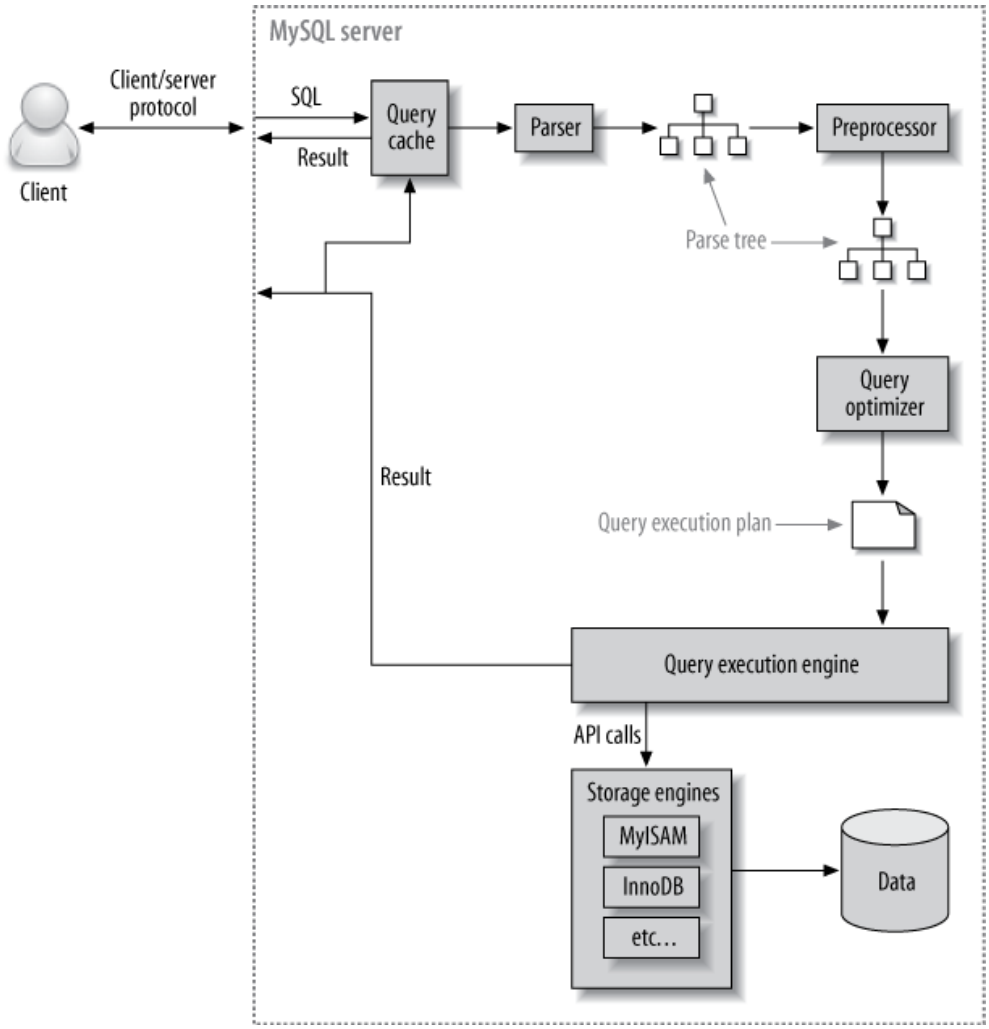


Figura 5.72
Plan de ejecución de una Query en MySQL.
Fuente: High Performance MySQL, Pag 211 [243].

Como se puede evidenciar, hay muchos elementos implicados para la ejecución de una simple consulta. Para cada consulta que se solicite, se tiene que realizar un análisis léxico y sintáctico (`parser`), un análisis semántico (`preprocesor`) y calcular cual es el plan de ejecución más óptimo (`query optimizer`), con el consecuente coste en procesamiento asociado a cada una de estas tareas.

tesis. Lo único que tiene que ser considerado es que para cada consulta a ejecutar, se calculan los distintos planes de ejecución alternativos para elegir el más óptimo. Esta operación es realizada para todas y cada una de las consultas.

Supongamos que la consulta que se ejecuta en el motor de almacenamiento es léxica y sintácticamente la misma, es decir, el árbol resultante es el mismo, cambiando únicamente los parámetros de las consultas, como ejemplo las consultas reflejadas en el Código 5.56.

Código 5.56
Consultas SQL léxica y sintácticamente equivalentes.

```
1  SELECT * FROM paso WHERE tinicio BETWEEN "2018/09/01 09:00:00" AND "2018/09/01
   ↪  10:00:00 AND idSensor = 1122;"
2  SELECT * FROM paso WHERE tinicio BETWEEN "2018/09/02 09:00:00" AND "2018/09/02
   ↪  10:00:00 AND idSensor = 1124;"
```

Debido a que el árbol resultantes es análogo para ambas, el análisis semático de ambas se realizará sobre árboles con igual estructura, cambiando únicamente los valores de las etiquetas por los valores establecidos por cada una de las consultas concretas.

Si ambas consultas son semánticamente correctas, para cada una de ellas se generará un plan de ejecución óptimo y debido a que la estructura sintáctica es la misma, serán planes análogos donde únicamente variaran los valores que deberá emplear el motor de base de datos para las consultas, no el plan de ejecución (orden y operaciones) en si mismo.

De esta forma, es indeseable que el sistema de almacenamiento local malgaste recursos en el análisis y optimización de consultas cuya estructura es similar a consultas ya realizadas en el sistema.

Para paliar este problema y conseguir un mayor aprovechamiento de los recursos `MySQL`[78] provee de un mecanismo denominado `Stored Programs`.

5.10.1.2 *MySQL Stored Programs*

Un programa almacenado dentro de la base de datos (o `MySQL Stored Program`) es un conjunto de instrucciones con un nombre asociado, que es almacenado y ejecutado dentro del motor de base de datos. Para ello se compila el programa bajo demanda y se almacena en una caché temporal[79] el código máquina resultante así como los planes de ejecución de los que haga uso el programa almacenado.

De esta forma, sucesivas llamadas al mismo programa almacenado, harán uso del código compilado y los planes de ejecución ya establecidos, suponiendo un ahorro en computo considerable en sistemas donde se haga uso intensivo de las mismas consultas de forma reiterativa, tal y como es el sistema de monitorización propuesto.

Este sistema de reutilización puede ser también aplicado dentro de las subconsultas que realice un programa almacenado, empleando para ello

78 ↑Al igual que otros gestores de base de datos.
79 ↑Esta caché es común a todos las peticiones de la misma conexión a la base de datos.

`MySQL Statements`[80] que permiten elaborar plantillas de consultas basadas en el mismo principio general que los programas almacenados: el reutilizar los análisis y planes de ejecución.

El empleo de programas almacenados permite también al motor de base de datos ofertar el mismo conjunto de instrucciones básicas a todas los diferentes software que se conecten a él de forma unificada, estándar y segura, ya que impide la ejecución de código no controlado en el motor, así como los mecanismos de inyección de código más rudimentarios.

Ya que el procesamiento es realizado tan cerca del almacenamiento, no se realizan transmisiones de grandes flujos de datos[81] lo que permite un ahorro tanto en ancho de banda como en uso de las comunicaciones.

`MySQL` provee cuatro mecanismos para almacenar programas dentro del gestor: disparadores (`triggers`), procedimientos (`stored procedures`), funciones (`stored functions`) y eventos(`events`[82]), siendo el contexto donde son ejecutados su principal diferencia. En esta tesis serán relevante los procedimientos (que permiten la devolución uno o varios conjuntos de valores tabulares, como una `query` habitual) y las funciones (que permiten la devolución de un único valor primitivo).

5.10.1.3 *Cuando emplear procedimientos*

Es necesario notar que la mayoría de autores [115, 243] coinciden en señalar que únicamente las consultas que impliquen operaciones en las que el motor de base de datos es eficiente (cómo la búsqueda por índices, los cotejamientos, las reducciones, agrupaciones,etc.) son provechosas del empleo de este mecanismo.

[115, 243] MySQL stored procedure programming, High Performance MySQL: Optimization, Backups, and Replication

No toda la lógica de aplicación del sistema ha de ser procesada cerca del almacenamiento, sólamente aquellas operaciones en las que tal cercanía haga uso de mecanismos de optimización (cómo los presentados en la Sección 5.9) para resultar más eficientes que aplicaciones que hagan uso del conjunto de datos seleccionado como entrada de su flujo de procesamiento.

De igual manera, emplear consultas complejas generadas por otros lenguajes, supone una barrera en la unificación de las herramientas y los procedimientos de acceso eficiente disponibles. Disponer de un mecanismo unificado eficiente que permita el procesamiento de los datos, supone que cualquier herramienta o aplicación que acceda a ellos, independientemente de cuando haya sido implementada, hará uso de los mecanismos de procesamiento eficiente aquí presentados, pues suponen un elemento software independiente a la aplicación desarrollada.

80 ↑`https://dev.mysql.com/doc/refman/5.7/en/sql-syntax-prepared-statements.html`
81 ↑Como los que implicaría que una aplicación externa accediese a todos los datos disponibles y realizase el procesamiento requerido.
82 ↑Únicamente desde la versión 5.1 de `MySQL`

5.10.2 *Procedimientos desarrollados*

A continuación se presentan los procedimientos y funciones desarrollados en el motor de base de datos para el acceso a los datos presentados en la Sección 5.1.

Así se presentan procedimientos para la obtención eficiente de pasos de dispositivos (Sección 5.1.3), dispositivos simultáneos (Sección 5.1.5) y trazalidad del movimiento (Sección 5.1.6).

Estos procedimientos permiten calcular de forma eficiente las estructuras de datos que serán fruto de los análisis (Sección 5.11) y algoritmos (Sección 5.12) empleados para la extracción de conocimiento.

5.10.2.1 *Procedimiento: Recuperación de Pasos*

La unidad básica de información almacenada por el sistema de almacenamiento local es el *paso* de un dispositivo por un nodo concreto. Normalmente, se va a solicitar información sobre uno o más nodos determinados por un periodo acotado de tiempo.

En el Código 5.57 se presenta el procedimiento almacenado desarrollado para dicha recuperación de información.

Código 5.57
Código del procedimiento de selección de pasos.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `selectPasos`(
 in Sensor varchar(200),
 in fechaMIN timestamp,
 in fechaMAX timestamp)
BEGIN

SET @query = CONCAT ('SELECT * FROM abdiel.pasoDispositivos WHERE
idSensor IN (', Sensor ,') AND
tinicio > "',fechaMIN,'" AND tinicio < "',fechaMAX,'"');

PREPARE stmt FROM @query;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;

END
```

El empleo de `statements` sirve para componer la petición haciendo uso de concatenaciones simples de texto, lo cual permite proveer de forma sencilla un listado de sensores concreto haciendo uso de la sentencia `IN`. Es habitual en lenguajes de aplicación hacer uso de la sentencia `FIND_IN_SET` cuando se quiere pasar un listado de objetos como cadena de texto, al carecer MySQL de mecanismos nativos para el almacenamiento de vectores. La eficiencia de ambas alternativas es puesta a prueba en el Estudio 5.10.1.

En la Figura 5.73 se presenta el plan de ejecución realizado por el procedimiento, donde se pueden ver los elementos de optimización empleados. En este caso concreto, se puede ver como gracias al índice ventanaTemporal se puede recuperar las 40 tuplas del nodo determinado haciendo uso de un `Index Range Scan` que implica un coste del 48 % del total del procedimiento. La correspondencia entre el dispositivo detectado y la información almacenada se resuelve de forma directa mediante un `Unique Key Lookup` y supone un 28 % del coste del procedimiento. El coste restante es debido a la ordenación del conjunto de datos[83].

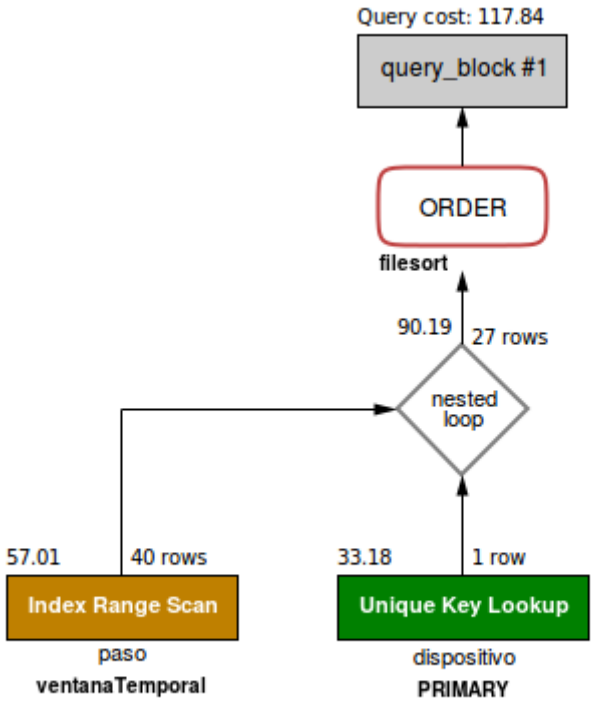


Figura 5.73
Plan de ejecución del procedimiento de selección de pasos.

Ninguno de los elementos implicados en el plan de ejecución está acaparando la mayor parte del coste, siendo los valores obtenidos en el ejemplo significativos de la eficiencia del procedimiento.

83 ↑Aunque los valores se almacenan ordenados para cada sensor al operar el procedimiento con varios sensores, devuelve la información ordenada en función del tiempo, con independencia del sensor. De ahí, el coste asociado a la ordenación.

Estudio 5.10.1: Eficiencia de la selección de pasos

Aunque no resulte evidente, el empleo de programas almacenados ofrece una mejora frente a la ejecución de consultas simples de forma sucesiva. Aunque la consulta empleada está muy optimizada, cada reincidencia de la consulta pasa por los mecanismos descritos en la Sección 5.10.1.1, lo cual implica los análisis, optimizaciones y planificaciones sucesivos, siendo innecesarios al tratarse de consultas de la misma naturaleza.

La tabla 5.28 recoge los tiempos de ejecución de una consulta que solicita los pasos de una ventana de tiempo acotada de un nodo determinado haciendo uso de una consulta SQL habitual y el procedimiento presentado en la Sección 5.57. Se compara también la eficiencia de los métodos `IN` y `FIND_IN_SET`, debido a que este último es muy empleado en las consultas en lenguajes de programación de aplicación para convertir una cadena de texto en un conjunto de búsqueda, debido principalmente a la carencia de `MySQL` de contenedores nativos.

Tabla 5.28
Estudio de eficiencia de los índices, comparados con la ejecución sin los índices indicados. El tiempo indicado se refiere al tiempo de procesamiento del gestor de base de datos, siendo despreciado el empo de `fetch`.

VENTANA DE TIEMPO	CONSULTA IN	CONSULTA FIND_IN_SET	PROCEDIMIENTO	TUPLAS IMPLICADAS
1 hora	0.044s	0.062s	(*) 0.045s	40
1 hora	0.044s	0.062s	0.042s	40
1 día	0.066s	0.079s	0.057s	587
1 mes	0.141s	1.454s	0.134s	16 156
1 año	0.289s	4.508s	0.197s	97 847

(*) La primera sentencia no hace uso de la reutilización del código y de plan de ejecución.

Conclusiones

Aunque la mejora en tiempo es ínfima, sirve para justificar el empleo de procedimientos almacenados frente a los detractores de los mismos que sostienen que el empleo de procedimientos es ineficiente frente al uso de consultas en la totalidad de escenarios.

Referente a la comparativa entre los métodos `IN` y `FIND_IN_SET`, el empleo del método nativo `IN` resulta mucho más eficiente. Esto hace más complicadas las consultas desde fuera de `MySQL` pero supone un coste asumible frente a la mejor eficiencia alcanzada.

5.10.2.2 *Procedimiento: Agrupación Pasos Por intervalos*

Como se presentó en la Sección 5.1.3 los pasos de dispositivos detectados pueden ser agrupados en ventanas temporales, como por ejemplo horas o días, lo que implica calcular cuantos dispositivos distintos han pasado por las inmediaciones del *nodo* en el intervalo de tiempo determinado. Según se presentó en la Figura 5.9, en el ámbito de esta tesis se considera que un paso determinado sólo puede puede ser contabilizado en un único intervalo de tiempo, aunque abarque más tiempo su estancia. De igual manera, un mismo dispositivo sólo puede ser contabilizado una única vez, aunque haya realizado varios pasos en la ventana de tiempo marcada. De esta manera si la ventana de tiempo es de un día, aunque un dispositivo haya pasado varias veces cerca del nodo a lo largo de todo ese día, se deberá contabilizar una única vez.

Para el cálculo de los pasos de dispositivos por un intervalo de tiempo acotado se provee el procedimiento almacenado presentado en la Código 5.58.

Código 5.58
Código del procedimiento de agrupación de pasos.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `agrupaPasos`(in Sensor
↪  varchar(200), in fechaMIN Varchar(40),in fechaMAX Varchar(40),in
↪  intervalo INT)
BEGIN
DECLARE inter INT DEFAULT intervalo*60; DECLARE corte INT DEFAULT 19;
IF intervalo >= 1440 THEN SET corte = 10; END IF;
SET @query = CONCAT ('SELECT
 SUBSTR(FROM_UNIXTIME(TRUNCATE(UNIX_TIMESTAMP(tinicio) / ',inter,',0) *
↪  ',inter,'),1,',corte,') AS Fecha,
  idSensor, COUNT(*) AS Total, COUNT(DISTINCT idDispositivo) AS Unicos
 FROM abdiel.paso WHERE
  idSensor IN (', Sensor ,') AND
  tinicio > "',fechaMIN,'" AND tinicio < "',fechaMAX,'"
 GROUP BY TRUNCATE(UNIX_TIMESTAMP(tinicio) / ',inter,',0),idSensor
 ORDER BY Fecha ASC;');
PREPARE stmt FROM @query;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Este código se basa en varios principios destacables. Trabajar con estructura de datos de tiempo siempre es complejo, lo cual complica el establecimiento de la ventana de tiempo del intervalo de agrupamiento. En el código se ha resuelto empleando una división entera[84] respecto al tamaño del intervalo de la ventana de tiempo expresado en segundos. Tal cifra es nuevamente multiplicada por el número de segundos de intervalo, lo cual convierte todos las marcas de tiempo comprendidas en el mismo intervalo, en el mismo valor numérico debido a el formato `UNIX TIME` se mide a partir de 00:00:00 UTC del

84 ↑Es decir, los dígitos decimales son despreciados.

1 de enero de 1970. La tabla 5.29 muestra algunos valores para ejemplificar el simple pero efectivo método de calculo.

Tabla 5.29
Ejemplo de calculo de intervalo mediante `UNIXTIMESTAMPP`

MARCA DE TIEMPO	PASO 1	PASO 2	VENTANA
542620803 (...08:00:03)	$\lfloor \frac{542620803}{3600} \rfloor = 150728$	$15072 \times 3600 = 542620800$	Ventana 1 (...08:00:00)
542620925 (...08:02:05)	$\lfloor \frac{542620803}{3600} \rfloor = 150728$	$15072 \times 3600 = 542620800$	Ventana 1 (...08:00:00)
542624415 (...09:00:15)	$\lfloor \frac{542624415}{3600} \rfloor = 150729$	$15072 \times 3600 = 542624400$	Ventana 2 (...09:00:00)
542627411 (...09:50:11)	$\lfloor \frac{542624415}{3600} \rfloor = 150729$	$15072 \times 3600 = 542624400$	Ventana 2 (...09:50:11)
542628001 (...10:00:01)	$\lfloor \frac{542628001}{3600} \rfloor = 150730$	$15072 \times 3600 = 542628000$	Ventana 3 (...10:00:00)

Este método permite que la ventana sea cualquier valor definible por una duración concreta de segundos. Y además, permite que la marca de tiempo sea convertible a una etiqueta concreta basada en el primer instante de tiempo que pertenece a dicho intervalo. Esta etiqueta es generada incluyendo las horas o no en función del valor de la variable `corte` que es calculado mediante el tamaño de la ventana del intervalo deseado.

La Figura 5.74 muestra el plan de ejecución del procedimiento. La búsqueda de los pasos empleando el índice ventanaTemporal y un `Index Range Scan` supone un coste del 99.15 % del coste total del procedimiento. El resto del coste, es el asociado a la agupación (`GROUP`) que calcula la reducción a cada intervalo de tiempo. Este coste resulta ínfimo debido a lo optimizados que están los motores de bases de datos para realizar tareas de agrupación y reducción.

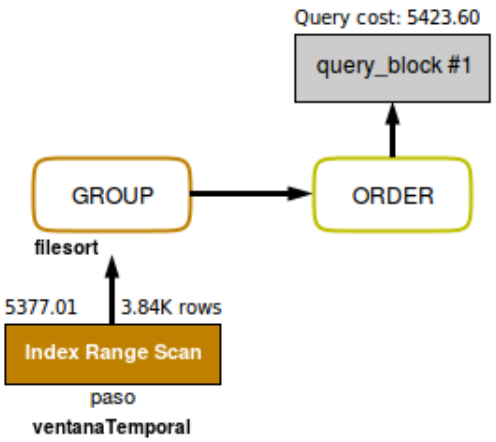


Figura 5.74
Plan de ejecución del procedimiento de agrupación de pasos.

La importancia de la eficiencia de este método es crítica, debido a que el sistema de monitorización propuesto no está interesado en el comportamiento individual, sino en el comportamiento colectivo en base a ventanas de tiempo. La flexibilidad, eficiencia y escalabilidad de este método han sido claves para la obtención de los datos actuales en tiempos cercanos al real.

Estudio 5.10.2: Eficiencia empírica de la agrupación de pasos

Para demostrar la eficiencia del procedimiento de agrupamiento de pasos se calcula el coste de la consulta y el tiempo de ejecución de la misma para distintos tamaños del conjunto de datos[a] en la Tabla 5.30. Es decir, para el primer método se ha empleado una ventana de búsqueda de 1 hora, para la segunda un ventana de búsqueda de 1 día, y así de forma sucesiva.

Tabla 5.30
Estudio de eficiencia de la agrupación de pasos en función del tamaño del conjunto de datos buscado.

TAMAÑO CONJUNTO	COSTE TOTAL	COSTE CONSULTA	COSTE REDUCCIÓN	TUPLAS IMPLICADAS	PORCENTAJE BÚSQUEDA	TIEMPO EJECUCIÓN
1 hora	200.12	198.41	1.71	141	99.15 %	0.0076s
1 día	5423.6	5377.01	46.59	3840	99.14 %	0.075s
1 semana	63326.73	62782.31	544.42	44844	99.14 %	0.15s
1 mes	306659.93	304025.01	2634.92	217160	99.14 %	0.662s
6 año	823175.66	816084.81	7072.85	582917	99.14 %	3.971s

a ↑No confundir el tamaño del conjunto de datos, con el tamaño de ventana del intervalo de agrupación.

Conclusiones

El porcentaje de coste implicado del procedimiento en la búsqueda de túplas que cumplan las condiciones temporales establecidas parece permanece costante independientemente de la ventana de búsqueda. Esto es debido, a que tal y como se presentó en la Sección 5.9.6.1, la búsqueda de ventanas temporales gracias a las optimizaciones realizadas tiene un orden de eficiencia de $O(2 \times log\ n)$.

El tiempo de ejecución crece de forma lineal debido a que una mayor ventana temporal de búsqueda implica una mayor cantidad de datos a ser procesados. Debido a que los datos no se distribuyen a lo largo del tiempo de forma uniforme, esta linealidad no es exacta en ventanas pequeñas. Sin embargo, el tiempo de ejecutar los datos de 6 meses es aproximadamente seis veces el tiempo necesario para ejecutar los datos de un mes.

La eficiencia del procedimiento de agupamiento de pasos y su escalabilidad es más que manifiesta en base a los resultados obtenidos en el estudio.

5.10.2.3 *Procedimiento: Agrupación Simultáneos Por intervalos*

Cómo se presentó en la Sección 5.1.5, una métrica interesante que debe proporcionar el sistema de monitorización propuesto se basa en indicar en un instante concreto de tiempo cuantos dispositivos estaban siendo detectados de forma simultánea.

El cálculo de cuando se considera que un dispositivo está siendo detectado para un instante de tiempo concreto se detalla en la Figura 5.11 y es relativamente sencillo, requiriéndose unicamente que el instante de tiempo determinado se encuentre acotado por los valores que enmarcan temporalmente al paso dispositivo.

Sin embargo, realizar un agrupamiento al número de dispositivos simultáneos detectado en un intervalo de tiempo, en lugar de aun instante de tiempo, resulta algo más complejo. Para ello se implementa la función presentada en el Código 5.59. Una función en el entorno de `MySQL` es análoga a las funciones de los lenguajes de aplicación comunes, esto es, un programa almacenado que devuelve un valor primitivos[85] en base a unos parámetros de entrada.

Código 5.59
Código de la función que calcula el número de dispositivos simultáneos para una ventana de tiempo acotada.

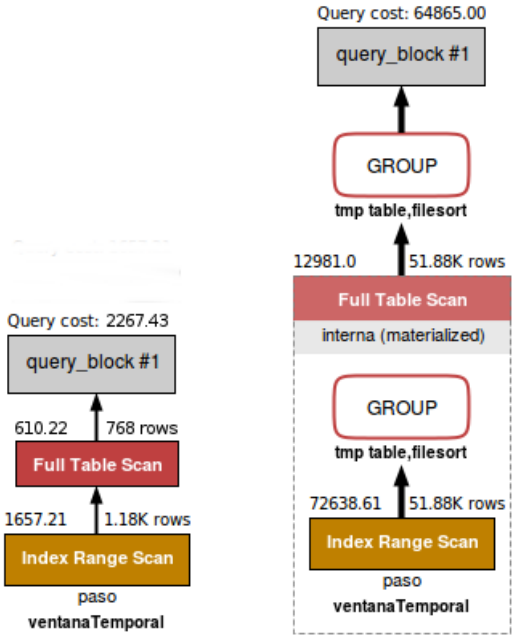
```
1 CREATE DEFINER=`root`@`localhost` FUNCTION `simultaneos`(Sensor INT(10),
  ↪ fechaMIN Varchar(40), fechaMAX Varchar(40)) RETURNS int(10)
2 BEGIN
3 DECLARE temp INT; DECLARE umbral INT default 21600; # (1)
4 SELECT count(distinct idDispositivo) INTO temp FROM paso
5  FORCE INDEX(ventanaTemporal) WHERE idSensor=Sensor
6   AND (tinicio > TIMESTAMPADD(SECOND,-umbral,fechaMIN)) AND (tinicio <
    ↪ fechaMAX)
7   AND (tinicio <= fechaMAX)  AND (tfin >= fechaMIN) ;
8 RETURN temp;
9 END
```

Esta función determina un intervalo de tiempo, al cual se le desea determinar cuantos dispositivos se encontraban en las inmedicaciones. Debido a que el sistema de almacenamiento propuesto es únicamente eficiente en la detección de pasos de dispositivos en función del tiempo de primera detección (`tinicio`), implicar al tiempo de finalización de la detección requeriría emplear toda la tabla en una búsqueda lineal. Para evitarlo, se implementa un umbral de búsqueda dentro de la tabla determinado por el `tinicio`, el cual resulta eficiente en la búsqueda al emplear los índices.

Este umbral determina el máximo de tiempo que sera contemplado como estancia máxima[86] así como el espacio de búsqueda en la base datos para la resolución de la función.

85 ↑Al contrario que un procedimiento, que devuelve conjuntos de datos de consultas.
86 ↑Determinado por el umbral más el tamaño del intervalo de agrupamiento.

Este aspecto puede ser fácilmente reconocible observan el plan de ejecución de la función propuesta, representado en la Figura 5.75(a). De esta forma, se hace evidente como en primer lugar se obtiene una porción del todo el conjunto de datos determinado por el umbral haciendo uso del mecanismo eficiente `INDEX RANGE SCAN` por medio del índice `ventanTemporal` y posteriormente se hace un `Full Table Scan` lineal en toda la tabla para determinar si cada elemento de dicha tabla pertenece al intervalo acotado por la función.



(a) (b)

Figura 5.75
Planes de ejecución de la función simultáneos (Figura 5.75(a)) y el procedimiento encargado del agrupamiento de simultáneos (Figura 5.75(b)).

Estudio 5.10.3: Importancia del umbral en el cálculo de simultáneos

La ventaja de realizar una primera aproximación mediante el uso del mecanismo eficiente proporcionado por el índice `ventanaTemporal` es que se reduce de forma considerable el número de elementos que deben ser evaluados de forma lineal para determinar si pertenecen o no al intervalo proporcionado.

En la tabla 5.31 se presenta la diferencia en tiempo de ejecución del empleo de ambos métodos con distintos tamaños de los conjuntos de datos.

Tabla 5.31
Estudio de eficiencia cálculo de simultáneos con y sin umbral de búsqueda.

TAMAÑO CONJUNTO (PASOS)	TIEMPO SIN UMBRAL	TIEMPO CON UMBRAL
79316	0.212s	0.012s
312347	0.991s	0.015s
1002720	11.776s	0.013s
3809017	27.176s	0.0091s

Conclusiones

Debido a que tiene que hacer una búsqueda exhaustiva en todo el conjunto de datos, el método que no hace uso de un umbral de búsqueda escala de forma lineal según el tamaño de este, lo cual atenta contra la escalabilidad del prodecimiento.

El método que emplea búsqueda mediante umbral, parece permanecer constante, pero realmente su tiempo va asociado al número de dispositivos detectados en el umbral determinado, debido a que es sobre dicho conjunto sobre el que tiene que realizar la búsqueda exhaustiva.

Esta función puede ser empleada en conjunción con el procedimiento de agrupación de pasos presentado en la Sección 5.10.2.2 para calcular el número de dispositivos simultáneos detectados en intervalos de tiempo sucesivos, por ejemplo, realizando agrupamientos cada hora. El código de este procedimiento se presenta en el código 5.60.

Código 5.60
Código del procedimiento de agrupación de dispositivos simultáneos

```
1  CREATE DEFINER=`root`@`localhost` PROCEDURE `agrupaSimultaneos`(in Sensor
   ↪  varchar(200), in fechaMIN Varchar(40), in fechaMAX Varchar(40), in
   ↪  intervalo INT)
2  BEGIN
3  DECLARE inter INT DEFAULT intervalo*60; DECLARE corte INT DEFAULT 19;
4  IF intervalo >= 1440 THEN SET corte = 10; END IF;
5  SET @query = CONCAT ('SELECT *,
6   abdiel.simultaneos(idSensor,Fecha,TIMESTAMPADD(SECOND,',inter,',Fecha)  )
   ↪  as simultaneos
7   FROM (SELECT SUBSTR(FROM_UNIXTIME(TRUNCATE(UNIX_TIMESTAMP(tinicio) /
   ↪  ',inter,',0) * ',inter,'),1,',corte,') AS Fecha, idSensor
8    FROM abdiel.paso WHERE idSensor IN (', Sensor ,') AND
9     tinicio BETWEEN fechaMIN AND fechaMAX
10   GROUP BY idSensor,TRUNCATE(UNIX_TIMESTAMP(tinicio) / ',inter,',0)
11  ) as interna group by idSensor,Fecha;');
12 PREPARE stmt FROM @query; EXECUTE stmt; DEALLOCATE PREPARE stmt;
13 END
```

Este método presenta un plan de ejecución y eficiencia similar al empleado para agrupar los pasos (Sección 5.10.2.2) y se presenta en la Figura 5.75(b).

5.10.2.4 *Procedimiento: Cálculo de tráfico*

En la Sección 3.2 se han presentado diversas magnitudes empleadas en el estudio del tráfico de vehículos, tanto haciendo uso de un único sensor como haciendo uso de varios sensores secuenciales. Para ofrecer información sobre el tráfico haciendo uso de un único sensor[87], se provee el procedimiento almacenado presentado en el Código 5.61.

Código 5.61
Código del procedimiento de cálculo de tráfico empleando un único nodo sensor.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `calculaTrafico`(in Sensor
↪  varchar(200), in fechaMIN Varchar(40), in fechaMAX Varchar(40),in
↪  intervalo INT)
BEGIN
DECLARE inter INT DEFAULT intervalo*60; DECLARE corte INT DEFAULT 19;
IF intervalo >= 1440 THEN SET corte = 10; END IF;
SET @query = CONCAT ('SELECT
 SUBSTR(FROM_UNIXTIME(TRUNCATE(UNIX_TIMESTAMP(tinicio) / ',inter,',0) *
↪  ',inter,'),1,',corte,') AS Fecha, n.idSensor as idSensor, n.nombre as
↪  nombre, n.tipo as tipo, COUNT(idDispositivo) AS Total, COUNT(DISTINCT
↪  idDispositivo) AS Unicos, avg(diferencia) as t_avg, max(diferencia) as
↪  t_max, min(diferencia) as t_min, std(diferencia) as t_std,
↪  group_concat(diferencia) as t_todos
 FROM (SELECT idSensor,tinicio,idDispositivo,tfin-tinicio as diferencia
   FROM paso WHERE
    idSensor IN (', Sensor ,') AND  tinicio > "',fechaMIN,'" AND tinicio <
↪  "',fechaMAX,'"
    ) as t INNER JOIN nodoSensor as n ON t.idSensor = n.idSensor
  WHERE diferencia < 1000 AND diferencia > 0.2
 GROUP BY TRUNCATE(UNIX_TIMESTAMP(tinicio) / ',inter,',0),idSensor
 ORDER BY Fecha ASC;');
PREPARE stmt FROM @query; EXECUTE stmt; DEALLOCATE PREPARE stmt;
END
```

Este código hace uso de la función `group_concat` para recopilar información individual de los elementos agrupados (o reducidos), concretamente se almacena la diferencia de tiempo entre la primera y última detección para todos los pasos considerados en la agrupación. Esto permite el empleo de otros descriptivos estadísticos que no están implementados en `MySQL` nativamente de forma eficiente, como por ejemplo percentiles o medianas.

Adicionalmente, en este código se ejemplifica (Figura 5.76) el método de acceso a la información de nodo y sensores de forma eficiente por medio de las claves primarias, que se resuelve de forma constante.

87 ↑El procesamiento para múltiples sensores se aborda en la Sección 5.10.2.5.

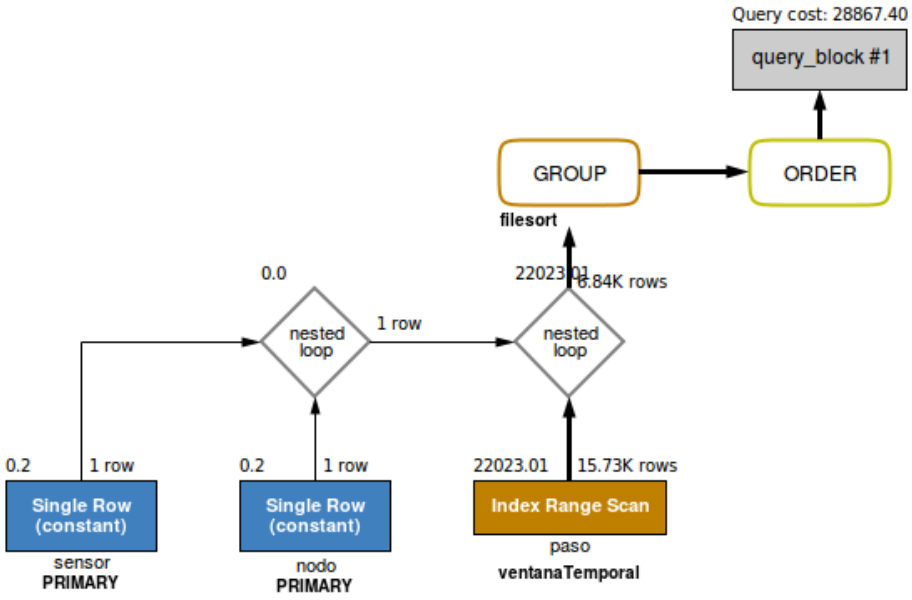


Figura 5.76
Plan de ejecución del procedimiento de cálculo de tráfico empleando un único nodo sensor.

Este método de procesamiento del tráfico únicamente emplea un sensor para el cálculo de las métricas. Debido a que se pretende desplegar una red de sensores, es deseable proveer un método para el cómputo de la trazabilidad.

5.10.2.5 *Procedimiento: Cálculo de trazabilidad de dipositivos*

En la Sección 5.1.6 se han presentado los fundamentos de la trazabilidad de los movimientos de los dispositivos detectados. En esta Sección se presenta la complejidad inherente en el cálculo y procesamiento de las trazas y las rutas, entendiéndose como ruta a aquella sucesión de trazas que componen el recorrido absoluto del dispositivo detectado.

Se presentan también los métodos implementados tanto para el cálculo eficiente de las trazas, así como para realizar el agrupamiento en intervalo de tiempos.

Eficiencia e implicaciones del cálculo de trazas

El cálculo de las trazas entre varios nodos sensores implica, que para cada dispositivo detectado por un nodo sensor denominado nodo origen, ha de ser buscado ese mismo dispositivo entre los posibles nodos destinos. Si ese dispositivo ha sido detectado en otro nodo posteriormente a la última detección del nodo origen, se dice que existe una traza entre el nodo origen y el nodo destino para ese dispositivo. En la Figura 5.77 se recogen a modo de ejemplo los distintos pasos detectados por una red de sensores.

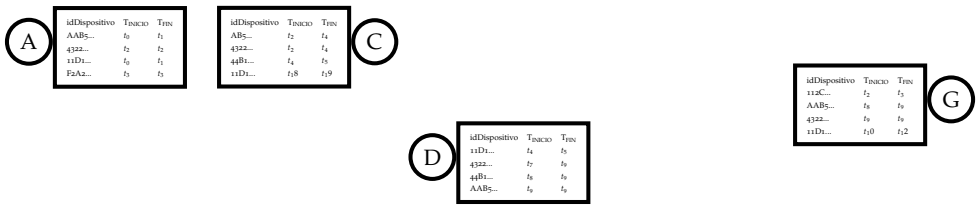


Figura 5.77
Ejemplo de pasos a lo largo distintos nodos sensores. Cada nodo sensor ha generado una lista de pasos de dispositivos que han sido detectados en sus inmediaciones.

Si se desea por ejemplo calcular las trazas $A \rightarrow C$ se han de explorar todos los dispositivos detectados en el nodo A y comprobar si estos han sido detectados a posteriori en el nodo C. Supongamos que el número de dispositivos detectados en un nodo se denota como n_{nodo}. La búsqueda de los dispositivos de A que han sido detectados en C tiene un orden de eficiencia de $O(n_A \times n_C)$, y la operación inversa también tiene que ser calculado, ya que los dispositivos que han realizado la traza $A \rightarrow C$ no son los mismos que han realizado $C \rightarrow A$.

Por lo tanto la eficiencia de realizar la búsqueda de dispositivos que se han desplazado entre dos nodos cuales quiera se puede notar como sigue:

$$O(n_A \times n_C + n_C \times n_A) = O(2 \times n_C \times n_A) \tag{5.1}$$

Para facilitar el cálculo, como es habitual, se considera que el tamaño de ambos conjuntos es n. La eficiencia depende, por tanto, del número de nodos

sensores implicados y del tamaño del conjunto de datos. Así para un número indeterminado de nodos, la eficiencia viene determinada por:

$$O\left(\sum_{i=0}^{nodos} n^2 \times (2 \times (i-1))\right) = O\left(n^2 \times (i-1) \times i\right) = O\left(n^2 \times (i^2 - i)\right) \quad (5.2)$$

Esto implica que cualquier método encargado de calcular las trazas entre distintos nodos sensores de una red, escalará de forma cuadrática tanto en el orden del tamaño de los datos capturados por cada nodo, como por el número de nodos contenidos en la red de sensores. Así por ejemplo, una red con 5 nodos realizaría $5^2 - 5 = 20$ búsquedas de $O(n^2)$, mientras que una red con 10 nodos, $10^2 - 10 = 90$ búsquedas de igual orden de eficiencia.

Sin embargo esta ecuación de eficiencia considera que la búsqueda de dispositivos recurrentes tiene un orden de eficiencia cuadrático, lo cual no es cierto gracias a los mecanismos de optimización del motor de almacenamiento presentados en la Sección 5.9.6.2.

El mecanismo de almacenamiento emplea un Árbol-B o `BTREE` (Figura 5.65) para almacenar información sobre los pasos de cada dispositivo concreto, y de esta forma, localizar los pasos recurrentes de un mismo dispositivo de forma eficiente. Esto permite resolver la búsqueda de dispositivos en un orden de eficiencia $O(n \times log_2(n))$ en lugar de $O(n^2)$. El coste a pagar por esta eficiencia es una mayor ocupación de espacio en disco, aspecto que es discutido en la Sección 5.9.6.4. lo cual supone una mejora significativa de la eficiencia.

La eficiencia por tanto sobre el sistema de almacenamiento local propuesto se presenta en la Ecuación 5.3:

$$O\left(n \times log_2(n) \times (i^2 - i)\right) \quad (5.3)$$

El único aspecto que resulta crítico para la eficiencia es el tamaño de la red de sensores desplegada. Sin embargo este problema puede ser salvado fácilmente gracias al principio de localidad espacial de los nodos sensores, que hace factible que cuanto más alejados estén dos nodos, menos dispositivos sean detectados desplazándose directamente entre ellos.

Finalmente, hay que tener en cuenta una consideración adicional. Debido a que una traza tiene un origen y un destino acotados en un instante de tiempo, un mismo origen e instante de tiempo puede dar lugar a numerosas trazas, siendo tantas posibles como destinos donde haya sido detectado posteriormente el dispositivo detectado en el origen. Un ejemplo de esta situación se representa en la Figura 5.78.

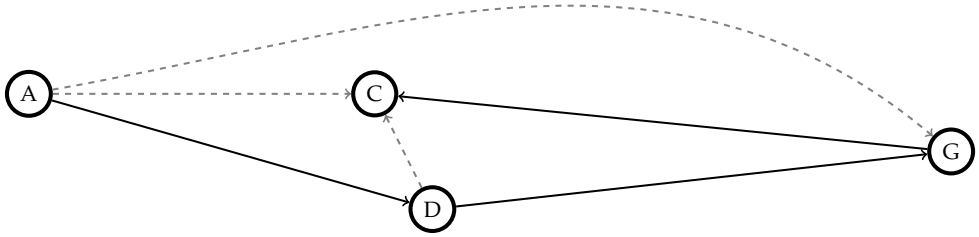


Figura 5.78
Ejemplo trazabilidad de movimiento en una red de sensores.
Las trazas reales han sido pintadas con línea continua. Todas las posibles trazas existentes y calculables han sido representadas en línea discontinua.

En dicho ejemplo, se ha producido el movimiento de un dispositivo detectado siguiendo el orden real $A \rightarrow D \rightarrow G \rightarrow C$. Debido a que las trazas se calculan de forma independiente para cada par origen destino, aunque el desplazamiento natural ha consistido en ir de $A \rightarrow D$, al calcular las tramas de origen A destino C se generará una trama $A \rightarrow C$.

Una generación independiente de las trazas genera también las trazas $A \rightarrow C$, $A \rightarrow G$ y $C \rightarrow D$, lo cual no resulta carente de sentido pues a efecto prácticos, se ha producido un desplazamiento $A \rightarrow C$, debido a que el dispositivo ha sido detectado en A y posteriormente ha sido detectado en C. Sin embargo, el dispositivo detectado no ha realizado una ruta directa ente el nodo A y C, sino que primero ha pasado por los nodos D y G.

Esta peculiaridad complica sobremanera el cómputo y cálculo de las trazas que permitan reconstruir las rutas de manera fiable. No sólo se tienen que calcular las trazas entre los distintos nodos sensores implicados, sino que también han de generarse únicamente las trazas siguiendo el recorrido natural que ha seguido el dispositivo detectado.

Para ejemplificar el método teórico propuesto de cálculo de trazas, se presenta el Estudio 5.10.4, en el que se realiza un cálculo de trazas para un dispositivo detectado concreto.

Estudio 5.10.4: Ejemplo teórico del cómputo de trazas

Se presenta en la Tabla 5.32a los pasos de un dispositivo concreto en los distintos nodos sensores de la red desplegada , siendo t_n el instante de tiempo en el que el dispositivo ha sido detectado por un nodo, con T_{INICIO} recogiendo el tiempo de primera detección y T_{FIN} el instante de tiempo de última detección.

SENSOR	T_{INICIO}	T_{FIN}
A	t_0	t_1
B	t_2	t_2
C	t_{18}	t_{19}
D	t_4	t_5
E	t_6	t_8
F	t_{13}	t_{15}
G	t_{10}	t_{12}

(a) Pasos

	A	B	C	D	E	F	G
A	X	1	18	3	5	12	9
B	X	X	16	2	4	11	8
C	X	X	X	X	X	X	X
D	X	X	X	X	1	8	5
E	X	X	13	X	X	5	2
F	X	X	3	X	X	X	X
G	X	X	6	X	X	1	X

(b) Trazas.

Tabla 5.32
Pasos (Figura 5.32a) que indican los instantes de primera y última detección por cada cada sensor de un dispositivo determinado.
Trazas (Figura 5.32b) generables en función de los pasos detectados de dicho dispositivo por cada sensor. La matriz presenta en la fila el origen de la traza, en la columna el nodo sensor destino, y como valor de la matriz el tiempo de desplazamiento empleado.

En base a dichos pasos se pueden construir las trazas existentes, recogidas en la Tabla 5.32b. De esta forma, se obtiene los nodos sensores de origen y destino del movimiento del dispositivo, además del tiempo requerido para realizar dicho desplazamiento.

La cantidad total de posibles trazas generables se recogen en la Figura 5.79. Sin embargo, estas trazas no implican ningún tipo de secuencialidad, orden o ruta, indicándose unicamente que un dispositivo ha sido detectado por varios nodos distintos a lo largo del tiempo.

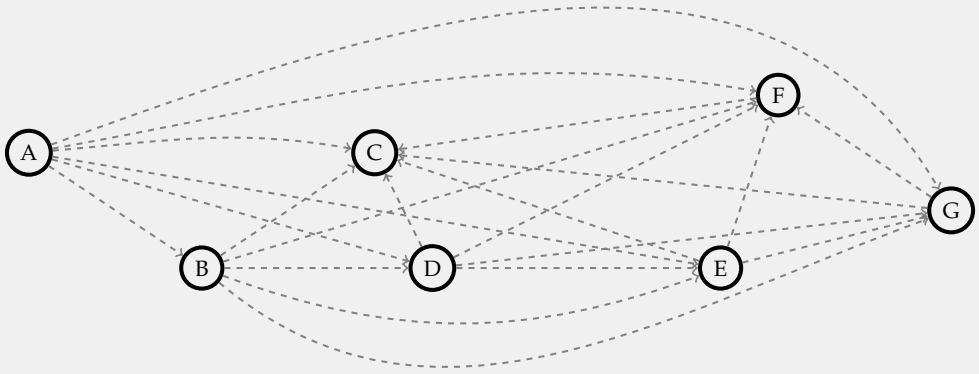


Figura 5.79
Ejemplo cálculo de trazas: Posibles trazas calculables, donde se genera una traza por cada par origen destino, indicando así que el dispositivo detectado ha realizado un desplazamiento de un nodo a otro.

Generar todas las trazas implica que únicamente se conoce el tiempo necesario para ir de un origen a un destino, sin importar la elección de la ruta o camino seguido. Por ejemplo, existe una traza $A \rightarrow C$ con un coste de 18 unidades de tiempo. Esto podría darnos una información sobre el tiempo necesario para realizar el desplazamiento entre los nodos A y B es de 18 unidades de tiempo para este dispositivo en concreto. Sin embargo, se está obviando la ruta seguida por el dispositivo detectado, lo cual puede resultar decisivo en el tiempo necesario para realizar el desplazamiento si no ha sido la ruta más óptima.

Para el estudio del tráfico, como se ha visto en la Sección 3.2, la ruta elegida resulta crítica. Una de las fortalezas del sistema de monitorización propuesto, es que es capaz de reconstruir la ruta seguida por el dispositivo detectado. Para ello se debe desarrollar un procedimiento que sea capaz, en base a todas las posibles trazas generables, quedarse unicamente con aquellas que reconstruyen en la medida de lo posible la ruta real seguida por el dispositivo. La figura 5.80 presenta el resultado esperable de la ejecución de un algoritmo que resuelva el problema propuesto.

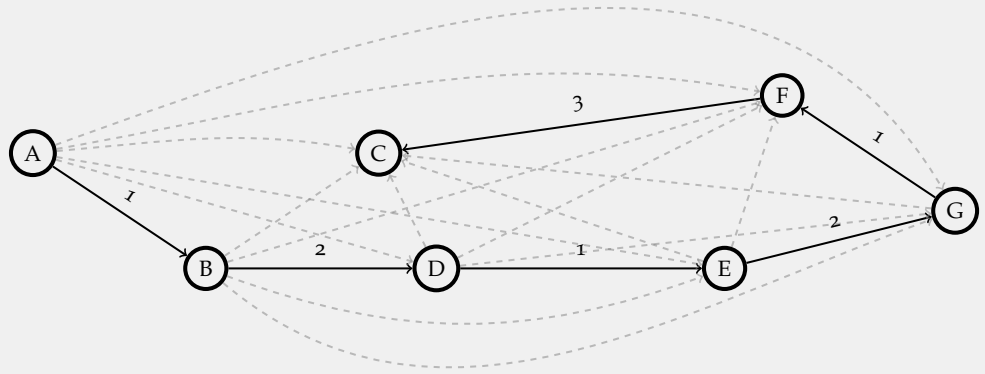


Figura 5.80
Ejemplo cálculo de trazas: Ejemplo de trazas reales calculadas en base a la información de los pasos generados por varios nodos sensores. Las trazas no generadas son pintadas en línea discontínua. Las trazas generadas son pintadas en línea continua, con el coste en tiempo de desplazamiento reflejado.

Conclusiones

El cálculo de trazas de los movimientos no sólo resulta costoso computacionalmente, sino que además, debe ser capaz de reconstruir las rutas seguidas por los dispositivos detectados. Para ello, debe de descartar las trazas que no han sido realizadas de forma directa.

En la Sección 5.10.2.5 se presenta el algoritmo desarrollado de forma eficiente y escalable para el cálculo de las trazas y recostrucción de rutas.

Debido a que el sistema propuesto es esperable que detecte varios dispositivos realizando desplazamientos por la superficie abarcada por su red de nodos sensores, se podrán generar trazas que sigan rutas distintas. El estudio de cual de las rutas seguidas por cada uno de los dispositivos permite discernir que ruta resulta más óptima, así como estudiar las características habituales del tráfico, como por ejemplo, la predilección de los vehículos en un cruce. En la Figura 5.81 se presentan dos dispositivos detectados siguiendo dos rutas distintas con el mismo origen y destino final.

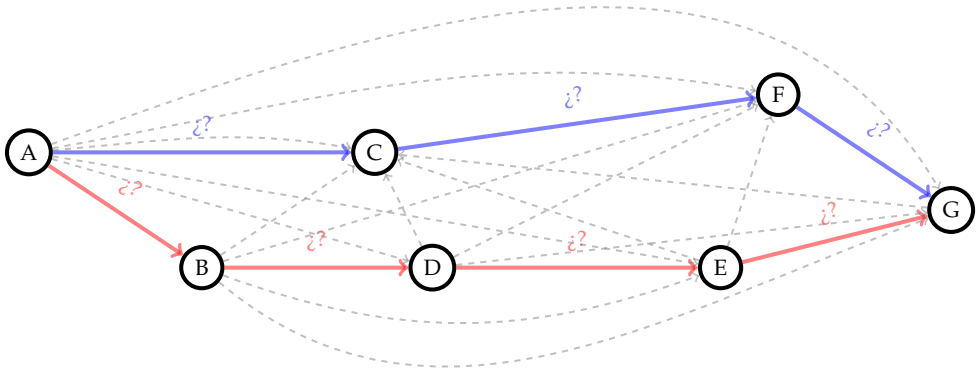


Figura 5.81
Trazas de dos dispositivos distintos sobre la red de sensores. Se presentan dos dispositivos detectados, uno de ellos siguiendo la ruta $A \rightarrow C \rightarrow F \rightarrow G$ y otro siguiendo la ruta $A \rightarrow B \rightarrow D \rightarrow E \rightarrow G$.

Disponer de este tipo de estructuras analizables permite determinar cual de las rutas o predilecciones de caminos está resultando más eficiente, en lo que a tiempo de desplazamiento se refiere, en un instante o circunstancias de tiempo concretos.

Así puede resultar interesante por ejemplo estudiar en primer lugar las trazas entre los nodos A y G para conseguir los tiempos de desplazamiento absolutos entre los dos nodos y posteriormente, calcular las trazas de toda la red de nodos sensores para determinar que ruta ha seguido cada sensor.

Por último, no es realmente relevante el recorrido realizado por dispositivos individuales, sino que lo interesante está en el estudio del comportamiento generalizado de los diversos dispositivos detectados. De esta forma, por ejemplo, puede resulta interesante determinar el porcentaje de dispositivos detectados yendo de $A \rightarrow G$ que optan por hacer $A \rightarrow C$ o $A \rightarrow B$, así como estudiar si este comportamiento es influido por factores externos o cíclicos.

Procedimiento para el cálculo de trazas

Como se ha presentado en la Sección anterior, el cálculo de trazas presenta varios problemas en cuestión de eficiencia y escalabilidad.

Código 5.62
Procedimiento almacenado en memoria para la obtención de trazas de dispositivos detectados entre dos nodo sensores.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `localizaTrazas`(in Sensor
↪  varchar(200), in fechaMIN timestamp, in fechaMAX timestamp, in
↪  intervalo INT, in maximalTime INT)
BEGIN
DECLARE inter INT DEFAULT intervalo; DECLARE corte INT DEFAULT 19;
DECLARE fMax TIMESTAMP DEFAULT fechaMAX; DECLARE fMin TIMESTAMP DEFAULT
↪  fechaMIN;
DECLARE i TIMESTAMP DEFAULT fMIN;
IF intervalo >= 1440 THEN SET corte = 10; END IF;
DROP TEMPORARY TABLE IF EXISTS trazas; SET sql_mode = '';
CREATE TEMPORARY TABLE trazas  (Origen int(10), Origen_tiempo TIMESTAMP,
↪  Destino int(10), Destino_tiempo TIMESTAMP,diferencia
↪  int(4),estancia_origen int(4),estancia_destino int(4), mac binary(20),
↪  UNIQUE INDEX `no_repes` (`Origen`, `Origen_tiempo`) ) ENGINE=MEMORY;
SET @query = CONCAT('INSERT IGNORE INTO trazas SELECT STRAIGHT_JOIN
  t1.idSensor as Origen,t1.tinicio as Origen_tiempo,
  t2.idSensor as Destino,t2.tinicio as Destino_tiempo,
  TIMESTAMPDIFF(SECOND,t1.tinicio,t2.tinicio) as diferencia,
  TIMESTAMPDIFF(SECOND,t1.tinicio,t1.tfin) as estancia_origen,
  TIMESTAMPDIFF(SECOND,t1.tinicio,t2.tfin) as estancia_destino,
  t1.idDispositivo as MAC
 FROM (
  SELECT idSensor,tinicio,tfin,idDispositivo,tDB FROM abdiel.paso
   WHERE tinicio BETWEEN ? AND
    ADDDATE(?, INTERVAL ',inter,' MINUTE)
    AND idSensor in (',Sensor,')) as t1
  INNER JOIN (
   SELECT idSensor,tinicio,tfin,idDispositivo,tDB FROM abdiel.paso
    WHERE tinicio BETWEEN  ?
     AND  ADDDATE(?, INTERVAL ',inter,'+',maximalTime,' MINUTE)
     AND idSensor in (',Sensor,')) as t2
  ON t1.idSensor <> t2.idSensor
   AND t1.idDispositivo = t2.idDispositivo  AND t2.tinicio > t1.tfin
   AND TIMESTAMPDIFF(MINUTE,t1.tfin,t2.tinicio) < ',maximalTime,'
   AND TIMESTAMPDIFF(MINUTE,t1.tinicio,t1.tDB) < 60
   AND TIMESTAMPDIFF(MINUTE,t2.tinicio,t2.tDB) < 60
  ORDER BY diferencia ASC;');
PREPARE stmt FROM @query; SET @i = i;
WHILE (i+inter<=fMaX) DO
 EXECUTE stmt USING @i,@i,@i,@i;
 SET i = ADDDATE(i,INTERVAL inter MINUTE);
 SET @i = i;
END WHILE; DEALLOCATE PREPARE stmt;
END
```

Para intentar minimizar el tiempo de ejecución del método del cálculo de las trazas se han empleado varios mecanismos que son descritos brevemente

a continuación. Para contextualizar estos mecanismos se hará referencia al Código 5.62 donde se presenta el procedimiento almacenado encargado de realizar dicho cálculo.

Al igual que con los anteriores métodos se ha implementando un procedimiento almacenado en memoria que prepara un `Statement`, el cual es ejecutado en bucle.

Se hace uso de una tabla temporal almacenada en memoria que será receptora de las trazas computadas. Una tabla temporal[88] existe unicamente durante el tiempo de conexión. Esto conlleva que dos conexiones distintas pueden emplear el mismo nombre de tabla sin que se produzcan conflictos.

Adicionalmente, se fuerza un `ENGINE` basado en el motor `MEMORY`[89] para forzar que la tabla es almacenada unicamente en memoria volátil, lo cual garantiza la eficiencia de lectura y escritura. De no forzar este `ENGINE`, se generaría una tabla en disco, lo cual es indiscutiblemente más ineficiente.

Se opta por este método, en lugar de tablas temporales gestionadas automáticamente por MYSQL para poder imponer condiciones adicionales a dicha tabla, como por ejemplo el `ENGINE` o poder emplear métodos de optimización de la tabla. Esta tabla temporal, denominada trazas, es el contenedor donde se insertarán las trazas calculadas.

El procedimiento recibe por parámetro el listado de nodos sensores (`Sensor`), la ventana de tiempo (determinada por las variables `fMIN` y `fMAX`) que indica sobre que rango de valores se debe de realizar el cálculo. El parámetro (`intervalo`) se emplea realizar una división en subtareas del cálculo que será detallado más adelante. Finalmente `maximalTime` determina un umbral a partir del cual no serán calculadas las trazas, esto es, establece un tiempo máximo de consideración para el desplazamiento del dispositivo. Este parámetro es necesario para que el sistema no genere trazas de dispositivos que carezcan de sentido, así como optimizar la eficiencia del cálculo.

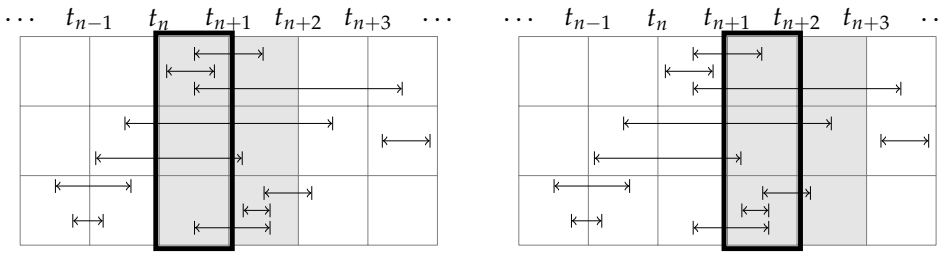
Se realiza un `INNER JOIN` al que se le especifica que la primera tabla se tiene que generar antes que la segunda tabla, mediante la directiva `STRAING_JOIN`. Combinar tablas en `MySQL` resulta costoso, es por ello que el gestor antes de realizar esta operación calcula la cardinalidad de la unión. La cardinalidad se basa en el tamaño de las dos tablas a unir, y el número de combinaciones posibles, que determinará el número de comprobaciones que se tendrán que realizar para poder realizar la unión[90].

Esto implica que realizar un `JOIN` sobre tablas con numerosos registros puede no ser con calculado por el motor, ya que la cardinalidad teórica calculada sobrepase al parámetro de control `MAX_JOIN_SIZE`. Para lidiar con este problema, se propone descomponer el cálculo en cálculos más pequeños, que permitan no sobrepasar la cardinalidad calculada, empleando para ello distintas ejecuciones del mismo `STATEMENT` sobre distintos subconjuntos del

88 ↑`https://dev.mysql.com/doc/refman/5.7/en/create-temporary-table.html`.
89 ↑`https://dev.mysql.com/doc/refman/5.5/en/memory-storage-engine.html`.
90 ↑Por ejemplo, una tabla con 1 000 registros y otra tabla con 5 000 registros y ningún mecanismo de optimización adicional, tendría que realizar 5 000 000 de comprobaciones.

conjunto de datos determinado por la ventana de tiempo solicitada. De esta forma en lugar de realizar una única unión que implique todos los pasos, se realizan varias uniones con subconjuntos más pequeños de los datos, de forma que la cardinalidad no se vea sobrepasada.

Además, debido a que se limita el tiempo máximo de la trazas mediante un umbral determinado por `maximalTime`, también se obtiene una pequeña mejora en eficiencia al limitarse el espacio de búsqueda total, pues carece de sentido emplear un subconjunto mayor que el umbral de búsqueda. Esto implica que todos los pasos registrados que sobrepasen ese umbral, tendrán que ser descartados, por lo que para ese paso no es necesario buscar sus trazas. Esto supone un ahorro al limitar el espacio de búsqueda del subconjunto. En la Figura 5.82 se presenta como se realiza la división del conjunto de datos determinado por la ventana de tiempo en distintos subjuntos iterativos.



(a) Iteración i (b) Iteración $i+1$

Figura 5.82
División iterativa del conjunto de datos para el cálculo de trazas. Se presentan los pasos de varios dispositivos a lo largo de tiempo, dividido en conjuntos de tiempo de igual duración denotados t_n. La caja negra indica el subjunto de búsqueda de la primera tabla de la unión. En gris, el espacio de búsqueda de la segunda tabla.

La primera tabla generada consigue los pasos cuyos tiempos de primera detección se encuentren acotados entre los dos instantes del tiempo del intervalo. En el caso de la primera iteración (Figura 5.82(a)) los instantes t_n y t_{n+1}. Esto divide el tamaño de la primera tabla en el número de intervalos del tamaño determinado por el parámetro `intervalo` en la ventana de tiempo determinada por las variables `fMIN` y `fMAX`[91].

La segunda tabla generada acota los pasos cuyos tiempos de primera detección se encuentren acotados entre el instante de tiempo del intervalo y el tiempo determinado del umbral (`maximalTime`). En el caso de la primera iteración (Figura 5.82(a)), supuesto que el umbral determinado tenga el mismo tamaño que el intervalo de muestreo[92], se encuentra acotado por t_n y t_{n+2}.

91 ↑Por ejemplo, si la ventana tiene el tamaño de un día, y el intervalo se determina a 1 hora, existirán 24 intervalos.

92 ↑Se ha optado por estos valores para simplicar el ejemplo, pero podrían no ser el mismo. Por ejemplo, realizar el muestreo en bloques de 24 horas, pero que el tiempo máximo umbral fuese de 6 horas o 15 minutos, o cualquier tiempo que se desee considerar para descartar las trazas.

El empleo de estos subconjuntos permite minimizar la cardinalidad de las uniones de los conjuntos de datos, reduciéndose tanto el número de comprobaciones a realizar de forma conjunta, como las totales, como se puede ver en el Estudio 5.10.5. Sin embargo, la división en subconjuntos aunque puede ofrecer una leve mejora de la eficiencia, no es un mecanismo suficiente para realizar la unión de forma asumible, debido al alto número de comprobaciones que ha de realizarse.

Estudio 5.10.5: Reducción de la cardinalidad por la división en subconjuntos para el cálculo de trazas.

Supuesto que el conjunto de datos tiene 1 000 registros, la cardinalidad de la unión sobre todo el conjunto de datos tendría que realizar 1 000 000 comprobaciones. La división de la unión en 10 subconjuntos de datos y con un umbral del tamaño de dicho intervalo, tendría que realizar 10 uniones distintas. Supóngase que los datos se encuentran uniformemente distribuidos, cada iteración tendría una cardinalidad de $100 \times 200 = 20\,000$ comprobaciones, es decir 50 veces menos de comprobaciones. Al producirse 10 iteraciones, el número de comparaciones totales realizadas por el sistema sería de 200 000, 10 veces menos que si no se dividiese el conjunto de datos.

Con conjuntos de datos más grandes e intervalos de tiempo más pequeños, la reducción de la cardinalidad se acentúa. Supuesto que el conjunto tenga 1 000 000 de registros, la cardinalidad de la unión será de 10^{12} comprobaciones. Supóngase que el intervalo de división tiene un tamaño de forma de que cada subconjunto tenga 1000 pasos. Se tendrán que realizar 1 000 uniones distintas, cada una de las cuales tendrá una cardinalidad de $1\,000 \times 2\,000 = 2\,000\,000$. Se tendrán que realizar por tanto un total de $2\,000\,000 \times 1\,000 = 10^9$ comprobaciones, es decir, 1 000 veces menos comprobaciones, pero aún así, un número excesivo de las mismas.

Debido a este cómputo, en la Sección 5.9.6.2 se presenta como se ha preparado el sistema de almacenamiento local para ser capaz de resolver este tipo de cálculos de forma asumible. En la Figura 5.83 se presenta el plan de ejecución de una iteración del método de cálculo de trazas presentado.

La primera tabla es determinada mediante el empleo del índice `ventanaTemporal` que permite acotar de forma eficiente una ventana de tiempo determinada, en el caso de esta una iteración i, ventana enmarcada por los instantes de tiempo comprendidos entre i e $i + intervalo$.

La segunda tabla es construída en base la primera por medio de la directiva `STRAING_JOIN`. Es por ello que en lugar de obtener todo el conjunto de datos y comprobar cuales cumplen las características determinadas, se realiza primero la comprobación de la características y luego la obtención de las tuplas que la cumplen. Esto es posible gracias al índice `busquedaDispositivos`, que permite recuperar de forma eficiente las reocurrencias de un dispositivo determinado de la base de datos.

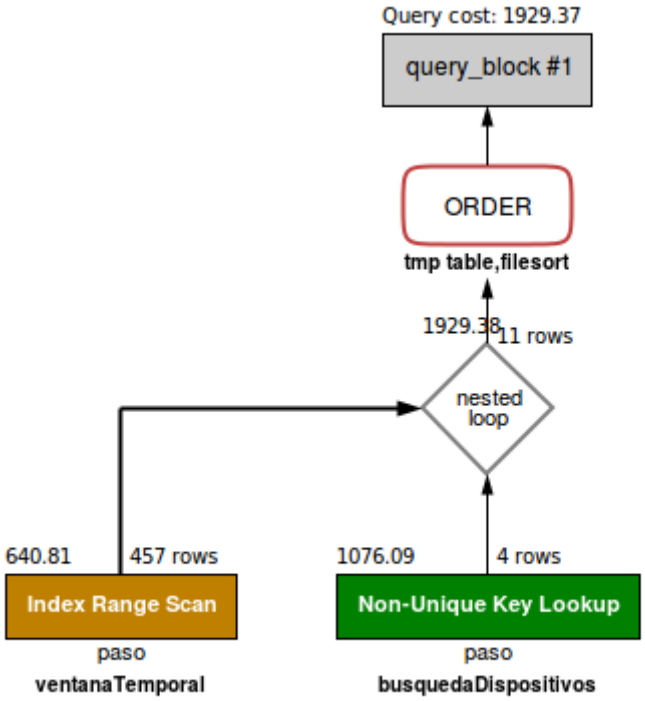


Figura 5.83
Plan de ejecución de una las iteraciones del método de cálculo de trazas.

De esta forma, el `JOIN` calcula la primera tabla, obteniendo un listado de pasos de unos dispositivos determinados. Para todos esos dispositivos, se consulta sus reocurrencias, quedándose con aquellas que cumplen los criterios determinados por el `JOIN`.

El primero de estos criterios es que el paso no puede ser del mismo sensor. Como se presenta en la Sección 5.9.6.4, para la identificación de los pasos se emplea un `cluster index` determinado por los valores de las columnas `idSensor`, `idDispositivo` y `tinicio`. De esta forma, el `JOIN` puede determinar, empleando el `cluster index` que ha recuperado por medio del índice `busquedaDispositivos`, si el paso del dispositivo ha sido detectado por otro sensor, sin necesidad de tener que realizar una búsqueda.

El segundo criterio se basa en comprobar que paso de la traza fue originado primero. Debido a que las trazas son direccionales, el origen de la misma es determinado por el sensor cuyo paso se realizó primero. Para ello, el instante de tiempo de la última detección del nodo origen, tiene que ser menor que el instante de primera detección del nodo destino [93]. Nuevamente, empleando el `cluster index` es posible determinar para el `JOIN` que pasos de los resultantes del primer criterio cumplen esta nueva condición. El instante

93 ↑Esto puede parecer obvio, pero es el mecanismo que impone que las trazas sean direccionales, es decir, que garantizan que la traza de un dispositivo es $A \rightarrow B$ y no $B \rightarrow A$, ya que si no, ambas serían generadas en el proceso de unión.

de tiempo de última detección de la primera tabla, es accesible debido a que esta tabla ha sido generada primero. Además, debido a que la segunda tabla se constituye con un rango sobre de valores `tinicio`, el motor de base de datos resuelve ya esa condición adicional, quedándose únicamente como los pasos que cumplen ese criterio[94].

Debido a que el `cluster index` incluye los campos necesarios para `JOIN`, no son necesarias búsquedas ineficientes, ni realizar comprobaciones sobre dispositivos distintos.

Estudiando los costes reflejados en la Figura 5.83 se aprecia que el coste de realizar la primera tabla es de un 33.21 %, el de constituir la segunda tabla de un 55.77 % y que el coste restante es aplicado en ordenar la tabla antes de realizar la inserción.

Esta ordenación resulta crítica para resolver las rutas de las trazas generadas, problema que ha sido abordado en el Estudio 5.10.4. En dicho estudio se establece que de todas las posibles trazas generables, unicamente serán generadas aquellas que han sido realizadas de forma directa, lo que implica no haber pasado antes por ningún otro nodo.

Realizar este cómputo supondría una nueva comprobación sobre todas las trazas generadas en el `JOIN`, lo cual implicaría nuevamente una nueva unión para quedarse unicamente con aquellas trazas que hayan ocurrido sin que hayan sido detectadas antes en otro nodo sensor.

Sin embargo, el empleo de una tabla temporal nos permite imponer criterios adicionales, como ha sido planteado anteriormente. Esto permite crear para dicha tabla temporal un índice de tipo único basado en los campos nodo de origen y tiempo de origen. Esto implica que en la tabla temporal no podrán existir dos trazas que tengan el mismo nodo sensor de origen en el mismo instante de tiempo. Y de intentar insertarse, sería descartado según impone la cláusula `IGNORE` del `INSERT...SELECT`[95].

Para garantizar que la traza insertada en la tabla temporal es la que corresponde, es decir, es aquella que ha realizado un trayecto directo sin haber pasado por ningún otro nodo, se ordena el resultado de la unión en base al tiempo de desplazamiento de la traza.

De esta forma, las trazas que han requerido menor tiempo de desplazamiento serán introducidas en la tabla antes que aquellas que han realizado un mayor tiempo de desplazamiento. De esta forma, se consigue garantizar que las trazas introducidas son las que han realizado un trayecto directo, debido a que si la traza a introducir fuese $A \rightarrow C$ y se hubiese realizado una ruta $A \rightarrow B \rightarrow C$, la traza $A \rightarrow B$ implicaría menor tiempo de desplazamiento que la traza $A \rightarrow C$. Y de no ser así, entonces la ruta seguida sería $A \rightarrow C \rightarrow B$ en su lugar.

94 ↑Esto es realizado gracias al optimizador de consultas, que unifica las condiciones que empleen los mismos campos.

95 ↑http://ftp.nchu.edu.tw/MySQL/doc/refman/5.0/en/insert-select.html

Estudio 5.10.6: Eficiencia del método de cálculo de trazas

El cálculo de trazas sin el empleo de los mecanismos de optimización aquí presentados resulta prácticamente inabordable, resultando en tiempos muy costosos. En la Figura 5.84 se presenta el tiempo de ejecución de una versión de la consulta de cálculo de trazas sin mecanismos de optimización para distintos tamaños del conjunto de datos y implicando a 2,3 y 4 nodos respectivamente. Debido a que la eficiencia del cálculo de nodos es cuadrática (según la Ecuación 5.2), representar de forma lineal los tiempos no resulta demasiado adecuado. Se presentan los mismos tiempos representados de forma logarítmica en la Figura 5.89(a).

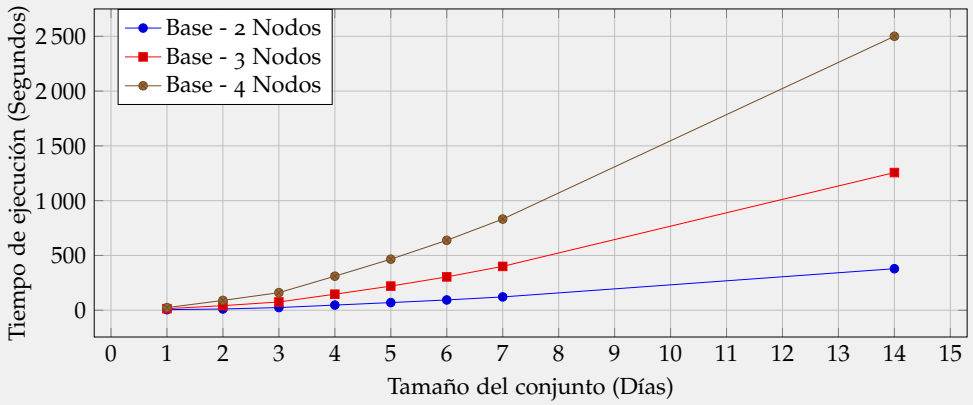
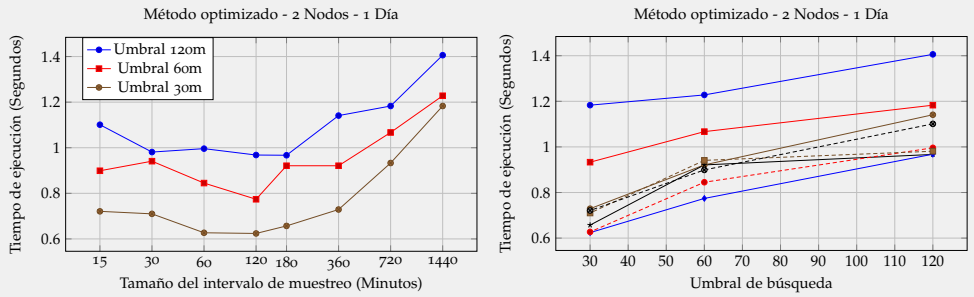


Figura 5.84
Tiempos de ejecución del método cálculo de trazas sin optimizaciones.

El método propuesto presenta dos parámetros de entrada, que se basan en definir un intervalo de muestreo para realizar subconjuntos iterativos, así como determinar un umbral de búsqueda. Se hace necesario estudiar el comportamiento de estos parámetros para distintos valores, con el fin de determinar como influyen su elección en la eficiencia del método optimizado empleado.

Para ello se presenta la Figura 5.85 en la que para un tamaño de conjunto fijo, se ha estudiado la influencia del tamaño de muestreo (Figura 5.85(a)) y del umbral de búsqueda (Figura 5.85(b)).

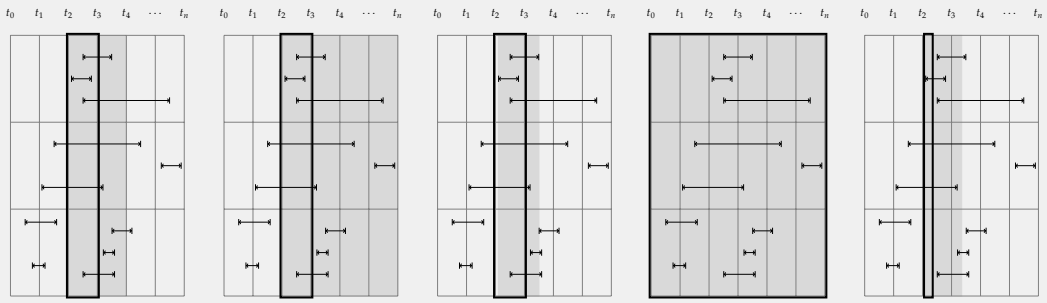


(a) Intervalo de muestreo (b) Umbral de búsqueda

Figura 5.85
Efecto del intervalo de muestreo y umbral de búsqueda en la eficiencia del método de cálculo de trazas.

Aumentar el umbral de búsqueda afecta de forma lineal a la eficiencia, debido a que implica que el tamaño de la segunda tabla del agrupamiento será mayor, al incluir pasos más alejados según se presenta en la Figura 5.86(b) además de que se puede proporcionar información que no sea realmente relevante, como por ejemplo en desplazamientos que habitualmente requieren pocos minutos realizados en horas o días.

De igual manera, reducir el umbral de búsqueda (Figura 5.86(c)) implica descartar pasos en la unión que si podrían cobrar sentido. Por ejemplo, si los nodos se encuentran muy alejados y se requiere un tiempo de desplazamiento mayor que el umbral de búsqueda establecido, el procedimiento no encontrará ninguna traza.



(a) Óptimo (b) Umbral grande (c) Umbral pequeño (d) Intervalo grande (e) Intervalo pequeño

Figura 5.86
Ejemplos de Divisiones iterativas del conjunto de datos para el cálculo de trazas en función del umbral y el tamaño de muestreo

Respecto al intervalo de muestreo, emplear un tamaño de muestreo demasiado grande puede implicar no realizar ninguna división iterativa si el intervalo es más grande que el tamaño de la ventana del conjunto de datos (Figura 5.86(d)) . De igual manera emplear un tamaño demasiado pequeño (Figura 5.86(e)) implica que se realizarán demasiadas iteraciones con pocos

pasos en cada una de ellas. Esto repercute negativamente en la eficiencia, como se observa en la gráfica de la Figura 5.85(a).

En base a esos criterios se evidencia como el intervalo de selección del muestreo esta estrechamente relacionado con el umbral de búsqueda, obteniéndose la mejor eficiencia cuando el intervalo de muestreo es aproximadamente la mitad del tamaño del umbral de búsqueda. Esto es debido a que si el intervalo de muestreo es mayor que el umbral de búsqueda, habrá siempre más pasos en la primera tabla de la unión que en la segunda tabla, debido a que la primera tabla abarca un periodo más grande de tiempo, lo cual afecta a la eficiencia del `JOIN` debido a que por la cláusula `STRAIGHT_JOIN` la segunda tabla se produce en base a la primera. Y como se ha indicado antes, si el intervalo de muestreo es demasiado pequeño, se realizará más uniones, pero que implicarán a pocos pasos.

Con este criterio de selección del intervalo de muestreo en base del umbral, se miden los tiempos necesarios para la resolución de los mismos conjuntos de datos empleados para el caso base (Figura 5.84), resultados que son recogidos en la Figura 5.87 y la Tabla 5.33.

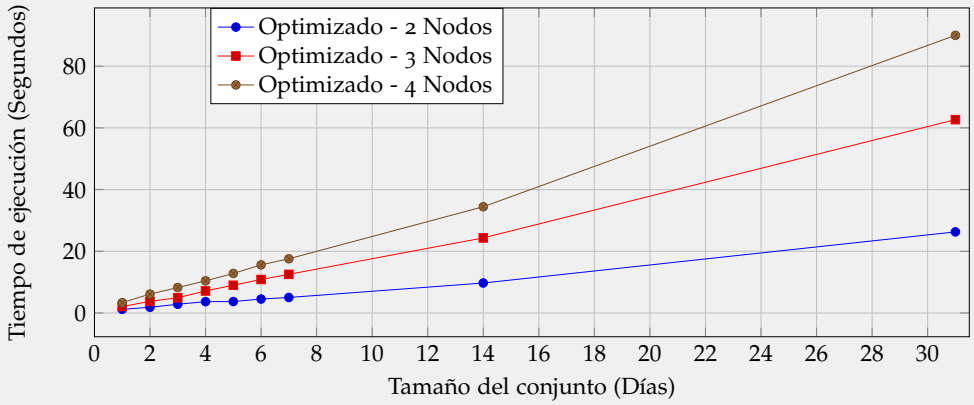
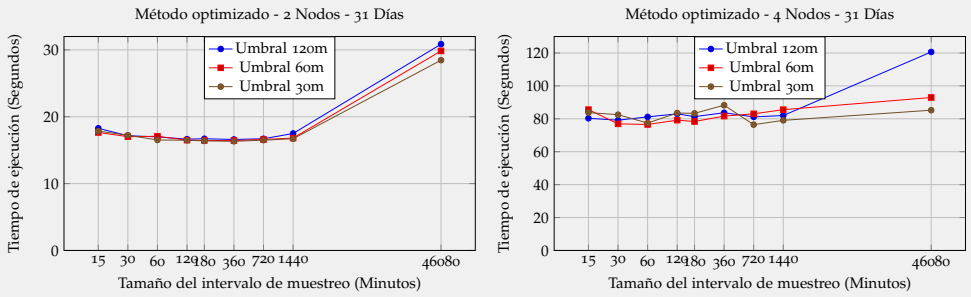


Figura 5.87
Eficiencia del método optimizado de cálculo de trazas

En este caso, la eficiencia es lineal al crecimiento del conjunto de datos como predice la Ecuación 5.3, siendo mucho más escalable que la eficiencia del método base que escala de forma cuadrática.

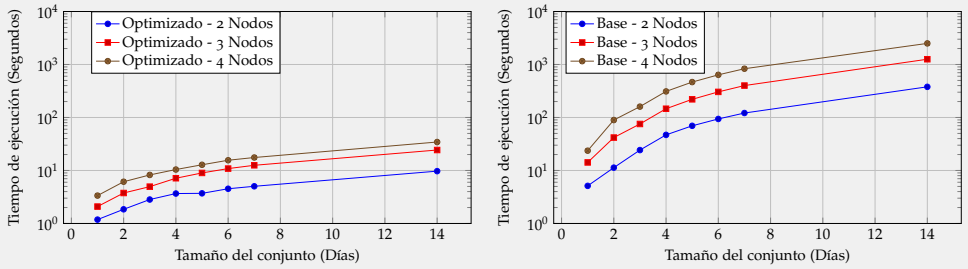
El empleo del intervalo de muestreo añade una pequeña mejora en la eficiencia, lo cual se puede evidenciar en la Figura 5.88 donde el valor máximo del intervalo de muestreo es superior al tamaño del intervalo de datos empleado, por lo que ninguna división iterativa es realizada.



(a) (b)

Figura 5.88
Eficiencia de la división iterativa en el cálculo de trazas entre varios nodos.

Para comprar los dos métodos, el método base sin optimizar y la función optimizada empleando la división iterativa presentada en la Sección 5.10.2.5 y empleando los mecanismos de optimización presentados en la Sección 5.9.6 se presenta la Figura 5.89, empleando una escala logarítmica para poder comparar mejor los procedimientos. De este modo, por ejemplo, para el conjunto de datos de 14 días, el procedimiento optimizado resulta del orden de casi 50 veces más eficiente.



(a) Optimizado (b) Sin optimizar

Figura 5.89
Tiempos de ejecución de los métodos de selección de trazas. El eje de ordenadas se presenta en escala logarítmica para facilitar la compáración de los procedimientos.

Tabla 5.33
Tiempos de ejecución de los métodos de selección de trazas

	BASE			OPTIMIZADO		
TAMAÑO	2 NODO	3 NODOS	4 NODOS	2 NODO	3 NODOS	4 NODOS
1	5.126	14.187	23.632	1.183	2.084	3.357
2	11.310	41.8	89.524	1.851	3.759	6.134
3	24.265	75.452	160.283	2.831	4.953	8.241
4	46.973	146.291	310.571	3.672	7.138	10.418
5	69.674	220.188	465.945	3.705	8.991	12.813
6	93.849	304.187	637.998	4.521	10.856	15.600
7	121.401	400.243	831.926	5.030	12.530	17.570
14	378.919	1256.334	2500.134	9.714	24.33	34.441
31	-	-	-	26.283	62.664	89.992

Procedimiento para el agrupamiento de trazas

El método anterior únicamente devuelve el listado de trazas, pero no realiza ninguna operación de reducción/agrupación en base a un intervalo de tiempo. Para realizar esta operación se propone un método que funciona de manera similar a los métodos presentados en las Secciones 5.10.2.2 y 5.10.2.3 para la agrupación de pasos y dispositivos simultáneos.

Código 5.63
Código del procedimiento de agrupamiento por intervalo de tiempo de las trazas entre dos nodos sensores.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `cuentaTrazas`(in Sensor varchar(200),in fechaMIN timestamp, in fechaMAX timestamp, in intervalo INT, in tiempoMinimo INT)
BEGIN
DECLARE inter INT DEFAULT intervalo*60; DECLARE corte INT DEFAULT 19;
IF intervalo >= 1440 THEN SET corte = 10; END IF;
CALL localizaTrazas(Sensor,fechaMIN,fechaMAX,intervalo,tiempoMinimo);
SELECT
 FROM_UNIXTIME(TRUNCATE(UNIX_TIMESTAMP(Origen_tiempo) / inter, 0) * inter) AS intervalo, CONCAT(Origen,"-",Destino) as idTraza, Origen, Destino,
 COUNT(*) AS total, AVG(diferencia) as tiempos_average, STD(diferencia) as tiempos_STD, group_concat(diferencia) as tiempos,
 group_concat(DATE_FORMAT(Origen_tiempo,"%i")) as tinicios, t1.latitud, t1.longitud, t2.latitud, t2.longitud
 FROM trazas
 INNER JOIN (SELECT latitud,longitud,idSensor from nodoSensor) as t1 ON Origen = t1.idSensor
 INNER JOIN (SELECT latitud,longitud,idSensor from nodoSensor) as t2 ON Destino = t2.idSensor
 GROUP BY idTraza,Origen,Destino,TRUNCATE(UNIX_TIMESTAMP(Origen_tiempo) / inter,0)
 ORDER BY intervalo DESC;
 END
```

Unicamente resulta destacable que el método devuelve la información de los nodos sensores implicados, incluyendo la posición geográfica de estos.

5.10.2.6 *Procedimiento: Cálculo de reincidentes*

La traza de un dispositivo implica el desplazamiento entre un nodo sensor origen y un nodo sensor destino, considerando que ambos nodo sensores son distintos. Esto implica que las visitas reinicidentes de un mismo dispositivo a un mismo sitio acotado por un nodo sensor determinado, quedan fuera de ese procedimiento. Es decir, ese método es incapaz de determinar si un nuevo dispositivo que pasa por un nodo sensor concreto, había sido detectado con anterioridad en ese nodo sensor.

Este tipo de información es muy útil en entornos donde se desee estudiar la fidelidad de clientes o los hábitos de visita periódicos de los portadores de los dispositivos.

El código del procedimiento almacenado en memoria que resuelve el cálculo de visitas reincidentes se presenta en el Código 5.64.

Código 5.64
Procedimiento almacenado en memoria para el cálculo de visitas reincidentes.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `selectReincidencias`(in Sensor varchar(200), in fechaMIN timestamp, in fechaMAX timestamp, in intervalo INT,  in maximalTime INT, in minimalTime INT)
BEGIN
 DECLARE inter INT DEFAULT intervalo; DECLARE corte INT DEFAULT 19;
 DECLARE fMax TIMESTAMP DEFAULT fechaMAX; DECLARE fMin TIMESTAMP DEFAULT fechaMIN;
 DECLARE i TIMESTAMP DEFAULT fMIN; IF intervalo >= 1440 THEN SET corte = 10; END IF;
 DROP TEMPORARY TABLE IF EXISTS reincidentes; SET sql_mode = '';

 CREATE TEMPORARY TABLE reincidentes  (Origen int(10), Origen_tiempo TIMESTAMP, Destino int(10), Destino_tiempo TIMESTAMP,diferencia int(4), mac binary(20),  UNIQUE INDEX `no_repes` (`Origen`, `Origen_tiempo`, `mac`) ) ENGINE=MEMORY;

 SET @query = CONCAT('
  INSERT IGNORE INTO reincidentes
   SELECT STRAIGHT_JOIN
    t1.idSensor as Origen, t1.tinicio as Origen_tiempo,
    t2.idSensor as Destino, t2.tinicio as Destino_tiempo,
    TIMESTAMPDIFF(SECOND,t1.tinicio,t2.tinicio) as diferencia,
    t1.idDispositivo as MAC
   FROM
    (SELECT idSensor,tinicio,idDispositivo,tDB FROM abdiel.paso
      WHERE tinicio BETWEEN ? AND ADDDATE( ?, INTERVAL ',inter,' MINUTE)
       AND idSensor in (',Sensor,')
    ) as t1
   INNER JOIN
    (SELECT idSensor,tinicio,idDispositivo,tDB FROM abdiel.paso
      WHERE tinicio BETWEEN  ?
       AND  ADDDATE( ?, INTERVAL ',inter,'+',maximalTime,' MINUTE)
       AND idSensor in (',Sensor,')
    ) as t2
   ON t1.idDispositivo = t2.idDispositivo
    AND t1.idSensor = t2.idSensor AND t2.tinicio > t1.tinicio
    AND TIMESTAMPDIFF(MINUTE,t1.tinicio,t2.tinicio) > ',minimalTime,'
    ;');
PREPARE stmt FROM @query; SET @i = i;
WHILE (i+inter<=fMaX) DO
 EXECUTE stmt USING @i @i @i @i;
 DEALLOCATE PREPARE stmt;
SET i = ADDDATE(i,INTERVAL inter MINUTE); SET @i=i;
END WHILE;
SELECT *, HEX(mac) as MAC FROM reincidentes;
END
```

Este procedimiento emplea los mismos mecanismos de ejecución que el procedimiento encargado del cálculo de las trazas, con la salvedad de que no es suceptible de ser agrupado, sino que devuelve un listado de visitas de un mismo dispositivo indicando adicionalmente la vez inmediatamente anterior de la visita. Además, incorpora un nuevo parámetro de búsqueda, que es un tiempo mínimo `timeMinimal` para considerar que que es una visita reincidente", puediendo por ejemplo considerar que las visitas en el mismo día no serán consideradas como visita reincidentes.

Al ser un procedimiento donde los datos se buscan en el mismo nodo sensor, resulta mucho más eficiente que la búsqueda de trazas (que implican a varios nodos sensores). Su plan de ejecución se presenta en la Figura 5.90, donde se observa como gracias a los mecanismos de búsqueda eficiente presentados en las Secciones 5.9.6.1 y 5.9.6.2 para ventanas temporales y dispositivos reincidientes no suponen un alto coste al gestor de base de datos.

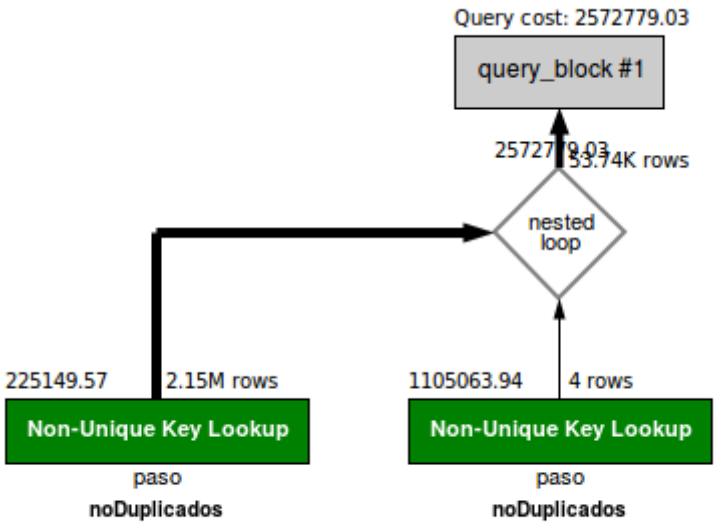


Figura 5.90
Plan de ejecución del procedimiento de cálculo de visitas reinicidentes a un mismo nodo sensor.

5.10.3 *Trabajando con datos geográficos*

Finalmente, en esta Sección se presentan a modo de ejemplo algunas consultas y procedimientos que hacen uso de las facultades geográficas que provee el módulo SPATIAL para poder realizar búsqueda empleando factores geoposicionales. A modo de ejemplo, se presenta el Código 5.65 donde se obtienen los nodos sensores cercanos a uno concreto en un radio de búsqueda acotado. Este método, por ejemplo, puede ser empleado para determinar que nodos sensores pueden interaccionar con un nodo sensor concreto en el cálculo de trazas descrito en la Sección 5.10.2.5.

Código 5.65
Ejemplo de consulta SQL empleando funciones de la librería SPATIAL:
Retorno de nodos más cercanos a un nodo determinado.

```
1  CREATE DEFINER=`root`@`localhost` PROCEDURE `nodosCecanos`(Sensor INT(10),
   ↪  radioBusqueda INT(10))
2  BEGIN
3  SELECT idNodo,
4  ST_Distance_Sphere(
5   (SELECT Point(longitud,latitud) FROM nodo WHERE idNodo = Sensor),
6   Point(longitud,latitud)) as distancia
7  FROM nodo
8  WHERE latitud!=0.0 AND longitud !=0.0
9  AND distancia < radioBusqueda
10 ORDER BY distancia ASC;
11 END
12 -- Duration: 0.00042 sec - Fecth: 0.000014 sec
13 -- Rows returned: 33
14 -- Duration: 0.00035 sec - Fecth: 0.000010 sec Sin ORDER
```

Esta librería puede ser empleada, además, para realizar operaciones de cálculo de distancias en línea recta tal y como se presenta en el Código 5.66.

Código 5.66
Ejemplo de consulta SQL empleando funciones de la librería SPATIAL.
Cálculo de la matriz de distancia (en línea recta) entre nodos.

```
1  SELECT origenID,destinoID,
2   ST_Distance_Sphere(origenLoc,destinoLoc) as distancia FROM (
3     SELECT origen.idNodo as origenID,
4         Point(origen.latitud,origen.longitud) as origenLoc,
5         destino.idNodo as destinoID,
6         Point(destino.latitud,destino.longitud) as destinoLoc
7     FROM (SELECT * FROM nodo
8             WHERE latitud!=0.0 AND longitud != 0.0) as origen
9     INNER JOIN (SELECT * FROM nodo
10            WHERE latitud!=0.0 AND longitud != 0.0) as destino
11    ON origen.idNodo!=destino.idNodo
12  ) ;
13 #Duraction: 0.00012 sec - Fecth: 0.00010 sec
14 #Rows returned: 702
```

5.11 ANÁLISIS

En esta Sección se presentan de forma breve las estructuras empleadas para el análisis de los datos obtenidos por los procedimientos eficientes presentados en la Sección 5.10.2.

Se presentan en primer lugar las descripciones de estas estructuras de datos empleadas en el análisis, tales como las series temporales y las matrices origen destino. En este apartado se describirán las características y normas estudiadas en la bibliografía para el empleo de dichas estructuras, no tanto los métodos de aprendizaje que se realizarán sobre ellas, que serán descritos en la Sección 5.12.

Por último, se presenta brevemente la librería `Mobywit` desarrollada en `R` que permite la obtención de los datos en un entorno que facilita su análisis, interpretación, representación y exportación para su empleo en los algoritmos de aprendizaje.

5.11.1 *Series temporales*

Una serie temporal [37][96] es una sucesión de valores u observaciones de variables tomadas de forma equidistante en el tiempo. Su relevancia, estudio y tratamiento es de vital importancia en la monitorización, ya que los datos agrupados obtenidos son de este tipo. En el ámbito de esta tesis, se ofrecen intervalos de tiempo con valores asociados en multitud de escenarios: la cantidad de dispositivos detectados, la duración de las estancias, dispositivos que han realizado un desplazamiento, los tiempos promedios de esos desplazamiento o cualquier otro tipo de magnitud que sea calculada a lo largo del tiempo.

[37] Introduction to time series and forecasting

Las observaciones realizadas pueden seguir una distribución estadística conocida, es por ello que habitualmente son clasificadas en función de la media y varianza de las mediciones a lo largo del tiempo. En función de dicha variabilidad, se definen dos tipos de series temporales. Una *Serie estacionaria (`stationary`)* es aquella en la que la media y la variabilidad se mantienen constantes a lo largo del tiempo. Una *Serie no estacionaria (`no-stationary`)*, en contraposición, es aquella en la media y/o la variabilidad no se mantienen constantes, mostrando una tendencia a crecer o decrecer a lo largo del tiempo.

Adicionalmente, las series temporales se pueden clasificar en función de si las magnitudes se ven afectadas por valores cíclicos periódicos inherentes al propio valor temporal, más que a la magnitud medida. En función de la exis-

96 ↑La mayoría de definiciones y términos empleados relativos a las series temporales en esta Sección pertenecen en mayor o menor medida a [37]. Para facilitar la lectura, se ha optado por no repetir constante y sucesivamente la cita. Aquellas partes que pertenezcan a otra fuente, serán debidamente citadas.

tencia o no de estos ciclos, se consideran que la serie es *Cíclica o* `seasonal`[97] si presenta para uno o varios intervalos de tiempo una periodicidad cíclica, esto es, existen fluctuaciones periódicas justificadas por la componente temporal y no por la magnitud.

Las series que suelen presentar este tipo de componentes, habitualmente, son aquellas que miden o cuantifican el comportamiento humano.Por ejemplo, el tráfico de personas y de vehículos suele obedecer a comportamientos cíciclos en base a la hora, el día de la semana, si es primero o finales de mes, periodos vacaciones o festivos periódicos.

Finalmente, una serie temporal se puede considerar *autocorrelacionada (*`autocorrelated`*)*, si los valores que toma no son independientes entre si, sino que existe una relación entre un valor puntual y un subconjunto de valores anteriores. Por ejemplo, el número de personas que saldrá de un edificio está fuertemente relacionado con el número de personas que han entrado anteriormente a ese edificio.

5.11.1.1 *Estudio descriptivo de una serie temporal*

El estudio de una serie temporal de forma descriptiva se basa en descomponer la variabilidad de la serie temporal en componentes básicos, que puedan aislar los efectos de los tipos de series presentados anteriormente. La mayoría de los autores coinciden en determinar que una serie temporal puede descomponerse como la suma de varios componentes[98]:

$$X_i = T_i + S_i + I_i \tag{5.4}$$

Siendo X la serie temporal e X_i un valor puntual de dicha serie. Los distintos componentes son definidos como:

T_i es denominada la *tendencia* de la serie y refleja el comportamiento o movimiento suave de la serie a largo plazo.

S_i es la componente *seasonal* correspondiente a los movimientos de oscilación de los valores de la serie debidos a comportamientos cíclicos o periódicos.

97 ↑El empleo del término en inglés a lo largo de esta tesis, es debido a la no existencia de un término en castellano que refleje correctamente el significado de un afectación `seasonal`, pues el empleo de la palabra cíclica no describe correctamente el significado, además de que entra en conflicto con el término `cyclical` que es empleado en inglés para componentes `seasonal` superiores a un año. Además, muchos autores emplean erróneamente el término estacionaria como sinónima de `seasonal`, pero que una serie temporal que presenta una constancia en la media y variabilidad no implica que presente ciclos.

98 ↑Autores señalan que existe un componente adicional C_i correspondientes al componente `cyclical`, es decir, a aquellos periodos de más de un año de duración. Este componente es empleado en series temporales en los que existen históricos de datos de varias décadas pero con intervalos de tiempos basados en años o meses, donde se busca estudiar componentes como los cambios de eras o periodos de crisis o recesión. La escasa importancia de este componente en el ámbito de esta tesis, lo vuelve prácticamente irrelevante debido a que los intervalos trabajados van en el orden de días, horas o incluso minutos..

I_i es la *irregularidad* que recoge aquellas variaciones aleatorias de los valores de la serie que no pueden ser explicadas o recogidas por los componentes anteriores.

La mayoría de métodos estadísticos que emplean series temporales realizan la descomposición en estos componentes para aislarlos y poder estudiar un componente concreto sin la influencia de los otros factores. Así por ejemplo, los análisis de tendencia se centran en esta componente, siendo empleados para el estudio a largo de la magnitud.

5.11.1.2 *Herramientas estadísticas de análisis de series temporales*

Para probar que las series temporales son estacionarias, se emplean los test `Augmented Dickey-Fuller Test (ADF)` [93], `Kwiatkowski-Phillips-Schmidt-Shin Test` (KPSS) [120, 156] y `Phillips-Perron Test (PP)` [93, 208].

[93] Introduction to statistical time series

[120, 156] Generalizations of the KPSS-test for stationarity, Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?

[93, 208] Introduction to statistical time series, Testing for a unit root in time series regression

Para el estudio de los componentes `seasonal` se emplea el método descrito por M. Kendall y A. Stuart [145], que es el estándar de facto en la mayoría de escenarios genérico.

[145] The advanced theory of statistics.

La solera de estos métodos estadísiticos hace que estén disponibles, implementados y muy probados por la comunidad en multitud de lenguajes y herramientas software, lo cual facilita en gran medida su uso.

5.11.1.3 *Predicción de series temporales con métodos estadísticos*

El análisis estadístico de series temporales es una ciencia con casi 100 años de antigüedad [305] debido a su aplicación inmediata a la economía, campo que impulsó su desarrollo y donde ha sido originalmente empleada para el estudio de la dinámica de procesos, la relación entre variables o la influencia de los ciclos periódicos en aspectos como la economía doméstica o el ratio de desempleo [270].

[305] VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers

[270] Time series and forecasting: Brief history and future research

Los estudios realizados en la tendencia y estacionalidad de las series [298] que han sido presentados en la Sección 5.11.1 evolucionaron hasta convertirse en modelos matemáticos capaces de predecir un horizonte de valores [34] de forma aproximada. Según la tendencia de la serie temporal, en un primer momento, se determinaría como aditiva si seguía una tendencia lineal (muestra una forma creciente o decreciente constante) o multiplicativa en caso contrario (muestra una tendencia fluctuante) [177].

[298] Forecasting sales by exponentially weighted moving averages

[34] Time series analysis: forecasting and control

[177] Exponential Forecasting: Some New Variations

Posteriormente se añadirían más tipos de tendencia (ninguna, aditiva, aditiva alisada, multiplicativa y multiplicativa alisada) y estacionalidad ((ninguna, aditiva y multiplicativa)) [128] que permitirían establecer mejores modelos de predicción en función del tipo de tendencia y estacionalidad de las serie temporal, siendo el más conocido el de método de Holt-Winters[51] que se aplica con éxito en casos de tendencia aditiva y estacionalidad multiplicativa.

[128] A state space framework for automatic forecasting using exponential smoothing methods

[51] The holt-winters forecasting procedure

Originalmente estos métodos no gozaron de demasiado prestigio fuera del ámbito económico, hasta que su validez estadística fue probada y gene-

ralizada en los modelos de predicción ARIMA [34]. Los modelos ARIMA o Modelos autorregresivos integrados de media móvil se fundamentan en el empleo de los datos del pasado para realizar las predicciones hacia el futuro, empleando un modelo de tres componentes: autoregresivo (AR), integrada (I) y de media móvil (MA) que dan lugar al nombre del método.

[34] Time series analysis: forecasting and control

El componente autoregresivo se basa en los valores previos de las series al horizonte de predicción. El componente integrado indica que los datos del horizonte son obtenidos mediante la diferencia entre el valor de la serie y su valor previo, de forma que cada valor puntual indica la variación respecto al valor anterior, no la magnitud originaria de la serie. El componente de media móvil, indica que el error de regresión es determinado por la combinación lineal de los errores contemporáneos y previamente anteriores (en contraposición con los errores de toda la serie).

Los modelos ARIMA han sido ampliados con nuevos componentes dando lugar a los métodos VARIMA, VARMA, ARARMA o SARIMA [226] entre otros.

[226] Theory and practice of multivariate ARMA forecasting

Estos métodos resultan muy eficaces en series temporales de sensorización de magnitudes físicas, en donde los valores del horizonte de predicción son más dependientes de los valores anteriores, como por ejemplo, la velocidad del viento [52, 198, 306] o la temperatura [22, 27, 140]. Sin embargo, pueden no resultar preciosos en series temporales con periodicidad múltiple[99] debido a que fueron desarrollados para realizar predicciones en entornos económicos, donde la magnitud de tiempo entre los valores de la serie es de años o meses. Además, los modelos que hacen uso de la componente integrada resultan poco tolerantes a valores anómalos (Sección 5.12.5).

[52, 198, 306] ARIMA-based time series model of stochastic wind power generation, ARIMA vs. Neural networks for wind speed forecasting, ARIMA-based frequency-decomposed modeling of wind speed time series

[22, 27, 140] Predictive data mining on average global temperature using variants of ARIMA models, Trends in global temperature, ARIMA representation for daily solar irradiance and surface air temperature time series

Es por ello que algunos autores señalan que estos modelos pueden no ser suficiente para modelizar y predecir series temporales relativas al trafico y movilidad de personas [168, 171, 297] para cortos intervalos de tiempo, opinando que estos modelos deben ser complementados por métodos de aprendizaje máquina.

[168, 171, 297] Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, Traffic flow prediction with big data: a deep learning approach, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results

99 ↑Por ejemplo, una serie temporada basada en personas por minutos, presentará periodicidad en los minutos, horas, días, semanas, meses, años de los patrones de comportamiento de ese tipo de personas, siendo más relevante el día de la semana (por ejemplo), que la cantidad de personas que haya pasado anteriormente.

5.11.1.4 *Predicción de series temporales empleando `Machine Learning`*

Si los modelos de predicción basados en modelos estadísticos buscan descomponer la serie temporal en componentes susceptibles de ser estudiados y analizados, los modelos de predicción basados en técnicas de aprendizaje automático o `Machone Learning`[100] emplean estructuras complejas para extraer los patrones periódicos de la serie y emplearlos para replicar esos mismos patrones en el horizonte de predicción [197, 312].

[197, 312] Computational intelligence in time series forecasting, Forecasting with artificial neural networks:: The state of the art

Estas estructuras, al ser más complejas, resultan más difíciles de interpretar que los componentes clásicos de las series. Es por ello, que de igual manera que los primeros métodos de regresión estadística, existen numerosos detractores que reniegan de estos métodos al considerarlos demasiado opacos y de obtener resultados muy alejados de la perfección y simpleza matemática [113]. Sin embargo, empíricamente, los métodos basados en aprendizaje automático igualan y mejoran a los métodos estadísticos [8, 296].

[113] Mining the past to determine the future: Problems and possibilities

[8, 296] An empirical comparison of machine learning models for time series forecasting, Beyond regression: New tools for prediction and analysis in the behavioral sciences

En esta tesis las series temporales han sido objeto de los métodos de aprendizaje computacional por medio de `Soft Computing` para resolver dos problemas concretos muy estudiados por los expertos en el área: predicción y detección de anomalías.

En primer lugar se han realizado predicciones de series temporales sobre la fuente de datos propuesta para predecir los eventos futuros. Estos métodos son descriptos en la Sección 5.12.1 y su realización obedece más al estudio de la fuente de datos y su viabilidad para la predicción de eventos, que al avance en cuanto a métodos predictivos se refiere. Estos métodos suelen explotar la tendencia y los componentes `seasonal` para modelizar la serie temporal en base a unos parámetros de entrada y así poder obtener valores futuros.

Para determinar la efectividad de dichos métodos de predicción es habitual comparar los valores predichos con los valores reales de la serie. En el ámbito de la monitorización y sensores, esto es facilmente realizable debido a que los nuevos valores de la serie son generados a lo largo del tiempo. Las métricas empleadas se recogen en el Anexo A.8.

El componente irregular de la serie es difícilmente modelable ya que no obedece a parámetros que sean parametricables fácilmente. En ese comoponente se centra en parte el campo de la detección de anomalías. En la Sección **??** se presenta la definición de anomalía, la importancia de su detección así como los métodos empleados en la tesis para ello.

100 ↑Este tema es abordado en mayor profunidad en la Sección 5.12

5.11.2 *Matrices origen destino*

En la Sección 5.10.2.5 se ha presentado el procedimiento que permite establecer las trazas existentes entre dos nodos sensores. Este procedimiento devuelve un listado de trazas, compuesta cada una de ellas de la siguiente información:

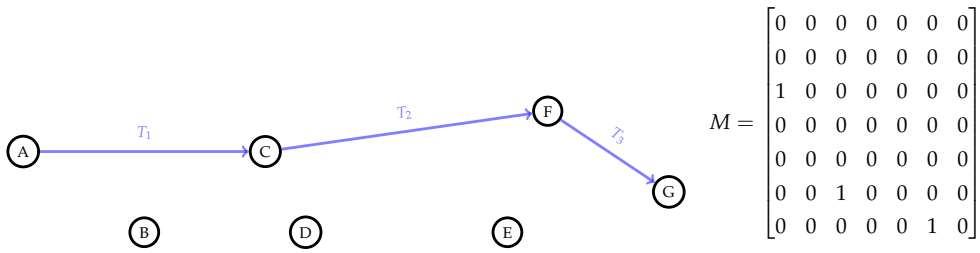
$$Traza \in \left\{ Nodo_{origen}, Nodo_{destino}, T_{origen_{inicio}}, T_{origen_{final}}, T_{destino_{inicio}}, T_{destino_{final}} \right\} \tag{5.5}$$

En base a esta información, como se he ejemplificado en las Figura 5.81, se puede construir un grafo dirigido que marca la ruta seguida por el dispositivo. Un digrafo o grafo dirigido es aquel grafo del cual todas las aristas son dirigidas, es decir, se impone una direccionalidad entre los vértices del grafo. A su vez, un grafo es definido en base a unos nodos (análogos a los nodos sensores de las trazas) y aristas (análogos a las trazas en nuestro caso.)

Un grafo dirigido puede ser resumido en una matriz de incidencias, una matriz de adyacencia o una matriz laplaciana, siendo estas estructuras de datos más fácilmente computables que un grafo.

Una matriz de incidencia representa los grafos en base a las aristas. Una matriz de adyacencia, lo hace en función de los nodos. Una matriz laplaciana combina ambas informaciones.

En la Figura 5.91 se presenta como para la ruta de un dispositivo concreto puede dar lugar a un grafo dirigido, y este a su vez, a una matriz de adyacencia.



(a) Grafo (b) Matriz

Figura 5.91
Grafo dirigido y matriz de adyacencia representado la ruta de un dispositivo entre los nodos.

Sin embargo, varios dispositivos dan lugar a varios grafos cada uno con su ruta. Si se desea estudiar como se ha comportado el trafico en la interacción de varios nodos sensores, los grafos direccionales y las matrices de adyacencia resulta insuficientes.

Se propone el empleo de matrices origen / destino que codifiquen él número de trazas existentes entre las distintas matrices de adyacencia de los

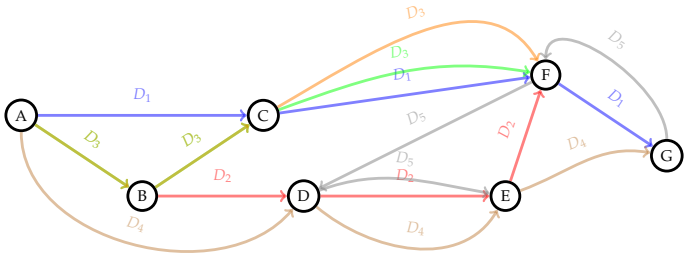
grafos que definen las rutas de cada dispositivo en un intervalo de tiempo acotado.

De igual manera que las matrices de adyacencia, los nodos se representarán empleando las filas y columnas, siendo el valor M_{ij} de la matriz el relativo al nodo i como origen y el nodo j como destino. El valor asignado a la matriz en la posición M_{ij} será el constituido por el número de trazas existentes con origen i y destino j para la ventana de tiempo deseada[101].

Las matrices origen destino o `Matrices O-D` resultan muy empleadas para el estudio del tráfico de vehículos y personas, siendo por ejemplo pieza clave de la teoría de la Modelización del transporte [229] donde son establecidas como el estándar de facto para el estudio de la distribución de los viajes.

[229] Computer modelling for sustainable urban design: Physical principles, methods and applications

De esta forma, se puede modelizar un grafo compuesto por las trazas de varios dispositivos en una Matriz O-D, como se presenta en la Figura 5.92.



(a) Grafo

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

(b) Matriz O-D

Figura 5.92
Grafo dirigido y matriz de adyacencia representado la ruta de un dispositivo entre los nodos.

Este tipo de matrices no son simétricas, ni muestran interacción entre todos los nodos. Adicionalmente, hay un componente analítico que tiene que ser contemplado y es la no completitud de este tipo de estructuras. Salvo sistemas cerrados muy controlados, caso contrario al que nos encontramos, existe tráfico que habrá pasado por un nodo concreto y no haya sido detectado en ningún otro nodo del sistema. Este tipo de información suele ser añadida en un fila-columna adicional, denominadas Z o `zero` para mostrar el número de vehículos que no han sido detectados en otros nodos sensores.

Es habitual además añadir el total de tráfico atraído y producido por cada nodo, con el fin de respetar la magnitud. Otra práctica, empleada en esta tesis, consiste en ofrecer las matrices normalizadas ese tráfico atraído (respecto al destino) y producido (respecto al origen). En el Estudio 5.11.1 se presenta brevemente el cálculo de estas matrices.

Finalmente, los modelos matemáticos más avanzados para la modelización del tráfico, añaden una función de impedancia entre pares de nodos para que los viajes más costosos (habitualmente en métrica de distancia) sean

101 ↑Adicionalmente, se puede emplear cualquier otra magnitud que pueda ser agregada de las trazas, como por ejemplo cualquier estadístico descriptivo de cualquier temporal o relativa a la velocidad.

penalizados. Sin embargo, dado que la ámbito de esta tesis es la aplicabilidad de la fuente de datos propuesta más que la modelización, se han empleado los modelos más sencillos.

Estudio 5.11.1: Matrices O-D normalizadas respecto al tráfico producido o al tráfico atraído.

Se presenta en la Tabla 5.34a el número de trazas contabilizadas en un marco de tiempo determinado, reflejando en las filas el nodo sensor de origen y en las columnas el nodo sensor destino de la traza respectivamente.

Origen \ Destino	A	B	C	TOTAL
A	-	14	16	30
B	11	-	14	25
C	4	22	-	26
TOTAL	15	36	30	81

(a) Matriz O-D

Origen \ Destino	A	B	C
A	-	$\frac{14}{30}$	$\frac{16}{30}$
B	$\frac{11}{25}$	-	$\frac{14}{25}$
C	$\frac{4}{26}$	$\frac{22}{26}$	-

(b) Destinos (Atraído)

Origen \ Destino	A	B	C
A	-	$\frac{14}{36}$	$\frac{16}{30}$
B	$\frac{11}{15}$	-	$\frac{14}{30}$
C	$\frac{4}{15}$	$\frac{22}{36}$	-

(c) Orígenes (Producido)

Tabla 5.34
Matriz O-D absoluta de ejemplo. La magnitud representa el número de trazas que han sido detectadas realizando un desplazamiento entre el nodo origen (fila) y el nodo destino (columna).

Ofrecer valores absolutos no aporta información sobre las predilecciones de rutas, unicamente sobre la magnitud de las mismas. Es por ello que se desea normalizar el número de trazas detectadas un valor acotado. Se presentan dos alternativas: normalizar respecto al origen o al destino determinado.

Normalizar los destinos frente a un origen dado, implica determinar `a donde van` los dispositivos que han sido detectados en dicho origen determinado. Se suele denominar *tráfico producido*.

Normalizar los orígenes frente a un destino dado, implica determinar `de donde vienen` los dispositivos que han sido detectados en un destino determinado. Se suele denominar *tráfico atraído*.

Ambos valores son interesantes de estudio, por lo que serán empleados de forma conjunta ateniéndose al concepto de estudio. En la Tabla 5.34 se presentan a modo de fracción los resultados de normalizar tanto respecto a origen (Tabla 5.34c) y destino (Tabla 5.34b).

La forma más habitual en esta tesis de presentar las matrices normalizadas, es la de indicar directamente el tanto por cierto resultante, tanto en base 1 como en base 100, tal y como se presenta en las Tablas 5.35.

Origen \ Destino	A	B	C
A	-	46.66 %	53.33 %
B	44.00 %	-	56.00 %
C	15.38 %	84.61 %	-

(a) Destinos frente a un Origen

Origen \ Destino	A	B	C
A	-	38.88 %	53.33 %
B	73.33 %	-	46.66 %
C	26.66 %	61.11 %	-

(b) Orígenes frente a un Destino

Tabla 5.35
Matrices O-D porcentuales respecto a origen (Figura 5.35b) y destino (Figura 5.35a).

5.11.3 *Conjuntos de datos clasificables*

Tanto la información de los pasos, las estancias, los dispositivos simultáneos o las trazas de un dispositivo para determinar su rutas son suceptibles de constituir un conjunto de datos clasificables a lo largo del tiempo, con el fin de extraer información sobre los patrones de comportamiento de los portadores de dichos dispositivos.

De esta forma es posible constituir un `dataset`, por ejemplo, en función de parámetros como la hora de entrada y salida a un sitio a lo largo de distintos días de varias personas (dispositivos) distintos, para estudiar si existen tipos o clases de personas con comportamientos similares.

La capacidad de crear nuevos conjuntos de datos datos a partir de la explotación de los datos a lo largo del tiempo, es una de las fortalezas del sistema de monitorización propuesto, ya que al poder reconocer a los dispositivos de forma unívoca, se puede realizar un seguimiento a lo largo del tiempo.

idDispositivo	Día 1 - Entrada	Día 1 - Salida	...	Día n - Salida
8e545e1c31f91f777c894b3bd2c2e7d7044cc9dd	8:24	21:09	...	-
56384a778a0ddf328e492f14ef1aa41588686385	10:47	14:14	...	15:36
052f55137ef0403ccb0a041b86c5f62ed6fc03db	7:14	22:49	...	02:01

Figura 5.93
Construción de conjuntos de datos clasificables en base a las unidades atómicas (p.e pasos).

La capacidad para elaborar conjuntos de datos con características temporales determinadas (por ejemploen base a la ocurrencia de los dispositivos) no es la única capacidad de explotación. Por ejemplo se pueden inducir carácteristicas conocidas, como el tiempo de estancia en función del día de la semana o cualquier otra variable exógena que sea conocida, por ejemplo, si un día es festivo o se ha realizado algún tipo de evento.

De igual manera, las marcas temporales pueden ser descompuestas en multitud de componentes para facilitar el entrenamiento de los sistemas de aprendizaje. Esta descomposición y sus implicaciones será presentado en la Sección 5.12.4.

5.11.3.1 *Librería MOBYWIT para el análisis de datos en R*

Para facilitar las tareas de análisis se implementa una librería en R denominada MOBYWIT que se encarga de realizar las conexiones a la base de datos local (Sección 5.9) y haciendo uso de los procedimientos descritos (Sección 5.10.2) generar conjuntos de datos o `data.frames`[102] con la información solicitada. Estos conjuntos de datos pueden ser analizados mediante la implementación de funciones en R que permitan la reutilización de los procesos analíticos empleados. De igual manera, la representación de los resultados de los análisis puede ser generada de forma procedural y reutilizable, ya sea mediante la generación de gráficos o de informes completos[103].

Se presenta en el Código 5.67 a modo de ejemplo, las llamadas a funciones para recuperar los pasos, pasos agrupados, trazas, trazas agrupadas y agrupaciones de dispositivos simultáneos empleando la librería MOBYWIT.

Código 5.67
Ejemplos de uso de la librería MOBYWITI para la generación de estructuras de datos analizables.

```
1 read.paso.mysql(c(1151,1161),"2018-12-01","2018-12-31 23:59:59")
2 read.paso.summary.mysql(c(1151,1161),"2018-12-01","2018-12-31 23:59:59",
  ↪  intervalo=60)
3 read.trazas.mysql(c(1151,1161),"2018-12-01","2018-12-31 23:59:59")
4 read.trazas.mysql.summary(c(1151,1161),"2018-12-01","2018-12-31 23:59:59",
  ↪  intervalo=60)
5 read.simultaenos.summary.mysql(c(1151,1161),"2018-12-01","2018-12-31
  ↪  23:59:59", intervalo=60)
```

El uso de esta librería permite reutilizar la mayoría de los estudios realizados a lo largo de la tesis, empleando únicamente nuevos parámetros de entrada para generar nuevos conjuntos de datos. Cualquier mejora en la eficiencia de cualquier elemento, es directamente aplicado a los scripts. De esta forma, mejorar el método `read.paso.mysql` hará que todos los scripts que hagan uso de él se vean beneficiados. Y al basarse el método en la invocación de los procedimientos almacenados (Sección 5.10.2), si estos mejoran su eficiencia, no es requerido realizar ninguna modificación en la librería. Además permite mantener el código muy legible y mantenible a largo plazo. Sirva de ejemplo el Código 5.68 donde se muestra un script en R completo encargado de solicitar datos al servidor de almacenamiento local y correr el algoritmo de detección de anomalías para finalmente mostrar el gráfico que representa de forma visual dichas anomalías.

102 ↑Estructura de datos habitual en el entorno R.
103 ↑Se aborda este tema en la Sección 5.14.5

Código 5.68
Ejemplo de uso de la librería MOBYWITI para la detección de anomalías.

```
p <- read.paso.summary.mysql(idSensor = (1152), fechaOrigen="2018/01/01
↪  00:00:00", fechaDestino = "2018/12/31 23:59:59", maxRows = 250000,
↪  intervalo = 60)
t <- mobywit::DetectarAnomaliasTs((p,  max_anoms = 0.05,plot=TRUE)
t$plot
```

Este código puede ser ejecutado bajo demanda por un elemento software periódico (ver Secciones 5.14.3, 5.14.4 y 5.14.6) o ser empleado para la elaboración automática de informes publicables (Sección 5.14.5). Sin embargo, la implementación de esta librería escapa del ámbito de esta tesis, siendo únicamente una herramienta más desarrollada y empleada para el análisis de los datos obtenidos de forma rápida y sencilla, por lo que no serán ofrecidos más detalles sobre la misma.

5.12 APRENDIZAJE AUTOMÁTICO

En la Sección 2.2 se han presentado los problemas de eficiencia a los que se enfrentan las ciudades en el futuro y como la mejora en la gestión de sus recursos resulta la vía más adecuada para su progreso en Europa. Para ello se necesita nutrir a las ciudades y sus administradores de información veráz sobre la utilización de los recursos, con el fin de emplear esta información para generar conocimiento que permita emprender estrategias óptimas y medir el impacto real de dichas estrategias.

Por otro lado, en la Sección **??** se han presentado las magnitudes y métricas que son empleadas en los estudios de movilidad de vehículos y que son aplicables y extrapolables a los estudios de movilidad tanto de personas como de dispositivos inteligentes. Estas magnitudes y métricas sirven de información (Sección 5.11) sobre el estado de la movilidad de las ciudades. Se busca convertir dicha información proporcionada en conocimiento útil para la gestión eficiente de las ciudades.

Finalmente, a lo largo de este capítulo, se ha presentado la metodología y prototipo detrás de la captación de comunicaciones inalámbricas para la monitorización tanto de personas y vehículos, fuente de datos que es propuesta y sujeto de estudio de esta tesis.

En esta Sección se presenta la aplicación de `Sof Computing` para la extracción de información y conocimiento, siendo estos fundamentos aplicables independientemente del origen y naturaleza de la fuente de datos de monitorización.

Se presenta en primer lugar un contexto del `Sof Computing` aplicado a los problemas de movilidad, para introducir a continuación los métodos empleados. La descripción de estos métodos no resulta exhaustiva, sino que se centra principalmente en la adecuación su empleo

5.12.1 *Soft Computing*

El `Sof Computing` es una de las ramas de la Inteligencia Artificial que engloba técnicas empleadas para solucionar problemas que resultan inmanejables con los métodos analíticos y matemáticos convencionales [30, 310].

La existencia del `Soft Computing` lleva implicita la existencia de un `Hard Computing` o computación tradicional. A diferencia de este, el `Soft Computing` es capaz de lidiar con problemas en los cuales la búsqueda exaustiva de la solución optima no es resoluble en tiempos polinómicos. Además el `Soft Computing` debe de estar preparado para trabajar frente a imprecisión, incertidumbre, inexactitud o información incompleta, de manera análoga a la que funciona la mente humana [308].

Las principales técnicas o componentes `del Soft Computing` son:

LÓGICA DIFUSA O FUZZ LOGIC (FL) se basa en el análisis de información real no basada en verdades o falsedades absolutas, sino basadas en posiciones diferenciales contextualizadas [309] entre dos o más elementos.

COMPUTACIÓN EVOLUTIVA O EVOLUTIONARY COMPUTATION (EC) se inspira en los mecanismos de evolución biológica para resolver problemas de optimización combinatoria en el que el elevado número de combinaciones posibles hace imposible realizar una comprobación exhaustiva [88].

RAZONAMIENTO PROBABILÍSTICO O PROBABILISTIC REASONING O (PR) combina la teoría de probabilidad con la lógica deductiva para manejar la incertidumbre [194].

APRENDIZAJE AUTOMÁTICO O MACHINE LEARNING O (ML) busca la generalización de comportamientos a partir de información suministrada como ejemplos [26].

En el campo de la extracción de conocimiento, es en el `Machine Learning` donde recae la mayor importancia. El aprendizaje automático o `Machine Learning` es la disciplina científica encargada de crear sistemas que sean capaces de aprender [237]. En este contexto se define *aprender* como la capaz de identificar patrones complejos partiendo de datos de ejemplo introducidos al sistema. En función de las características que presenten estos datos de ejemplo se habla de tres tipos principales de aprendizaje: supervisado, no supervisado o por refuerzo.

APRENDIZAJE SUPERVISADO donde los algoritmos predicciones basadas en un conjunto de datos de ejemplos etiquetados, generalmente reflejando información histórica de un suceso. Se dice que este conjunto de ejemplo está etiquetado, cuando presentan la magnitud de la que se desea extraer el patrón de conocimiento. Por ejemplo, el número histórico de vehículos circulando por una calle puede usarse para realizar estimaciones aventuradas del número de vehículos que van a circular por dicha calle en el futuro.

APRENDIZAJE NO SUPERVISADO donde la información no tiene etiquetas asociadas a ellos. Esto implica que la magnitud sobre la que se quiere extraer conocimiento no está presente en los datos históricos de ejemplo. En su lugar, el objetivo del algoritmo es el de encontrar un patrón o regla que sirva para organizar los datos históricos, describiendo su estructura. Esto puede significar agruparlos en clústeres o buscar diferentes maneras de examinar los datos completos para que parezcan más simples u organizados.

APRENDIZAJE POR REFUERZO donde el sistema elige una acción en respuesta a cada uno de los datos históricos. Para cada estimación, se recibe una recompensa que indica como de buena fue la estimación realizada. Según esta información adicional, el algoritmo modifica su estrategia para lograr una mejor recompensa.

En función de la entrada y la salida que se desee obtener del sistema de aprendizaje se puede hablar de clasificación,agrupamiento, regresión o detección de anomalías:

CLASIFICACIÓN cuando los datos históricos se usan para predecir una variable categórica. Esta categoría refleja o resume un característica común conocida entre todos los datos. Cuando solo existen dos categorías, se habla de clasificación binominal. Cuando el número de categorías es mayor de dos, se conoce como clasificación multiclase.

AGRUPAMIENTO cuando los datos históricos se usan para particionar los datos en grupos que posean características o patrones similares entre si.

REGRESIÓN cuando el dato a predecir es un valor numérico acotado en un dominio y esperable en función de unos parámetros de entrada establecidos. Un tipo especial de regresión, es aquel empleado en estructuras basadas en series temporales, donde existe una dependencia entre el valor numérico y un marca de tiempo (Sección 5.11.1). En este tipo de regresiones se espera predecir el valor o valores futuros en base a los valores anteriores de la serie. Este tipo de regresión es denominado predicción de series temporales.

DETECCIÓN DE ANOMALÍAS cuando el objetivo de la predicción es la de identificar datos que no son habituales. Es decir, se busca aprender que es lo normal en los datos históricos para ser capaz de indentificar lo que sea significativamente diferente.

El `Machine Learning` está siendo cada vez más empleado en los estudios de movilidad, sobre todo en escenarios donde grandes cantidades de datos son obtenidos. La Tabla 5.36 recoge alguna de las problemáticas relativas a la movilidad, los métodos empleados para su resolución y ejemplos relevantes de la bibliografía.

[200] Trace analysis and mining for smart cities: issues, methods, and applications

[36] The scaling laws of human travel

[257] Limits of predictability in human mobility

[166] Clustering moving objects

[97] Trajectory pattern mining

[304] T-drive: Enhancing driving directions with taxi drivers' intelligence

[221] Using mobile phones to determine transportation modes

[311] iBAT: detecting anomalous taxi trajectories from GPS traces

[43] The geography of taste: analyzing cell-phone mobility and social events

[75] Inferring friendship network structure by using mobile phone data

[199] Land-use classification using taxi GPS traces

PROBLEMA	MÉTODO	EJEMPLOS
Reglas de Movilidad	Modelos estadísticos	Leyes de potencia para longitudes de pasos [36], modelos de movilidad continuos en el tiempo para deanvulaciones y regularidad de los movimientos basado en el análisis de la entropía o patrones de visita [257].
Comportamientos del movimiento	Agrupación	Búsqueda de trayectos similares [166] o detección de rutas frecuentes [97].
Recomendación de rutas	Ranking	Navegación [304] y guía en itinerarios.
Reconocimiento de actividades	Clasificación	Reconocimiento de comportamientos de transporte [221] o detección de actividades cotidianas y comportamientos anómalos [311].
Reconocimiento de eventos	Clasificación	Detección entre distintos eventos sociales [43]
Correlación en redes sociales	Regresión	Correlación entre amigos registrados y co-presencia [75]
Predicción en transporte	Regresión	Predicción del número de pasajeros, bicicletas públicas requeridas o el estado del tráfico [199].

Tabla 5.36
Problemática, métodos y ejemplos de aplicación del Softcomputing en el estudio de la Movilidad de personas y vehículos.
Fuente: Trace analysis and mining for smart cities: issues, methods, and applications [200]

A continuación se presentan los métodos de clasificación, regresión y detección de anomalías empleados en la tesis más destacables.

5.12.2 *Redes Neuronales artificiales: Perceptrón multicapa*

Una neurona biológica consta de un cuerpo celular compuesto por un núcleo que se ramifica en una rama principal (axón) y varias ramas secundarias (axones terminales) y que tiene uno o varios conectores de entrada (dentritas). La neuronas se combinan a través de los axones terminales y las dentritas, formando redes que propagan señales electroquímicas de los núcleos de una neurona a otra [277].

[277] Frank Rosenblatt: principles of neurodynamics: perceptrons and the theory of brain mechanisms

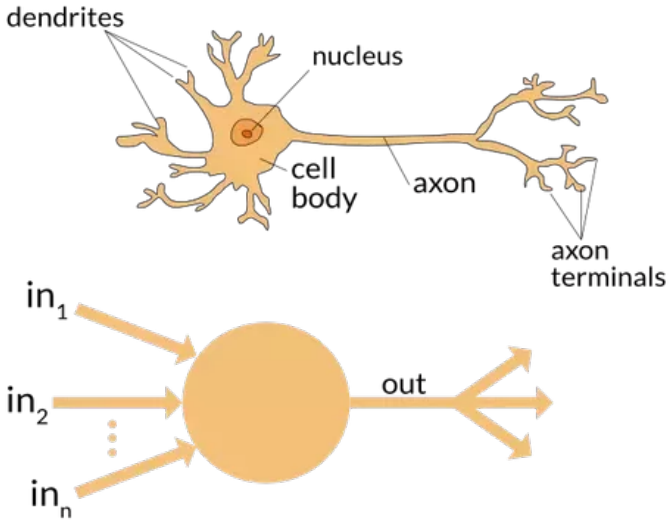


Figura 5.94
Neurona biológica y neurona artificial, compuestas por un núcleo, dentritas (input) y axones (outputs).
Fuente: Pattern matching with neural networks for the PANDA at FAIR experiment [10]

Una red neuronal artificial (`Artificial Neural Network` o `ANN`) imita el comportamiento de las neuronas biológicas de forma computacional [201]. Cada neurona está conectada con otras neuronas a través de los enlaces, formando una red. La salida de una neurona está conectada con la entrada de una o varias neuronas, modificada por un valor peso y/o acotada por un valor determinado por una función de activación.

[201] Electric load forecasting using an artificial neural network

El modelo más simple de neurona es el perceptrón [91], donde un número de dentritas (INPUTS) son combinadas linealmente a un axión (SALIDA), es decir, una serie de entradas determinan una única salida.

[91] Large margin classification using the perceptron algorithm

Un Perceptrón Multicapa [235] combina varios perceptrones formando capas de perceptrones conectadas de forma total o local con los perceptrones de las capas adyacentes. Esto es, las salida de las neuronas de la capa i, son las entradas de las neuronas de la capa $i+1$.

[235] Learning internal representations by error propagation

Las capas pueden clasificarse en tres tipos. La capa de entrada es aquella formada por las neuronas que introducen los valores de entrada en la red neuronal, y por tanto, no realizan ningún procesamiento. La capa de salida, es aquella que proporciona los valores de salida de la red. Por último, las capas contenidas entre la capa de entrada y salida se denominan capas ocultas, siendo la entrada de las neuronas la salida de la capa anterior.

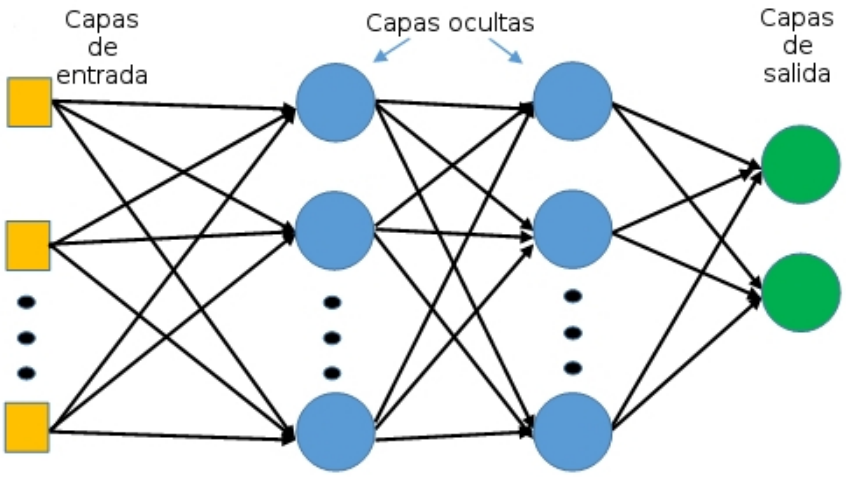


Figura 5.95
Capas de un perceptrón multicapa: la capa de entrada, la capa de salida y las capas ocultas. Fuente Original: Getting Started with TensorFlow [307]

Para el aprendizaje de la ANN se emplea una función de pérdida a optimizar, basada en la evaluación de la red en la resolución del problema. Para optimizar la función de pérdida, las redes neuronales artificiales se basan en el mecanismo de propagación hacia atrás o Backpropagation [294].

[294] Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems

De forma escueta, backpropagation realiza un sistema de dos fases en la propagación de los impulsos de la red neuronal. En la primera fase, para un estímulo de entrada en la red, se propaga desde la primera capa a la última, hasta generar una salida. La señal de salida se compara con la salida deseada o con el valor de la función de pérdida, calculándose una señal de error para cada una de las salidas.

Esta salida de error es propagada hacia atrás, partiendo de la capa de salida hacia todas las neuronas de las capas ocultas, aportando una contribución relativa a la aportación de la neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas hayan recibido su contribución relativa al error total [73]. El entrenamiento de la red, consistente por tanto en un aprendizaje supervisado basado en correción de error (o regla delta) realizando las dos fases descritas anteriormente.

[73] Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure

El campo de la redes neuronales artificiales es rico y se encuentra actualmente en auge gracias al Deep Learning [157], por lo que es campo de estudio muy amplio existiendo multitud de variantes, como las redes neuronales bayesianas (BNNs) [172] basadas en la formulación probabilista bayesiana o las redes de funciones de base radial (RBFNs, ver Sección 5.12.3) [39], que emplean una función de base radial como función de activación.

[157] Deep learning

[172] Bayesian interpolation

[39] Radial basis functions, multi-variable functional interpolation and adaptive networks

Este tipo de redes neuronales artificiales ha sido muy empleado para tareas de predicción de series temporales [21, 119, 123, 312], aportando como variables de entrada los valores autoregresivos de la serie para el valor a predecir. Por ejemplo, una serie temporal basada en horas, podría utilizar como entrada a la red las n horas previas al horizonte de predicción.

[21, 119, 123, 312] Neural network time series forecasting of financial markets, Large neural networks for electricity load forecasting: Are they overfitted?, The principles and practice of time series forecasting and business modelling using neural nets, Forecasting with artificial neural networks:: The state of the art

Las redes neuronales han sido aplicadas con éxito en las tareas de predicción relativas al tráfico y movilidad de personas y vehículos, tanto a la predicción del volumen de tráfico a largo [136] o corto plazo [1], así como en las magnitudes relativas a velocidades y tiempos de desplazamientos [133].

[136] Dynamic wavelet neural network model for traffic flow forecasting

[1] Short-term traffic flow prediction using neuro-genetic algorithms

[133] Short-term prediction of travel time using neural networks on an interurban highway

El funcionamiento de las redes neuronales artificiales para predicción recuerda en gran medida a los métodos ARIMA (Sección 5.11.1.3), por lo que son numerosos los autores que han realizado comparativas entre ambos métodos [16], si bien los estudiosos más puristas, consideran los métodos basados en `ANNs` demasiado opacos y con poca capacidad de interpretación. Sin embargo, la tendencia futura pasa por el empleo de ambas técnicas combinadas [77, 169].

[16] Comparison of ARIMA, neural networks and hybrid models in time series: tourist arrival forecasting

[77, 169] Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA, Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction

5.12.3 *L-Co-R*

`Lags COevolving with Rbfns` [202] es un algoritmo que emplea redes neuronales de base radial [80] autogeneradas empleando un algoritmo CoEvolutivo [54] con dos poblaciones: la población de `RBFNs` y la población de `lags`.

[202] Coevolution of lags and rbfns for time series forecasting: L-co-r algorithm

[80] Self-generation RBFNs using evolutional PSO learning

[54] A coevolutionary multi-objective evolutionary algorithm

Los `lags` cumplen la misma función que componente autoregresivo de los métodos de predicción clásicos, indicando los valores previos de la serie que serán empleados para la predicción. Sin embargo, este componente autoregresivo se ve ampliado, ya que no impone que los valores empleados tengan que ser los n predecesores al horizonte, sino que pueden ser una selección no correlativa de valores previos[104].

Este método ofrece muy buenos resultados para la predicción con intervalos de tiempo cortos, medios y largos [203] pero debido al proceso evolutivo no alcanza una solución determinista, ya que depende de las `RBFNs` evolucionadas. Sin embargo, al trabajar con poblaciones de `RBFNs` y `LAGs`, cada par de individuos de cada población es una solución, por lo que es un modelo de predicción válido.

[203] Short, medium and long term forecasting of time series using the L-Co-R algorithm

5.12.4 *SMOreg: Support Vector Machine for Regression*

Las máquinas de soporte vectorial (`Support Vector Machines` o `SVMs`) [280] son un conjunto de algoritmos de aprendizaje supervisado que partiendo de un conjunto de muestras posicionadas en un espacio multidimensional[105] encuentra los hiperplanos que separan los espacios de cada par de clases

[280] Support vector machines

104 ↑Por ejemplo, una serie temporal basada en intervalos de hora, un componente autoregresivo de 2 empleará los valores de las 2 horas anteriores para realizar la predicción. El `lag`, en cambio, puede ser establecido para que use el valor de la hora anterior, y el valor de la misma hora del horizonte de predicción del día anterior, es decir los $lags(1, 24)$. El número de entradas o `INPUTS` de ambos modelos es dos, pero hacen uso de valores previos distintos de la serie. De esta forma, cabe preguntarse, si para predecir que ocurrirá a las 10 de la mañana, es más relevante lo que haya ocurrido a las 8 y 9 de ese mismo día, o lo que ya haya ocurrido a las 9 de ese día más lo que ocurriese el día anterior a las 10.

105 ↑Una dimensión por cada característica de la muestra.

mediante el vector originado por los dos puntos más próximos para cada dos clases distintas (denominado vector soporte) determinando el conjunto de hiperplanos que dividen el espacio multidimensional entre las distintas clases determinadas para la muestra.

Si bien la manera más sencilla de determinar la separación entre las clases de la muestra es la del empleo de una línea recta (en espacios bidimensionales), un plano recto (en espacios tridimensionales) o un hiperplano n-dimensional (en espacios n-dimensiones), esta representación resulta insuficiente en el mundo real [63]. Es por ello que se emplea una función kernel que aumentan la dimensionalidad del espacio proyectando la información de la muestra a un nuevo espacio de mayor dimensiones (Ecuación 5.6).

[63] Support-vector networks

$$\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, x \mapsto \phi(x), d_1 < d_2 \tag{5.6}$$

Numerosas funciones de kernel han sido empleadas en SVM siendo las más empleadas la función polinomial-homogénea[106], el perceptrón[107], la función de base radial Gaussiana[108] o la función Sigmoid [109].

Si bien originariamente SVMs se ideo para la clasificación el método fue ampliado para regresión (Support Vector Regression o SVR) [74], siendo el método basado en mínimos cuadrados (Leats-Squares Support Vector Machine o LS-SVM) [262] el más extendido.

[74] Support vector regression machines

[262] Least squares support vector machine classifiers

Este método busca entrenar [249] la máquina de vector soporte para minimizar $\frac{1}{2}\|\omega\|^2$ para el conjunto de datos etiquetados de entrenamiento, impuestas las siguientes restricciones:

[249] A tutorial on support vector regression

$$\begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon, \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon, \end{cases} \tag{5.7}$$

Donde x_i es el punto i de la muestra de entrenamiento, y_i el valor del punto i, el producto escalar $\langle \omega, x_i \rangle + b$ es la predicción para la muestra y ε es el umbral definido para la predicción. La Figura **??** presenta un ejemplo unidimensional, para facilitar la comprensión del fundamento de SVR.

106 $\uparrow K(x_i, x_j) = \left(x_i \times x_j\right)^n$

107 $\uparrow K(x_i, x_j) = \left\|x_i - x_j\right\|^n$

108 $\uparrow K(x_i, x_j) = e^{\frac{-\left(x_i x_j\right)^2}{2\times\sigma^2}}$

109 $\uparrow K(x_i, x_j) = tanh\left(x_i \times x_j - \theta\right)$

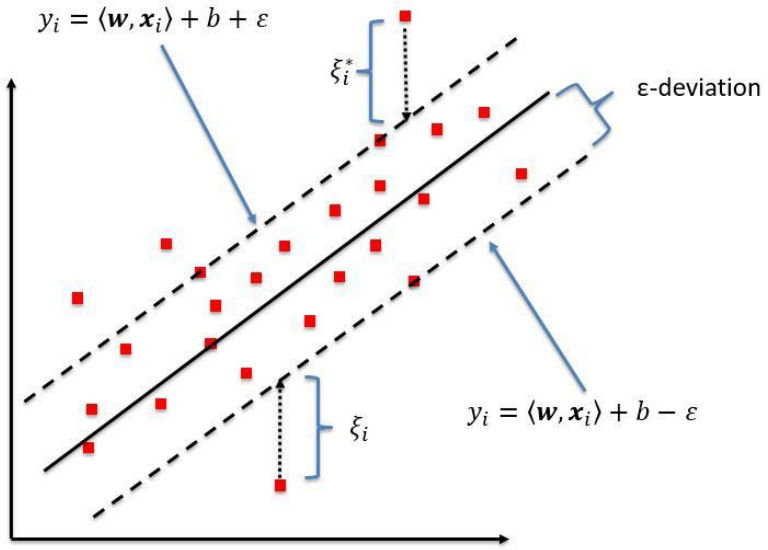


Figura 5.96
Ejemplo de regresión con SVM en un espacio unidimensional. Los puntos rojos representan los datos del conjunto de entrenamiento, siendo los valores X los relativos a las características (una única característica en este caso, al ser unidimensional) y el eje Y la magnitud de regresión. Mediante SVM se busca el hiperplano (recta en este caso) que minimiza el error de los puntos no acotados por los hiperplanos ($y_i = \langle \omega, x_i \rangle + b \pm \varepsilon$). Aunque sea un entorno unidimensional y el hiperplano es una recta, el concepto subyacente para espacios multidimensionales ampliados por funciones kernel es el mismo.
Fuente: Indresh Bhattacharyya
https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff

La regresión por SVM se trata por tanto de un problema de minimización funcional, por lo que se suele emplear en conjunción a algoritmos de minimización, siendo el más extendido la Optimización minima Secuencial (Sequential minimal optimization o SMO) [211].

[211] Sequential minimal optimization: A fast algorithm for training support vector machines

SMO, de forma simplificada[110], es un algoritmo iterativo que divide un problema de minimización con restricciones en pequeños subproblemas, empleando multiplicadores de Lagrange para incorporar las restricciones. Para ello, encuentra un multiplicador de Lagrange que viole las condiciones de Karush-Kuhn-Tucker (KKT)[299][111], escoge un segundo multiplicador y optimiza el par formado[112] por ambos multiplicadores. El método finaliza cuando todos los multiplicadores de Lagrange satisfacen las condiciones KKT o el algoritmo no sigue convergendiendo, es decir, las variaciones del proceso iterativo de minimización no implican variaciones por debajo de un umbral determinado.

[299] The Karush–Kuhn–Tucker optimality conditions in an optimization problem with interval-valued objective function

El valor de regresión de un nuevo valor a predecir para un nuevo conjunto de características viene determinado por su intersección con el hiperplano determinado por SVM y cuyo error ha sido minimizado por SMO.

110 ↑Nuevamente, porque los pormenores de este algoritmo escapan al ámbito de esta tesis. Si se desean más detalles del algoritmo tanto en http://cs229.stanford.edu/materials/smo.pdf como en https://www.youtube.com/watch?v=I73oALP7iWA se encuentra material docente sobre el mismo.
111 ↑Nuevamente, esto escapa al ámbito de esta tesis.
112 ↑Lo cual resulta se resuelve de forma analítica.

En una serie temporal, la principal característica del conjunto de datos de entrenamiento es el instante o marca de tiempo y se podría considerar que esta información es la única disponible. Sin embargo, un preprocesamiento de los datos eficiente, puede convertir un conjunto de datos en el que la única característica disponible sea la marca de tiempo, en un conjunto de datos multidimensional.

Para ello, se extrae información de la fecha y se añaden características al instante tomado. Así de una simple fecha concreta, se puede obtener el año, el mes, el día del mes, el día de la semana, la hora o los minutos, y constituir una característica de la muestra con cada uno de estos valores. Además, se puede incorporar como variables exógenas si un día era festivo, las inclemencias meteorológicas, si había algún tipo de evento, si el sitio estaba abierto o cerrado...

Debido a que `SVM` se basa en gran medida en el cálculo de distancias[113], esto permite poner a muestras cuyas características sean muy similares, y es esperable que tendrán patrones de comportamiento parecidos, de forma muy cercana aunque se encuentren alejadas a lo largo de la línea de tiempo. Ejemplificando esto, para determinar el número de personas un lunes a las 9:30 de la mañana, es posible que sea más relevante los valores de los anteriores lunes a las 9:30 de la mañana, que los valores del día anterior a las 16:20 de la tarde, aunque ese valor se encuentre en la línea tiempo más cercano que cualquier otro lunes a las 9:30. Este proceso se resume en la Figura 5.97.

Marca de tiempo	Valor
...	...
2019-04-21 04:25:00	201
2019-04-21 04:30:00	180
2019-04-21 04:35:00	274
...	...

→

t	Año	Mes	Día Mes	Día Semana	Semana	Hora	Minuto	...	Valor
...	...	...	...	...	...	...	...	...	...
1555820700	2019	4	21	Domingo	17	04	25	...	201
1555821000	2019	4	21	Domingo	17	04	30	...	180
1555821300	2019	4	21	Domingo	17	04	35	...	274
...	...	...	...	...	...	...	...	...	...

Figura 5.97
Preprocesamiento de series temporales para SVM, donde se convierte en el conjunto de datos unidimensional etiquetado, en un conjunto multidimensional que permite que extraer mejor los patrones de comportamiento de los distintos tipos de conjuntos de datos.

Debido a esta característica, los métodos basados en máquinas de soporte vectorial han resultado muy precisos en la predicción de magnitudes relativas al tráfico y movilidad como el flujo de vehículos [313] o la velocidad de estos [278]. Además, permiten trabajar con ventanas de tiempo más pequeñas sin influir negativamente en su rendimiento [315].

[313] Traffic forecasting using least squares support vector machines

[278] A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed

[315] Short-term traffic flow prediction method based on SVM [J]

113 ↑Tanto para la parte de establecer los vectores soportes, como en el caso de la predicción, para determinar la minimización del error de la distancia a los hiperplanos.

5.12.5 *Detección de anomalías*

En un conjunto de datos se denomina anomalía al dato puntual que no presenta un comportamiento normal o esperable [49]. En series temporales, cada valor de la serie está relacionado con los valores vecinos, debido a su orden temporal. Las series temporales se ven influenciadas por valores cíciclos, como se ha presentado en la Sección 5.11.1. Estos ciclos constituyen patrones de comportamiento periódico [93].

[49] Anomaly detection: A survey

[93] Introduction to statistical time series

Las series temporales que presentan patrones cíclicos, se descomponen en diferentes componentes (Sección 5.11.1.1). El componente irregularidad es aquel que no puede ser explicado por los otros componentes de la serie debido a que no obedece a un patrón de comportamiento modelizable por la serie [5].

[5] Outlier analysis

La detección de anomalías se centra en el estudio de este componente irregular, que estropea o rompe el patrón modelizado de la serie temporal. Debido a que se requiere un modelo, la primera parte del estudio de anomalías implica el empleo de los métodos de predicción de series temporales (Sección 5.12.1) para determinar que es normal o esperable [89]. Si ocurre un fallo del modelo en una única observación, esa observación se considera anomalía puntual. En el caso de que la anomalía se presente en una sucesión continua de puntos, se habla de una anomalía de forma [4]. La anomalía se denomina global si se muestra por encima (o por debajo) del valor esperado por el patrón cíclico de la serie y local en el caso de que se muestre unicamente por encima (o por debajo) de los valores colindantes [242]. Un umbral estático determinado, como por ejemplo determinar anómalo todo aquello que se sitúe por encima de un valor determinado (como un percentil o cuartil) solo puede detectar anomalías globales.

[89] Outliers in time series

[4] Data mining: the textbook

[242] Anomaly Detection for Monitoring

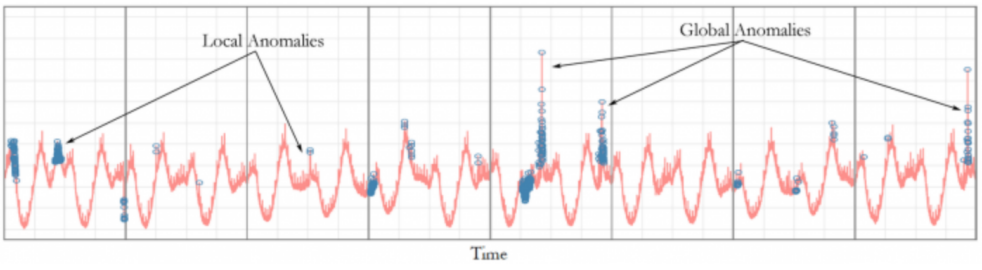


Figura 5.98
Ejemplos de anomalías locales y globales en series temporales.
Introducting practical and robust anomaly detection in time series [143].

El problema de detección de anomalías en series temporales estacionarias ha sido estudiado recientemente por su implicaciones en las redes sociales, como Twitter. En Twitter se puede considerar una serie temporal al número de mensajes o tweets publicados sobre un tema determinado a lo largo del tiempo. Debido a esto y el sistema de `trending topic`, Twiiter ha desarrolla-

do su propio algoritmo de detección de anomalías en series temporales[114]. Este algoritmo se denomina `Seasonal Hybrid ESD (S-H-ESD)` [6, 143], y se construye sobre el test `Generalized ESD` [231].

[6, 143] Unsupervised real-time anomaly detection for streaming data, Twitter Engineering: Introducing practical and robust anomaly detection in a time series

[231] Percentage points for a generalized ESD many-outlier procedure

`S-H-ESD` descompone la serie en componentes y emplea estadísticos robustos [124] en conjunción con `ESD`. Además, para series temporales largas, realiza una aproximación por partes. Esto se debe al hecho de que la extracción de la componente de tendencias en presencia de anomalías no es trivial para la detección de anomalías, esto es, la tendencia de la serie completa puede verse afectada por los valores anómalos a detectar. Realizar una aproximación por partes permite minimizar el efecto en la tendencia de los valores anómalos [276].

[124] Robust statistics

[276] A Novel Technique for Long-Term Anomaly Detection in the Cloud.

Los detalles de implementación del método [281] son irrelevantes para el ámbito de esta tesis, debido a que el método se encuentra disponible implementado en la librería de R `AnomalyDetection`[115].

[281] An Enhanced Seasonal-Hybrid ESD Technique for Robust Anomaly Detection on Time Series

Sin embargo, este método presentaba ciertas limitaciones, como por ejemplo lidiando con series temporales con intervalos de agrupamiento pequeños y con series temporales de varios meses. Estas carencias se han intentado salvar en la medida de lo posible en la implementación que se ha incorporado a la librería Mobywit de R (Sección 5.11.3.1).

114 ↑https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-html

115 ↑https://github.com/twitter/AnomalyDetection

5.13 SISTEMA DE ALMACENAMIENTO EN LA NUBE

Si bien se hace uso de una versión muy optimizada de MySQL para el almacenamiento y computo local frente a alternativas basadas en modelos de base de datos NoSQL muchos más empleadas en entornos de BigData, como se ha presentado en la Sección 5.9.2 no se puede negar que su eficiencia para algunos escenarios es muy superior a las ofrecidas por un sistema SQL como el presentado en la Sección 5.9.

Con las fortaleza de los sistemas NoSQL presentes, se decide emplear un gestor de base de datos distinto a MySQL para la publicación de los datos finales del análisis (Sección 5.11) y aprendizaje (Sección 5.12), así como todas las tareas de difusión y publicación que serán presentadas en la Sección 5.14.

De esta manera, se propone un sistema híbrido, con el cómputo local realizándose en la base de datos local MySQL muy optimizada para el cómputo, para posteriormete almacenar esos resultados procesados en un sistema NoSQL alojado en la Nube, volviéndolos publicamente accesibles.

Existen una gran cantidad de alternativas noSQL con almacenamiento en la nube, sin embargo para el prototipo se ha decantado por emplear una solución muy poco conocida ofertada por Google: Fusion Tables [101, 102].

[101, 102] Google fusion tables: web-centered data management and collaboration, Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud

5.13.1 *Google Fusion Tables*

Google Fusion Tables es un servicio basado en la nube para la administración e integración de datos tabulares potencialmente geolocalizados, el cual provee de diferentes maneras de visualizar, filtrar y procesar los datos almacenados. Soporta integración de datos de múltiples fuentes realizando `JOINS` cruzando tablas que pueden pertenecer a usuarios distintos. Este `JOIN` constituye una vista de datos cruzamos, no un mecanismo de selección en consultas.

Los usuarios pueden mantener sus datos privados, compartidos con un conjunto de colaboradores o hacer los datos públicos accesibles para cualquier motor de búsqueda.

Google Fusion Tables ofrece también una `API REST` para administración de las tablas, ventanas de información de las plantillas y estilos de visualización. Ofrece también un sistema de consultas para gestionar las tuplas (insercción, actualización o eliminación) así como realizar selecciones de tuplas basadas en condiciones de datos o geográficos. Dicha consulta puede ser devuelta en CSV, JSON o ser usada de forma nativa por otros entornos de *Google* como *Google Maps* o *Google Chat Tools*. Esta funcionalidad permite una rápida y transparente representación y publicación de los datos.

5.13.1.1 *Beneficios de uso*

A continuación se recoge a modo de sumario algunas de las características y ventajas que justifican el uso de esta plataforma para el almacenamiento en la nube implementado en el prototipo.

USO DE DATOS PÚBLICOS: Fusion Tables permite la búsqueda entre miles de tablas alojadas en Fusion Tables, o entre millones de tablas existentes en la web que se pueden importar en Fusion Tables. De igual manera, si se desea, se puede hacer que todos los datos alojados en Fusion Tables sean accesibles de forma pública.

IMPORTAR TUS PROPIOS DATOS: Ya sea mediante la subida de un fichero CSV, una hoja de cálculo o un fichero KML, o bien mediante el uso de la `API REST` que permite insertar, actualizar, eliminar y realizar consultas de forma programada, con Fusion Tables se puede subir los datos de forma rápida y sencilla a la nube.

VISUALIZAR DE FORMA INSTANTÁNEA: De forma inmediata, una vez los datos han sido importados, se pueden visualizar los datos en gráficos o establecer una geolocalización en un mapa. Además, se pueden realizar filtros para visualizaciones más selectivas.

PUBLICACIÓN INMEDIATA: Cualquier gráfico o mapa realizado puede ser embebido vía WEB o enviado vía correo electrónico, mostrando siempre los valores actualizados de la base de datos, sin tener que realizar ningún trámite adicional.

COTEJAR DATOS CON OTRAS BASES DE DATOS: Si alguien tiene de datos diferentes sobre la misma entidad de datos o si se dispone los datos en distintas tablas, con Fusion Tables se puede cotejar los datos, haciendo un JOIN transparente entre ambas tablas, lo que generará una nueva vista.

SIEMPRE ACTUALIZADOS: Cuando alguna de las bases de datos es actualizada (ya sea modificación o inserción de nuevos datos) las tablas cotejadas se actualizan de forma automática y transparente.

COMPARTIR SÓLO LOS DATOS DESEADOS: Si se necesita mantener alguno de los datos en privado, se permite compartir sólo un conjunto de columnas de la tabla principal que estarán siempre actualizadas respecto a la tabla original, pero con sus propios permisos de publicación.

CONSTANTE AUTORÍA DE LOS DATOS: Fusion Tables permite mantener de forma constante de donde provienen los datos y a quien pertenecen. Tanto durante la importación y la visualización se muestran la autoría de los datos. Incluso si las tablas son mezcladas, se seguirá manteniendo la autoría de los datos originales, y así será mostrado.

GEOLOCALIZACIÓN TRANSPARENTE DE LOS DATOS: Ya sean datos expresados en puntos, líneas, polígonos, direcciones, nombre de lugares,

países o cualquier registro susceptible de almacenar datos geográficos, es automáticamente interpretado y geolocalizado en un mapa.

PERSONALIZACIÓN DE LOS MAPAS: Aplicar colores o iconos basados en los datos. Hacer mapas de intensidades basadas en zonas. Usar polígonoss KML para el dibujado sobre el mapa. Mostrar miles de trazas al mismo tiempo.

COMPARTICIÓN DE MAPAS: Los mapas generados con Fusion Tables pueden ser embebidos en cualquier WEB, enviados mediante correo electrónico, ser salvado como un fichero KML para ser visto en cualquier software de geolocalización, o incluso usar Fusion Tables para la generación dinámica de KML ofrecidos como enlaces para mantener siempre un enlace al mapa constantemente actualizado.

ALTA ACCESIBILIDAD WEB: Fusion tables permite almacenar los datos y ofrecer un portal donde los usuarios puedan visualizar los datos sin necesidad de descargarlos. Pueden explorar los mapas, gráficos, realizar cálculos sobre ellos o hacer búsquedas y filtros sobre los datos para su posterior descarga. O si se desea, se puede impedir la descarga de los datos, permitiendo sólo su visualización ONLINE.

DIFUSIÓN DE LOS DATOS ACTUALIZADOS Y DE FORMA SEGURA: En lugar de mantener miles de copias de los datos en diversos discos duros locales, los datos en Fusion Tables están siempre actualizados y seguros, mostrando siempre los últimos datos en todas las gráficas y mapas que han sido generadas de forma automática.

POSIBILIDADES DE EXPANSIÓN MEDIANTE LA API: Cuando los datos se almacenan en Fusion Tables, se ofrece de forma automática una API para que los desarrolladores puedan acceder a los datos mediante REST. Ofertando tanto una forma pública (basa en token) como un sistema autenticado basado en oAuth. Únicamente mediante el sistema autenticado se podrá alterar los datos (modificar, eliminar, insertar,..)

5.13.1.2 *Arquitectura de Fusion Tables*

Se hace necesario explicar, aunque sea de forma breve, la arquitectura de Fusion Tables y el sistema de almacenamiento de los datos subyacente, con el fin de poder comprender porque resulta beneficioso su empleo para almacenar resultados pero no para el almacenamiento local de los datos.

En la Figura 5.99 se muentran los principales componentes de la arquitectura del servicio Fusion Tables.

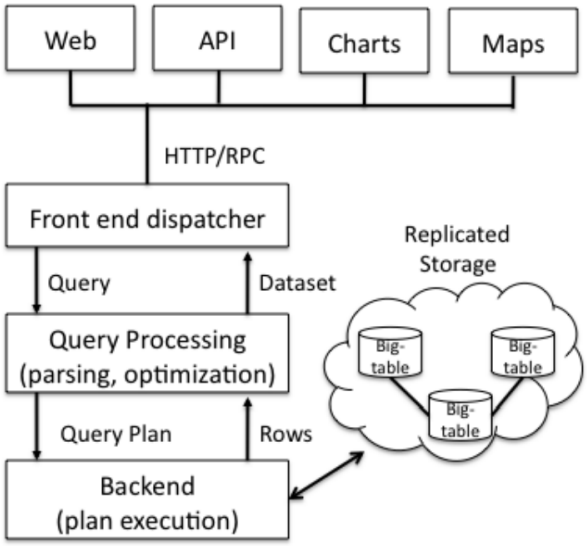


Figura 5.99
Arquitectura de almacenamiento de Google Fusion Tables.
Fuente: Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud [102].

Las peticiones al servicio pueden ser originadas desde distintas fuentes:

- La vista web de Google Fusion Tables
- La API REST de comunicación
- Visualizaciones embebidas soportadas de forma nativa
  - Un mapa de Google Maps
  - Un gráfico de Google Chart

Todas son procesadas igualmente, con la salvedad de las peticiones sobre mapas, que se generan mediante consultas espacio/estructurales planificadas sobre las tablas del sistema. Es decir, se utilizan las coordenadas de visualización del mapa para la obtención de imágenes estáticas que son geolocalizadas en un mapa con la información resultante sobreespuesta.

En el resto de casos, el FrontEnd convierte las peticiones (`Query`) de las distintas fuentes a una representación común, y lo envía al módulo de procesamiento de consultas (`Query Processing Module`) que genera una planificación para la consulta (`Query Plan`).

Dicho plan es ejecutado por el `Back End` haciendo uso de un conjunto síncrono y replicado de servidores de almacenamiento `Big Table` denominado pila de almacenamiento (`Storage Stack`) sobre los que se construyen las tablas de Fusion Tables empleando el motor `Bigtable` y una capa de abstracción auxiliar `Megastore` por encima de ella.

5.13.1.3 *Fundamentos del almacenamiento `Big Table`*

Las tuplas almacenadas dentro de una estructura `Big Table` son de tipo {clave,valor}, las cuales se encuentran ordenadas en función de la clave y distribuidos por rangos (de la propia clave) en múltiples servidores. El valor, puede ser cualquier estructura compleja formada por valores atómicos permitidos.

El entorno BigTable provee de un única operación de escritura que inserta una tupla de forma atómica, esto es, en tiempo constante y sin posibilidad de interrupción.

Se provee también de tres mecanismos de lectura:

- Búsqueda por clave (`Lookup by key`) donde para una clave identificada se devuelve el valor asociado a dicha clave.
- Búsqueda por prefijo de clave (`Lookup by key prefix`) donde se devuelve los valores para los cuales el comienzo de su clave comienza con el prefijo buscado.
- Búsqueda por rango de clave (`Lookup by key range`) donde se devuelve los valores comprendidos entre las dos claves especificadas.

Además, el entorno BigTable almacena un sello de tiempo histórico (timestamp) para cada tupla. Así se podría entender una tupla como el conjunto {clave,valor,marcaDeTiempo}, siendo la marca de tiempo el instante en el que ha sido escrito. Esto es debido a que el sistema permite almacenar tuplas con claves coincidentes.

`Megastore` es un conjunto de librerías situadas por encima de `BigTable`. En ellas se provee de primitivas de alto nivel, como índices secundarios (Seccion 5.13.1.4), transiciones sobre múltiples tuplas (Sección **??**) y replicaciones entre diversos servidores conservando la consistencia.

A continuación, se describe brevemente como se almacenan los distintos coponentes habituales de una base de datos en el motor `Big Table`.

Almacenamiento de tuplas (`Row Store`)

En Fusion Tables, todas las tablas de un usuario son almacenadas en una única tabla BigTable de nombre *Rows* (filas o tuplas). Cada tupla dentro de dicha tabla pertenece a una tupla en una tabla concreta del usuario. La clave empleada para la tupla, es la concatenación del identificador de la tabla y el identificador de la fila. Dicho identificador de fila es gestionada de forma interna y transparente para el usuario, que no necesita definir una clave primaria para su tabla.

Almacenamiento de esquemas (Schema Store)

Para almacenar los esquemas de las tablas se hace uso de otra tabla BigTable, con una tupla por cada tabla almacenada en el sistema. El valor de la clave, es el identificador de la tabla.

En el valor, se almacenan las columnas y permisos para cada tabla. De cada columna se almacena el nombre y el tipo de dato preferente. En los permisos, se almacenan los usuarios asociados a sus diversos permisos sobre la tabla. Las tablas públicas constante de un registro especial indicando que son visibles por todo el mundo.

Con este sistema, se permite que el esquema de una tabla pueda evolucionar con el tiempo, pudiendo los usuarios añadir y eliminar columnas sobre sus datos.

Almacenamiento de vistas (View Store)

Una de las principales funcionalidades de Fusion Tables es que permite que múltiples usuarios puedan cotejar sus datos "uniendo" sus tablas en una sola, incluso si dichas tablas pertenecen a usuarios distintos o incluso a usuarios terceros. Una tabla construida mediante uniones de tablas se denomina una vista, y dispone de un almacenamiento virtual. Esto es, las vistas no son almacenadas en la BigTable Rows de ninguno de los usuarios, solo se almacena su definición y permisos en la BigTable Schema. Las vistas tienen sus propios permisos, de igual manera que las tablas de usuarios.

Almacenamiento de comentarios (Comment Store)

Para facilitar las colaboraciones, Fusion Tables permite los comentarios tanto de tablas, tuplas, columnas o celdas individuales. Todos los comentarios son almacenados en una tabla BigTable, donde la clave es el elemento comentado y el valor es el comentario.

5.13.1.4 *Procesado de consultas en Fusion Tables*

Fusion Tables se encuentra limitado a las primitivas de lectura que ofrece BigTable para dar soporte a un conjunto muy limitado de consultas SQL. Actualmente el sistema soporta selecciones (SELECT) sobre los valores de cualquier columna; así como agregaciones (GROUP BY) y uniones (JOINS) basados únicamente en la clave principal.

La estrategia de ejecución de consultas se basa en traducir la consulta a procesar en un conjunto de las tres operaciones soportadas por las tablas BigTable presentadas anteriormente (Lookup by key),Lookup by key prefix y Lookup by key range).

Por tanto la posibilidad de ejecutar consultas elaboradas como las presentadas en la Sección 5.10 en un motor Big Table de forma eficiente son muy limitadas. Esto es debido a que estos motoroscarecen de medios adicionales

de optimización como los presentados en la Sección 5.9.6 para el almacenamiento local. Estos motores se basan principalmente en la recuperación de información mediante únicamente su clave, o mediante la marca de tiempo impuesta para la tupla. Sin embargo, el sistema presentado, como se ha expecificado en la Sección 5.1 requiere de al menos dos marcas de tiempo (la de primera y última detección). Además, estas marcas no son constituidas durante el instante de inserción de la tupla en la base de datos, sino por el *nodo* en el momento de la detección.

5.13.2 *Sistema de almacenamiento en la nube: `Ezequiel`*

Una vez presentada de forma breve la arquitectura en la que se basa el servicio de almacenamiento en la nube que se ha elegido, se presenta el sistema encargado de la gestión del almacenamiento en la nube, que consta de un software desarrollado denominado *Ezequiel*, encargado de la sincronización entre la base de datos del almacenamiento local y las bases de datos en la nube. Ezquiel se encuentra desarrollado en JAVA, emplando la librería `Fusion Table API Client Library for Java`[116] proporcionada por Google.

La funcionalidad de la libreria se extiende para obedecer a las necesidades concretas del proyecto, así como una mejora en la eficiencia mediante el procesado por lotes de las peticiones de inserción (Sección 5.13.2.2) o la incorporación de mecanismos abanzados no disponibles de forma nativa, como la directiva `ON DUPLICATE KEY UPDATE` (Anexo A.5). Se abordan también los principios de autorización y autenticación que garantizan la correcta seguridad de los datos almancenados. Finalmente, se presentan a modo de ejemplo, alguno de los módulos implementandos en el sistema que hacen uso de los métodos presentados en la Sección 5.10 para la generación tanto de pasos (Sección 5.1.3) y trazas (Sección 5.1.6) de dispositivos.

5.13.2.1 *Autorización*

Si bien el acceso de lectura de los datos puede ser público, la escritura y modificación de los datos de Google Fusion Tables requiere de mecanismo de autenticación que identifique a un usuario con permisos de escritura frente a un usuario sin ningún tipo de privilegios.

Google hace uso del popular servicio de autenticación OAuth[117] que permite la autorización de una API de modo estándar y simple. La gestión del modo de autenticación requiere el almacenaje y envío de credenciales de usuarios, basadas en un sistema de clave pública / privada [72][71].

Se realiza la autorización del sistema mediante la función recogida en el Código 5.69. Esa función es mostrada a modo de ejemplo, y resulta trascendete ya que la autenticación en primera instancia requiere del acceso mediante

116 ↑`https://developers.google.com/api-client-library/java/apis/fusiontables/v2`
117 ↑`https://developers.google.com/accounts/docs/OAuth2`
`https://developers.google.com/fusiontables/docs/v1/using#auth`

un navegador a la URL mostrada en la línea 10 de dicho código. Este acceso identifica la máquina donde se realizará el acceso a la API REST de `Fusion Tables`.

Código 5.69
Ezequiel: Gestión de credenciales de conexión a Google Fusion Tables

```
1  private Credential authorize() throws Exception {
2    FileInputStream _f = new FileInputStream(DATA_STORE_FILE);
3
4    GoogleClientSecrets clientSecrets = GoogleClientSecrets.load(
5        JSON_FACTORY,
6        new InputStreamReader(_f));
7
8    if (clientSecrets.getDetails().getClientId().startsWith("Enter") ||
     ↪  clientSecrets.getDetails().getClientSecret().startsWith("Enter "))
     ↪  {
9      Logger.getGlobal().log(Level.SEVERE,
10         "Enter Client ID and Secret from
           ↪  https://code.google.com/apis/console/?api=fusiontables "
11         + "into
           ↪  fusiontables-cmdline-sample/src/main/resources/client_secrets.json");
12     System.exit(1);
13   }
14
15   GoogleAuthorizationCodeFlow flow = new
     ↪  GoogleAuthorizationCodeFlow.Builder(
16       httpTransport, JSON_FACTORY, clientSecrets,
17
         ↪  Collections.singleton(FusiontablesScopes.FUSIONTABLES)).setDataStoreFactory(
18         dataStoreFactory).build();
19   return new AuthorizationCodeInstalledApp(flow, new
     ↪  LocalServerReceiver()).authorize("user");
20 }
```

Dicho código hace uso de una constante `DATA_STORE_FILE` que es inicializada en el constructor de la clase para abrir el fichero de credenciales, que debe ser generado mediante la herramienta `Google Cloud Console`, panel de administración de todos los aspectos relacionados con cualquier servicio de Google, como `Fusion Table`.

5.13.2.2 *Optimización de procesado: Sistema de colas*

La utilización de una API para la subida de datos en un servicio en la nube corre el riesgo de sobrepasar la cuota máxima de la que se disponga, que suele limitar tanto el número de peticiones por segundo como el máximo de peticiones diarias[118].

118 ↑Aunque en el desarrollo del prototipo en esta tesis, gracias al Feedback con Google en la herramienta, se logró disponer de una couta de uso muy elevada a coste cero, al indicarles que se trata de un proyecto de investigación.

Para gestionar de forma controladas las peticiones a la API REST desde los distintos módulos del sistema *EZEQUIEL* se implementa un mecanismo que serializa todas las peticiones de inserción a través de un único subsistema. De esta forma, todas las insercciones del software son realizadas de forma ordenada y paulatina por un único elemento.

En el código A.6 se presenta el subsistema encargado de la gestión de las colas de inserción en la nube. Es importante entender que este subsistema también es el encargado de reincidir en las peticiones que hayan sido rechazadas por exceso de alguna de las cuotas. Es por ello que mantiene siempre el órden de aparición de las distintas peticiones que se hacen al sistema de almacenamiento en la nube.

Sin embargo, es necesario notar que este mecanismo se emplea solamente para las inserciones (o actualizaciones) de nuevos datos. Las consultas a la base de datos en la nube son gestionadas fuera del sistema *EZEQUIEL*, como por ejemplo, en los mecanismos de difusión y publicación que son presentados en la Sección 5.14.1.

5.13.2.3 *Actualización de Nodos*

Este subsistema se encarga de actualizar la información relativa a los nodos en la nube. Almacena las coordenadas del nodo, así como una representación gráfica mediante un poligono KML[119] que permite una representación incluyendo el rango de acción aproximado del nodo. Esta funcionalidad se presenta en el Código A.7.

5.13.2.4 *Establecimiento fecha última actualización*

Debido a que la base de datos incrementa su tamaño de forma constante, es necesario disponer de un criterio que permita determinar que datos ya han sido sincronizados con la base de datos en la nube y cuales están pendientes. Para ello se provee la función que se muestra en el Código A.8 que se encarga de localizar la fecha a la cual pertenecen los últimos datos temporales alojados en la tabla de Fusion Tables. Este valor puede ser calculado, o puede ser indicado de forma manual, por ejemplo en el caso de realizar una primera sincronización, o ante errores que requieran que se compruebe la consistencia de datos pasados.

Debido a que los nodos pueden perder la comunicación con el servidor, y para paliar posibles desfases, se amplia la ventana de comprobación de los últimos valores susbidos. En el código A.8 ha sido establecido a 2 horas, esto implica que si los últimos valores introducidos pertenecen a las 10:00 de un día concreto, el sistema realizará una comprobación de la consistencia de los valores de hasta dos horas antes, esto es, desde las 08:00 de ese mismo día. Este valor, puede ser ajustado en base al tiempo de sincronización que se haya

119 ↑https://developers.google.com/kml/documentation/kml_tut?hl=es-419

establecido en el software *RAZIEL* que ejecutan los nodos de monitorización (Sección 5.6.5.3).

5.13.2.5 *Actualización de Pasos por Horas*

Este subsistema se encarga de calcular los pasos que han transcurrido por cada nodo/sensor agrupados en intervalos y subirlos a una tabla de Google Fusion Tables. Una vez que se ha establecido la fecha de la última actualización en la nube, ya sea mediante el Código A.8 o establecido de forma manual, se procede con la consulta que calcula los valores solicitados mediante el procedimiento SQL presentado en la sección **??**. Una vez concluida la ejecución, se empaquetan los valores calculados y se sincronizan con la base de datos en la nube. El código A.9 muestra un resumen de este funcionamiento.

En dicho código se presenta la variable `infoNodo_pasoPorHora`. Dicha variable obedece a un subsistema de cálculo de estadísticas que ha fecha de escritura de esta tesis no ha sido terminado.

5.13.2.6 *Actualización de Trazas por Horas*

Este módulo es encargado de sincronizar las trazas por horas entre pares de sensores, agurpados por un intervalo de tiempo determinado. Para el cálculo de los datos hace uso del procedimiento SQL presentado en la Sección **??**. Su funcionalidad se resume en el Código A.10.

Para facilitar la representación, este método dibuja una línea mediante KML para geoposicionar la traza un mapa de Fusion Tables. También calcula de forma aproximada la distancia en línea recta de los dos nodos detectados[120].

5.13.2.7 *Cliente Actualizador FT*

Todos estos submódulos son controlados por el módulo Actualizador FT, que consta de dos partes concretas: uno para el primer arranque (donde todos los valores deben ser subidos por primera vez) y otro modo automático que se encarga de realizar la sincronización de forma periódica.

Este módulo hace uso de un evento programado mediante el objeto temporizador que instanciado en el constructor tal y como muestra el código A.12.

Esta tarea es ejecutada cada X minutos por el módulo, en función de lo que indique el fichero de configuración.

120 ↑Esta información puede ser contrastada con tablas auxiliares que contengan la distancia real en carretera entre los distintos nodos.

5.14 HERRAMIENTAS DE GESTIÓN, DIFUSIÓN Y PUBLICACIÓN

En esta última sección se presentan de forma breve los componentes del prototipo de sistema de monitorización desarrollados a modo de ejemplo de vías de difusión y publicación de los resultados. Estos resultados implican desde la publicación de los valores en crudo obtenidos por la fuente de datos propuesta, la información sobre la implantación y estado de los nodos de monitorización y los resultados de los algoritmos de análisis y aprendizaje ejecutados de forma automática.

Si bien esta parte será abordada de forma liviana en esta tesis, el desarrollo de herramientas de esta naturaleza resulta de vital importancia para la gestión del sistema de monitorización, así como el empleo aplicado de la fuente de datos para representar información en tiempo real.

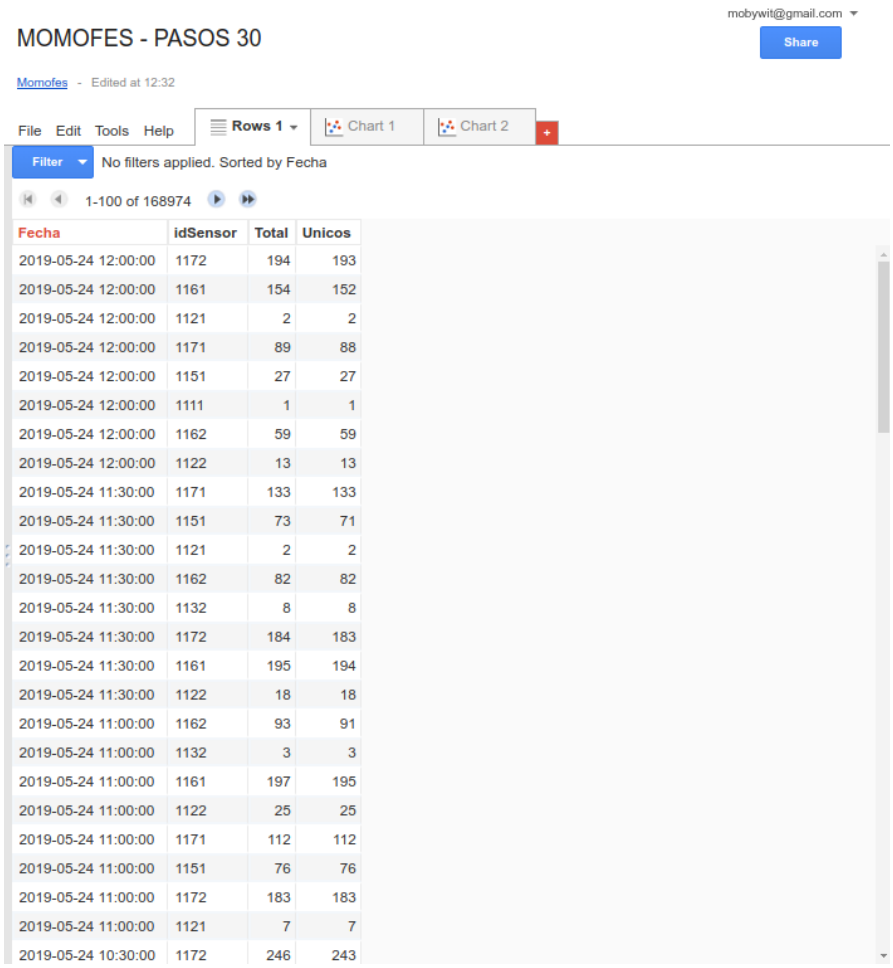
Se presentarán los métodos de publicación que ofrece Google Fustion Tables (Sección 5.13), la plataforma de difusión web desarrollada (Sección 5.14.2) , el panel de control de los nodos de monitorización (Sección 5.14.3), el agente de publicación en twitter (Seccción 5.105, así como las herramientas basadas en `R` para la generación automática procedural de informes (Sección 5.14.5) y paneles de información web o `dashboards` (Sección 5.14.6).

5.14.1 *Difusión mediante Google Fusion Tables*

La plataforma elegida para el almacenamiento en la nube empleada permite de forma nativa numerosas maneras para publicitar la información. En el Anexo A.6 se presentan a modo de ejemplo alguno de estos usos.

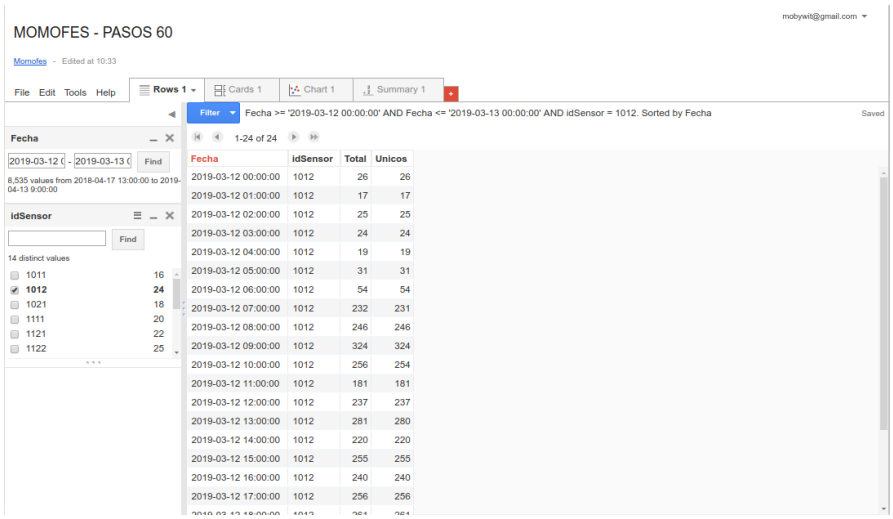
En la Figuras 5.100 y 5.101 se presentan algunos ejemplos de modos de representación de los que dispone la plataforma.

La ventaja de estos métodos es que quedan vinculados a las tablas, por lo que cualquier adición a las mismas que cumplan los criterios de visualización (como los filtros) será automáticamente añadida a las mismas.



Fecha	idSensor	Total	Unicos
2019-05-24 12:00:00	1172	194	193
2019-05-24 12:00:00	1161	154	152
2019-05-24 12:00:00	1121	2	2
2019-05-24 12:00:00	1171	89	88
2019-05-24 12:00:00	1151	27	27
2019-05-24 12:00:00	1111	1	1
2019-05-24 12:00:00	1162	59	59
2019-05-24 12:00:00	1122	13	13
2019-05-24 11:30:00	1171	133	133
2019-05-24 11:30:00	1151	73	71
2019-05-24 11:30:00	1121	2	2
2019-05-24 11:30:00	1162	82	82
2019-05-24 11:30:00	1132	8	8
2019-05-24 11:30:00	1172	184	183
2019-05-24 11:30:00	1161	195	194
2019-05-24 11:30:00	1122	18	18
2019-05-24 11:00:00	1162	93	91
2019-05-24 11:00:00	1132	3	3
2019-05-24 11:00:00	1161	197	195
2019-05-24 11:00:00	1122	25	25
2019-05-24 11:00:00	1171	112	112
2019-05-24 11:00:00	1151	76	76
2019-05-24 11:00:00	1172	183	183
2019-05-24 11:00:00	1121	7	7
2019-05-24 10:30:00	1172	246	243

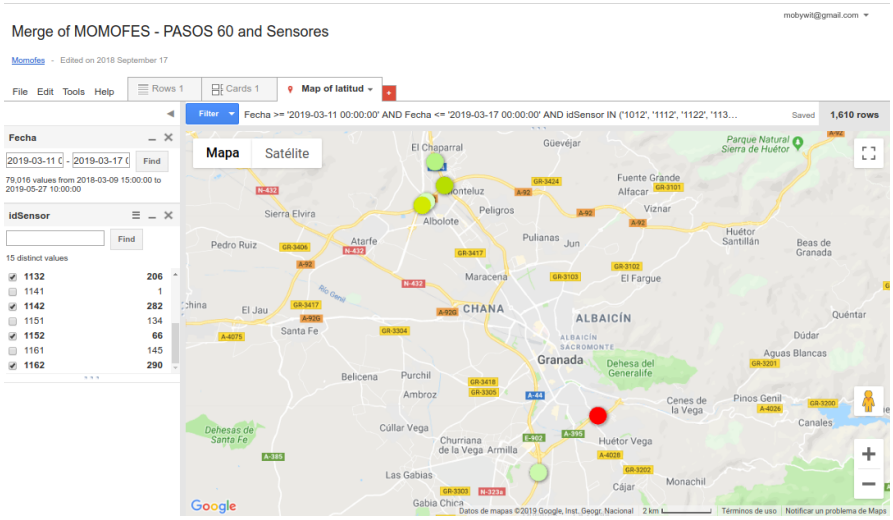
(a) Tabla



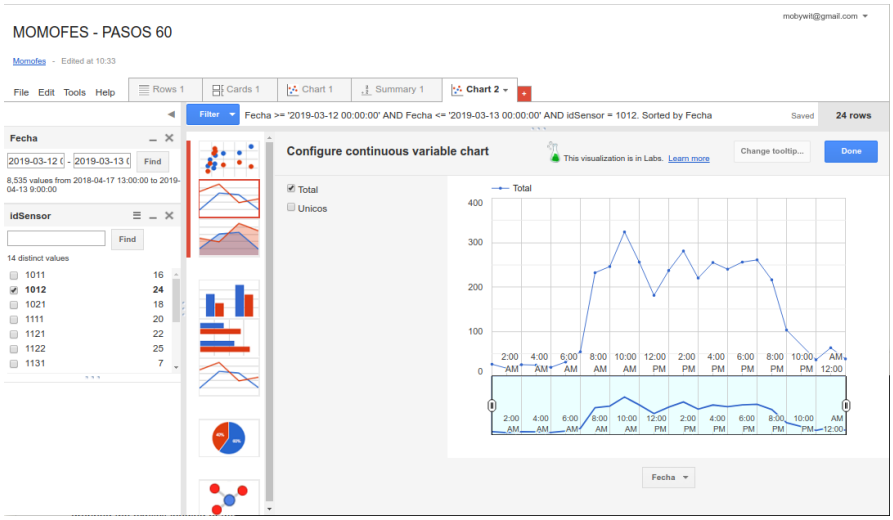
Fecha	idSensor	Total	Unicos
2019-03-12 00:00:00	1012	26	26
2019-03-12 01:00:00	1012	17	17
2019-03-12 02:00:00	1012	25	25
2019-03-12 03:00:00	1012	24	24
2019-03-12 04:00:00	1012	19	19
2019-03-12 05:00:00	1012	31	31
2019-03-12 06:00:00	1012	54	54
2019-03-12 07:00:00	1012	232	231
2019-03-12 08:00:00	1012	246	246
2019-03-12 09:00:00	1012	324	324
2019-03-12 10:00:00	1012	256	254
2019-03-12 11:00:00	1012	181	181
2019-03-12 12:00:00	1012	237	237
2019-03-12 13:00:00	1012	281	280
2019-03-12 14:00:00	1012	220	220
2019-03-12 15:00:00	1012	255	255
2019-03-12 16:00:00	1012	240	240
2019-03-12 17:00:00	1012	256	256

(b) Query

Figura 5.100
Ejemplo de publicación con `Google Fusion Tables`. En la subfigura 5.100(a) se presentan los datos almacenados en forma de tabla. Los datos de esta table pueden ser filtrados para realizar una consulta, como se muestra en la subfigura 5.100(b), donde se han seleccionado los pasos del nodo 1012 del día 12 de Marzo de 2019.



(a) Mapa



(b) Gráfica interactiva

Figura 5.101
Ejemplo de publicación con `Google Fusion Tables`: Mapas y Gráficas. `Google Fusion Tables` se integra de forma automática con otros servicios y librerías de Google. Su integración con `Google Maps` permite geoposicionar los valores de las tablas como se presenta en la subfigura 5.101(a). Su integración con `Google Charts` permite representar en forma de gráfico los valores almacenados como se muestra en la subfigura 5.101(b).

5.14.2 *Plataforma de difusión WEB*

Aunque la plataforma basada en la nube permite representar directamente los datos almacenados en un mapa, su capacidad de modificación es limitada.

Sin embargo, los datos pueden ser solicitados mediante una petición a la API REST, de forma que puedan ser representados de forma manual en cualquier aplicación geoposicional.

En la Figura 5.102 se representan un ejemplo de aplicación web basada en la adquisición de los datos almacenados en la nube para ser representados en un mapa interactivo. Este mapa dispone de un selector de Fecha y hora, y permite reproducir el tráfico entre los nodos. El código se presenta brevemente en el Anexo A.6.1.2.

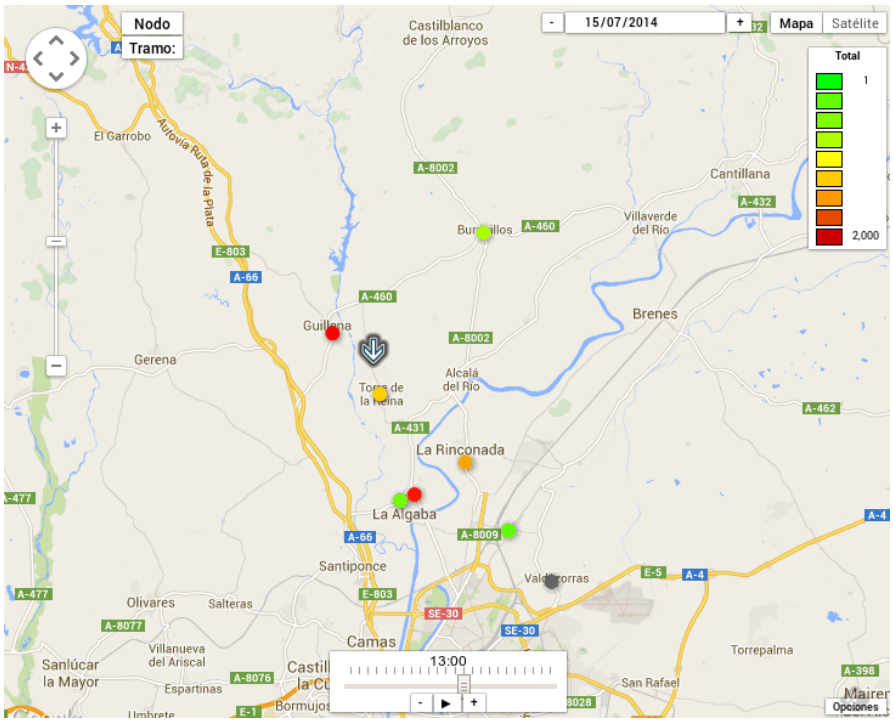


Figura 5.102
Ejemplo de plataforma web de difusión de resultados por medio de una visualización interactiva sobre un mapa.

La potencialidad para el desarrollo de aplicaciones con las que mostrar la información obtenida por el sistema de forma eficiente, así como modularidad y facilidad (tanto del despliegue como del desarrollo), permite difundir la información de múltiples maneras. Por ejemplo, como entregable de uno de los proyectos realizados, se realizó una aplicación móvil en Android que permitiese acceder a la información generada en el sistema. Algunas capturas de esta aplicación se recogen en la Figura 5.103.

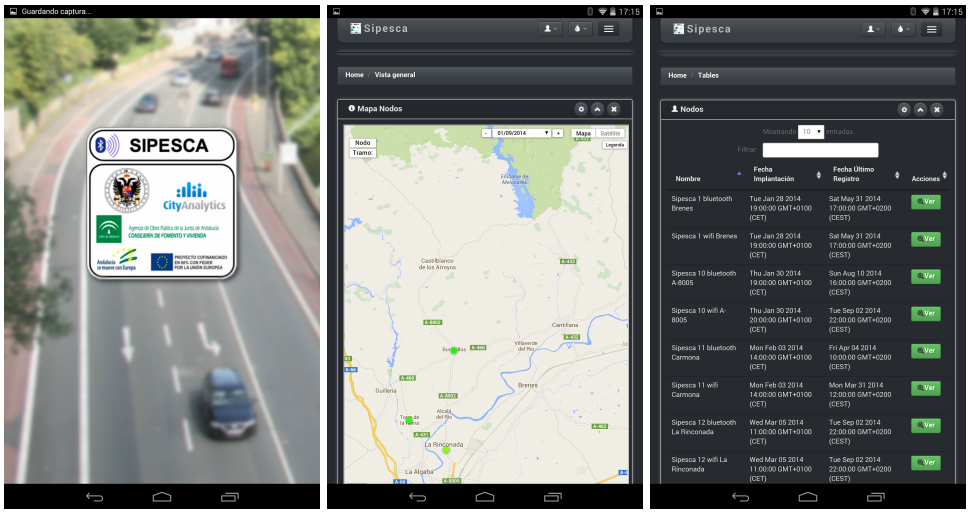


Figura 5.103
Tweets automáticos generados por el agente de Twitter.

Si bien ninguna de estas aplicaciones se presentan como resultado inmediato de la tesis, los métodos de desarrollo y metodologías presentadas a lo largo de este capítulo permiten la fácil interoperabilidad y accesibilidad de los datos obtenidos por la fuente de monitorización propuesta. Estos métodos y cuestiones de diseños, si bien son indepentientes del sistema de monitorización en si, sirven de ejemplo para demostrar la aplicabilidad de la fuente de datos propuestas y el sistema de monitorización prototipado que hace uso de ella.

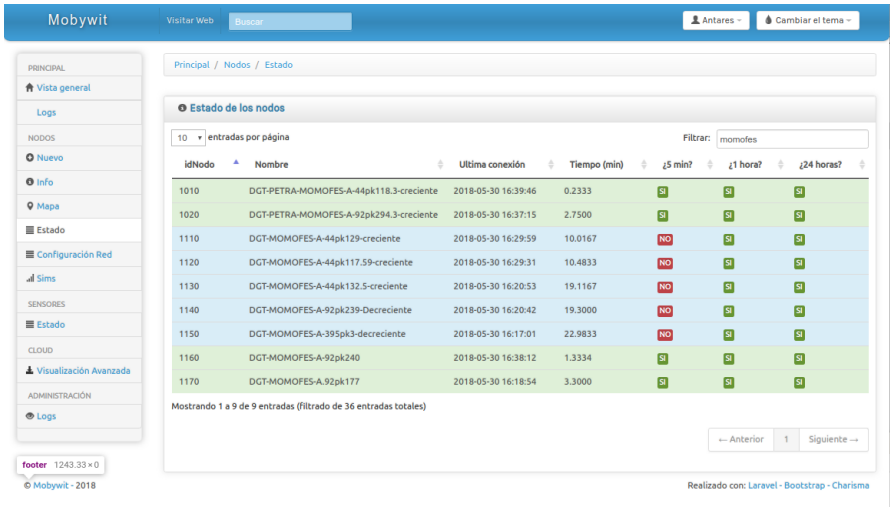
5.14.3 *Panel de control*

Con el fin de poder disponer de información en tiempo real sobre el estado de los nodos, se implementa una plataforma de control web. Esta plataforma permite ver a simple vista el funcionamiento de los nodos, así como su configuración, disposición y uso de la plataforma de comunicación (Sección 5.8).

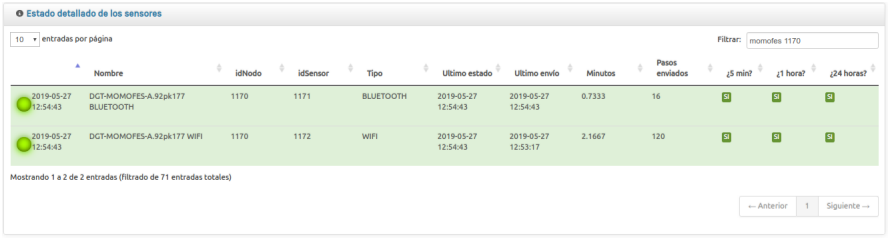
En la Figura 5.104(a) se presenta un vistazo de la plataforma, con la información del estado de un proyecto determinado, indicando cuando fue la última vez que establecieron comunicación con el servidor. Esta información, puede ser obtenida tanto para los nodos, como para los sensores de cada nodo, como se presenta en la Figura 5.104(b).

La plataforma web se emplea también para registrar los nuevos nodos del sistema, así como cambiar la configuración de los nodos ya existentes.

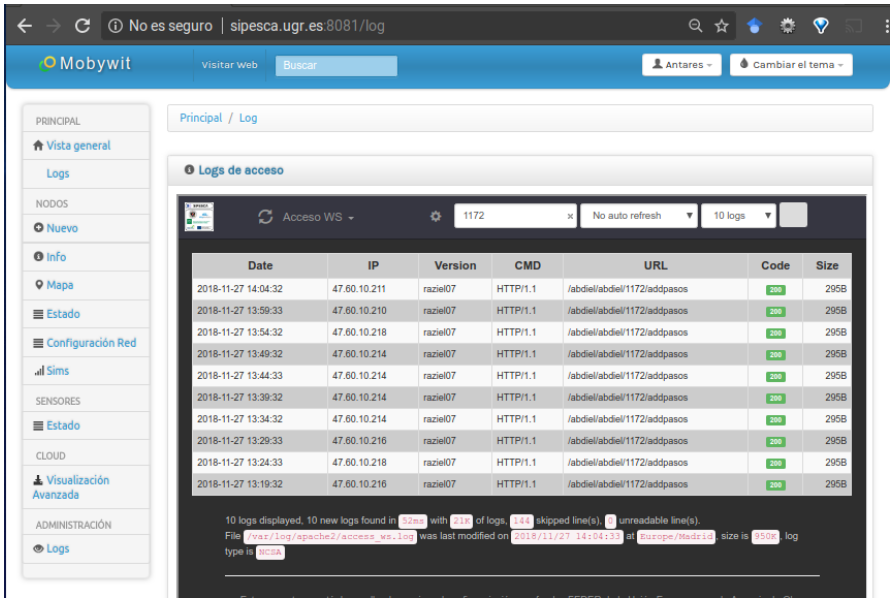
Por último, permite acceder a la información del acceso de cada nodo al servidor, mostrando en tiempo real los `logs` de la plataforma de comunicación como se presenta en la subfigura 5.104(c).



(a) Estado nodo



(b) Estado sensores



(c) Logs de acceso

Figura 5.104
Panel de control web, que permite acceder al estado de nodos y sensores, así como trazar la comunicaciones de los nodos con el servidor.

5.14.4 *Agente de publicación en Twitter*

Otro ejemplo de servicio de difusión implementado es un bot de twitter que publica de forma periódica información sobre el número de dispositivos detectados por un nodo o el tiempo de desplazamiento por un número de vehículos detectados por dos nodos. Algunos tweets de ejemplo se presentan en la Figura 5.105.

Figura 5.105
Tweets automáticos generados por el agente de Twitter.

El agente tiene un comportamiento pseudo-inteligente, realizando en cada iteración varios tweets y publicando únicamente aquel que muestra la información más relevante. Además, su frecuencia de publicación se adapta a la hora del día[121].

5.14.5 *Generación de informes automáticos con RMarkdown*

En la Sección 5.11.3.1 se ha presentado la librería `Mobywit` desarrollada en R para el acceso y análisis de la información.

Un parte importante de la librería son los módulos encargados de la generación procedural de informes sobre la monitorización realizada, el estado de los nodos o el tránsito. Estos módulos hacen uso de `Rmarkdown`[122] para generar documentos fácilmente replicales y reproduciles.

El uso de esta librería permite que se puedan realizar los análisis a los conjuntos de datos de forma automática, requiriéndose unicamente realizar una llamada al renderizado con los parámetros que se deseen, como se presenta en el Código 5.70. En este código, se renderiza el fichero `informe_nodo.Rmd` con los parámetros indicados.

121 ↑Por ejemplo, tiene poco sentido publicar tweets de madrugada, a no ser que se detecte alguna anomalía interesante.
122 ↑`https://rmarkdown.rstudio.com/`

Código 5.70
Generación de documentos con Rmarkdown

```
render("informe_nodo.Rmd", params=list(nodo=1010,DB=TRUE,MAPAS=FALSE),output_file =
↪  paste0(nombre,"_informe_",1010,".html"))
```

Este fichero mezcla código markdown para componer el texto, con bloques de código R llamados chunks. Estos bloques de código son ejecutados cuando el fichero es renderizado, añadiendo a la salida del fichero los resultados que haya producido la ejecución del chunk, como tablas, mapas, figuras o gráficos. De esta forma es posible generar un documento que conste de la información más actualizada posible. En la Figura 5.71 se presenta la estructura de un fichero Rmarkdown habitual.

Código 5.71
Ejemplo de fichero Rmarkdown. En las primeras líneas se define el autor, los parámetros de salida (html en el ejemplo), y los parámetros definidos con sus valores por defecto. Se emplea sintaxis markdown seguidos de bloques chunks de código R

```
---
author: "A.Fernández-Ares"
date: "29/11/2018"
output: html_document
params:
  set_title: "Infome MOMOFES"
  nodo: "1010"
  DB: false
  MAPAS: false
title: Información del nodo `r nodo`
subtitle: `r getNombreNodo(nodo)`
---

# Localización del nodo
```{r print_map, echo=FALSE,error=FALSE,warning=FALSE}
grid.arrange(mapTodos,mapaSensor,nrow=1)
```

##Estado de funcionamiento del Sensor BLUETOOTH
```{r grafico_funcionamiento_BT, echo=FALSE,warning=FALSE,error=FALSE}
fun_BT <- aggregate(p_h[p_h$idSensor==nodo+1,"d"],by = list(d
↪  =p_h[p_h$idSensor==nodo+1,"d"]),FUN = length)
calendarHeat(dates = fun_BT$d, values = fun_BT$x,ncolors = 3,varname ="BLUETOOTH")
```

##Estado de funcionamiento del Sensor WIFI
```{r grafico_funcionamiento_WIFI, echo=FALSE,warning=FALSE,error=FALSE}
fun_WIFI <- aggregate(p_h[p_h$idSensor==nodo+2,"d"],by = list(d
↪  =p_h[p_h$idSensor==nodo+2,"d"]),FUN = length)
calendarHeat(dates = fun_WIFI$d, values = fun_WIFI$x,ncolors = 3,varname = "WIFI")
```
```

Al estar la parte del análisis de datos implementado en la librería en R Mobywit, es inmediato volver a realizar un informe o análisis de los datos y obtener los resultado para un nuevo conjunto de datos. Esto es vital trascendencia debido a que los datos del sistema de monitorización propuesto se generan a lo largo del tiempo. De igual manera, incorporar nuevos análisis o

estudios a un informe, requiere unicamente añadir un nuevo bloque `chunk` que llame a la función de la librería encargada de realizar ese estudio.

El resultado del informe puede ser un documento `HTML`, `DOC` o `PDF` que puede ser ampliado con las anotaciones a mano que sean fruto de la interpretación de los resultados del análisis realizado proceduralmente. En la Figura 5.106 se presentan las miniaturas de las más de 40 páginas obtenidas por el renderizado de un informe.

Figura 5.106
Páginas del Informe generado proceduralmente mediante `RMarkdown`

Si bien las gráficas, tablas y demás recursos gráficos no constituyen por si solos un informe, la capacidad del sistema de generar las plantillas de estos incorporando la última información disponible o la información histórica deseada, así como los análisis deseados, permite generar y publicar la información de manera prácticamente inmediata a medida que se va produciendo.

5.14.6 *Generación de paneles de información en tiempo real*

De igual manera que se pueden generar informes mediante `Rmarkdown`, `R` incorpora por medio de su suite `Rstudio` un conjunto de librerías denominadas `Shiny`[123] que permite publicar en la web los resultados de la ejecución de scripts `R` bajo demanda y de forma automática. Entre los usos de esta herramienta, está la de poder diseñar paneles de información o `dashboard` que se actualicen de forma automática.

En la Figura 5.107 se presenta un panel de información con distintos elementos, que se actualizan de forma automática con la información del servidor.

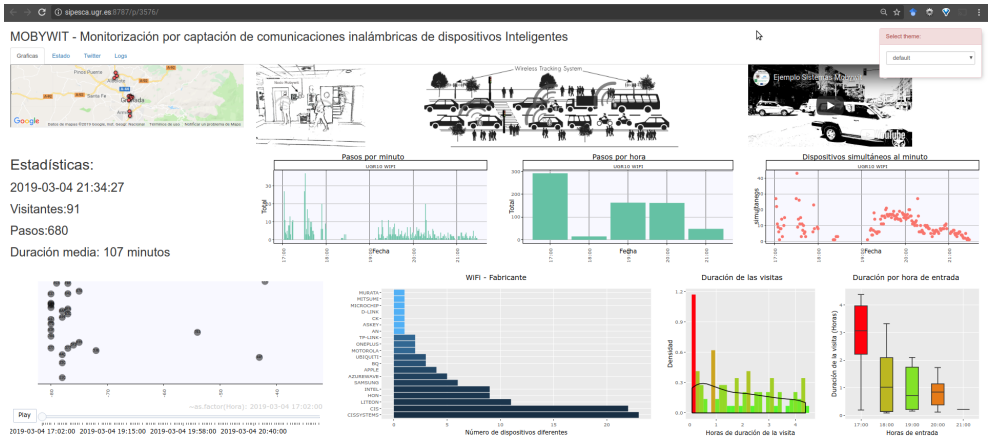


Figura 5.107
Ejemplo de `dashboard` con `RShiny`.

Cada elemento del `dashboard` se actualiza de forma automática según intervalos de tiempo. En el Anexo A.6.2 se presentan brevemente las estructuras de estos paneles, así como los fundamentos de los mecanismos de autoinvalidación que hace que la información se actualice de forma automática.

123 ↑https://shiny.rstudio.com/

Parte III

Resultados

6

EXPERIMENTACIÓN

¿Es este mi propósito? ¿Soy solo un experimento?
¿Un espécimen de laboratorio? No. Ese no puede ser mi destino.

— Mewtwo

En esta sección se presenta una selección de experimentos organizados siguiendo las hipótesis planteadas en la Sección 1. De esta forma, en primer lugar se presentarán experimentos para demostrar la viavilidad de la captación de comunicaciones inalámbricas para la monitorización de dispositivos inteligentes. En segundo lugar, se presentarán los experimentos que presentan estudios y análisis de la movilidad de personas y vehículos realizados con el sistema presentado en esta tesis. Finalmente, se presentan experimentos para demostrar viabilidad de la aplicación de técnicas de soft-computing en la fuente de datos del sistema, que permitan extraer información y conocimiento que sea útil para una ciudad inteligente.

Índice del capítulo

6.1 EXPERIMENTOS HIPÓTESIS I: SOBRE LA CAPTACIÓN, RECONOCIMIENTO Y MONITORIZACIÓN DE LAS COMUNICACIONES INALÁMBRICAS DE LOS DISPOSITIVOS INTELIGENTES

En esta Sección se presentan los experimentos que buscan demostrar la viabilidad de la captación de comunicaciones inalámbricas de dispostivos inteligentes para la monitorización. Se presentan para ello las cuestiones que complementan al Capítulo 4, que sustenta de forma teórica la viabilidad de la captación de las cominicaciones inalámbricas origen de la fuente de datos propuesta en esta tesis.

En primer lugar, se presentarán las cuestiones relativas a la detección de dispositivos Bluetooth y la de dispositivos WiFi. Se presentarán algunos estudios realizados en el laboratorio con el fin de garantizar la integridad de los datos de captación. Finalmente se presentarán algunos estudios sobre la influencia del emplazamiento de los nodos de monitorización en la cantidad de dispositivos detectables.

6.1.1 *Experimentos relativos a la detección de dispositivos Bluetooth*

Cómo se ha presentado en las Sección 4.2, Bluetooth es un protocolo orientado a la búsqueda de dispositivos, por lo que dispone de los mecanismos de detección implementados de forma nativa en el protocolo.

Existen dos cuestiones a resolver sobre la aplicación de búsqueda de dispositivos Bluetooth en el prototipo implementado: Cada cuanto puede detectar el nodo de monitorización implementado y cada cuanto tiempo emiten los dispositivos Bluetooth en las inmediaciones del nodo.

Adicionalmente, se realiza un análisis exploratorio de los tipos de dispositivos Bluetooth que han sido detectados por los nodos de monitorización en producción.

Experimentos relativos a la detección de dispositivos Bluetooth

Estudio 6.1.1: Determinación del intervalo de detección del nodo.

En la detección de dispositivos Bluetooth hay tres órdenes o comunicaciones con el Driver de la tarjeta de red Bluetooth que son perpetrados por el nodo. En primer lugar, el nodo solicita a la tarjeta de red que comience con una búsqueda de dispositivos bluetooth o `Inquiry` (Sección 4.2.3) mediante el método `deviceInquiryStartedCallback`.

La activación de este modo no es inmediato, por lo que una vez que la tarjeta de red Bluetooth inicia este modo devuelve un `startInquiry.return.true` al nodo de monitorización. Cada dispositivo Bluetooth detectado genera un mensaje `deviceDiscoveredCallback` una única vez en el intervalo de detección, aunque haya sido detectado más de una vez por la tarjeta de red en la duración de la `Inquiry`. Cada detección generará un mensaje que es procesado sin interrupción del sensor por parte del monitor (Sección 5.6.5.4.2).

Una vez expira el tiempo determinado para la `Inquiry`, la tarjeta de red devuelve un `runDeviceInquiry.ends` para indicarle al nodo que ha finalizado el modo de búsqueda y cesa el envío de mensajes hasta que el modo vuelva a ser activado. El nodo debe por tanto solicitar que se active nuevamente el modo `Inquiry` a la tarjeta de red.

En la Tabla 6.1 se presentan los instantes de tiempo medidos en un caso real de monitorización por Bluetooth, con el resto del sistema y sus módulos en funcionamiento.

Tabla 6.1
Intervalo de detección del nodo de implementado para comunicaciones Bluetooth.

deviceInquiryStartedCallback	startInquiry.return.true	runDeviceInquiry.ends
12:41:18.084	12:41:18.085	12:41:28.332
12:41:28.792	12:41:28.792	12:41:39.052
12:41:39.566	12:41:39.568	12:41:49.826
12:41:50.295	12:41:50.296	12:42:00.542
12:42:01.013	12:42:01.013	12:42:11.270
12:42:12.223	12:42:12.223	12:42:22.482
12:42:23.006	12:42:23.006	12:42:33.345
12:42:33.782	12:42:33.782	12:42:44.033
12:42:44.536	12:42:44.536	12:42:54.795

Entre que el nodo de monitorización envía el `deviceInquiryStartedCallback` y el driver devuelve el `startInquiry.return.true` indicando que ha activado el modo transcurre un tiempo aproximado de $1ms$. Esto es debido a que se fuerza que la tarjeta de red funcione en modo exclusivo para el software del nodo de monitorización, sin que ningún otro elemento software puedan hacer uso del dispositivo. De esta forma, la tarjeta de red no tiene ninguna otra solicitud que atender, transmisión que concluir o modo que finalizar antes de dar comienzo a la activación del modo `Inquiry`.

Sin embargo, desde que el nodo de monitorización recibe el mensaje `runDeviceInquiry.ends` hasta que comienza una nueva solicitud por medio

de un mensaje `deviceInquiryStartedCallback` transcurre un tiempo aproximado de 500*ms*. Produciéndose un gap o brecha de tiempo en el que los dispositivos en las inmediaciones no pueden ser detectados, representado gráficamente en las Figura 6.1.

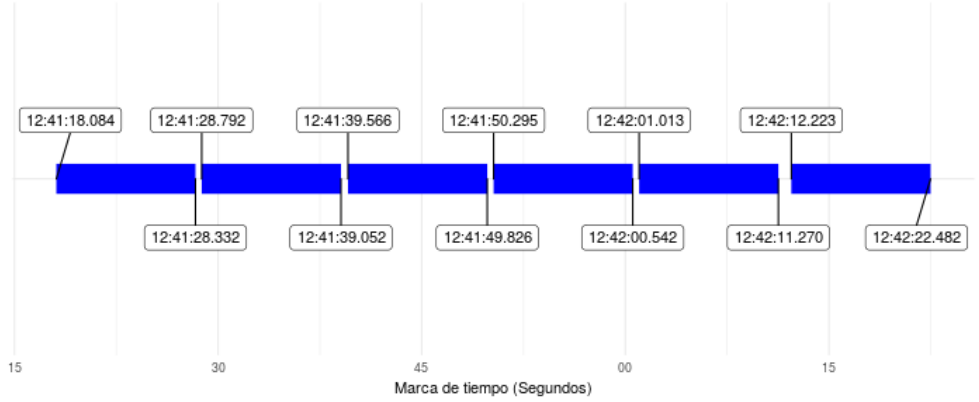


Figura 6.1
Brecha de derección en comunicaciones Blueooth debido a las distintas activaciones del modo.

La longitud del `Inquiry` depende de la implementación de la tarjeta de red y el driver y no puede ser cambiado [122, Pag. 50] sin modificación directa de estos elementos. En el caso de la tarjeta de red (Sección 5.4.2) y el driver empleado en el prototipo está establecido en 10 segundos. El modo puede ser cancelado mediante el envío de un `cancelInquiry`, pero añade 500*ms* de respuesta y otros 500*ms* en iniciar una nueva `Inquiry`.

La tabla 6.2 recoge las distancias que recorrerían los dispositivos potenciales de ser detectados a distintas velocidades habituales en los vehículos.

Tabla 6.2
Distancia recorrida según velocidades en la brecha del intervalo de detección Bluetooth

Distancia KM/H (KM/H)	Distancia m/s (m/s)	Distancia (metros) recorridos en la brecha	Distancia (metros) recorridos en el `Inquiry`
5	1.39	0.69	13.89
10	2.78	1.39	27.78
20	5.56	2.78	55.56
30	8.33	4.17	83.33
40	11.11	5.56	111.11
50	13.89	6.94	138.89
60	16.67	8.33	166.67
70	19.44	9.72	194.44
80	22.22	11.11	222.22
90	25.00	12.50	250.00
100	27.78	13.89	277.78
110	30.56	15.28	305.56
120	33.33	16.67	333.33
140	38.89	19.44	388.89
160	44.44	22.22	444.44
180	50.00	25.00	500.00
200	55.56	27.78	555.56
240	66.67	33.33	666.67

Debido a que la tarjeta Bluetooth empleada (Sección 5.4.2) es de Clase 1 (Tabla 4.1 en Sección 4.2) tiene un rango de operación hasta de $100m$ con una potencia de consumo de $100mW$. Si bien este rango de operación es esperable que en la práctica resulte menor, queda bastante lejos de los umbrales que puede recorrer los dispositivos detectables en el periodo de tiempo de la brecha entre activación y activación del modo `Inquiry`.

Por lo tanto, la probabilidad de que un dispositivo se encuentre en las inmediaciones del nodo únicamente durante la duración de un gap implica que ese dispositivo se encuentra a una velocidad muy excesiva o no se encuentra circulando por las inmediaciones del nodo de monitorización. Sin embargo, es necesario estudiar cuanto tiempo es necesario para que un nodo detecte al dispositivo en las inmediaciones.

Experimentos relativos a la detección de dispositivos Bluetooth

Estudio 6.1.2: Determinación del intervalo de detección de los dispositivos.

El otro aspecto a cuestionar es cuanto tiempo tarda en responder un dispositivo Bluetooth cuyo modo (Sección 4.2.3.2) permita su descubrimiento por un dispositivo en modo `Inquiry`. La Figura 6.2 representa los instantes de detección de un dispositivo bluetooth una vez se ha iniciado la ventana de detección.

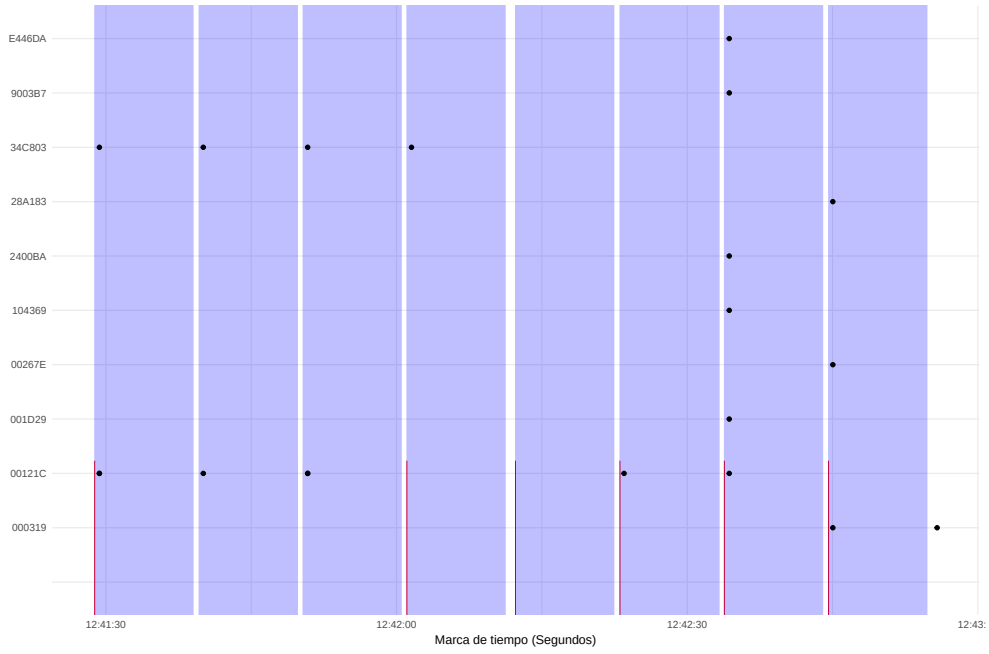


Figura 6.2
Ejemplo de detección de dispositivos Bluetooth. Las franjas azules denotan el tiempo de muestreo (10 segundos)

En la práctica, resulta prácticamente inmediato (inferior a $1ms$) desde que el dispositivo que se encuentra en un modo que permite ser detectado es detectado.

Experimentos relativos a la detección de dispositivos Bluetooth

Estudio 6.1.3: Tipos de dispositivos Bluetooth detectados

Los nodos de monitorización emplazados en carreteras (ya sea urbanas o interurbanas) desplegados en producción han detectado un más de 8 millones de veces dispositivos Bluetooth [1] pertenecientes a más de medio milón de dispositivos distintos.

Como se ha presentado en la Sección 4.2.2, la dirección MAC provee información sobre el fabricante del dispositivo. Se recoge en la Tabla 6.3 la proporción de Fabricantes de los dispositivos detectados reconocibles.

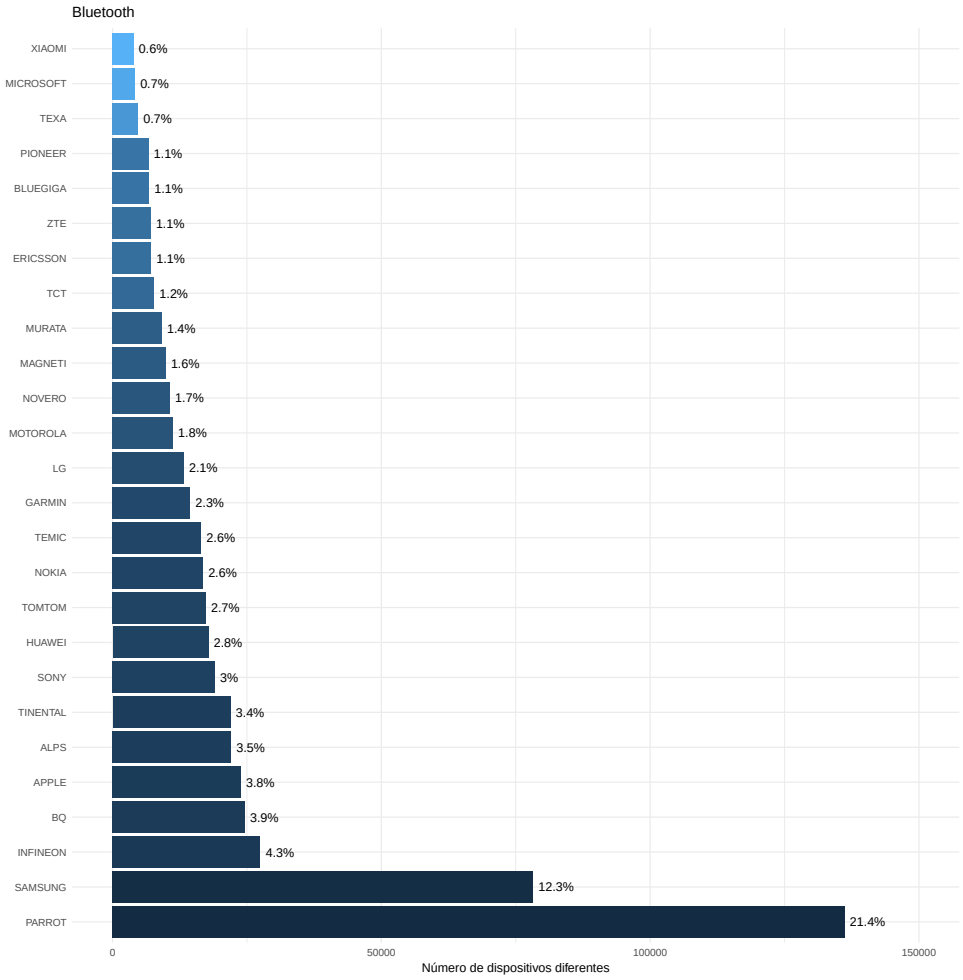


Figura 6.3
Proporción fabricantes de los dispositivos Bluetooth detectados identificables

De entre los fabricantes detectados, muchos de ellos pertenecen a fabricantes de dispositivos de manos libres (PARROT, NOKIA, SAMSUNG, ERICSSON, HUAWEI, MOTOROLA, LG, ZTE, XIAOMI), GPS (TOMTOM,GARMIN), reproductores de música (PIONER, SAMSUNG, SONY), de tarjetas de red o `NICs` para vehículos (NOVERO, TCT, TEMIC, BLUEGIGA, ALPS, RESEARCH). Así

1 ↑Un total de 8 306 092 de pasos Bluetooth a fecha de escritura de esta tesis, pertenecientes a 743 206 dispositivos Bluetooth, de los cuales ha podido ser reconocido el fabricante en 636 168 dispositivos.

como fabricantes de teléfonos móviles y `smartphones` (NOKIA, SAMSUNG, ERICSSON, HUAWEI, MOTOROLA, APPLE, LG, ZTE and XIAOMI).

Cómo se presentó en la Sección 4.2.3.5, los dispositivos Bluetooth identifican su naturaleza mediante una codificiación en el paquete `FHS`, empleando 5 bits para determinar su naturaleza principal (o `Mayor device class`) y 6 bits para su naturaleza secundaria ((o `Minor device class`)).

En la Figura 6.4 se presentan la proporción de dispositivos Bluetooth detectados en función de la naturaleza determinada por el `Mayor` y `minor device class` indicado por el paquete `FHS`.

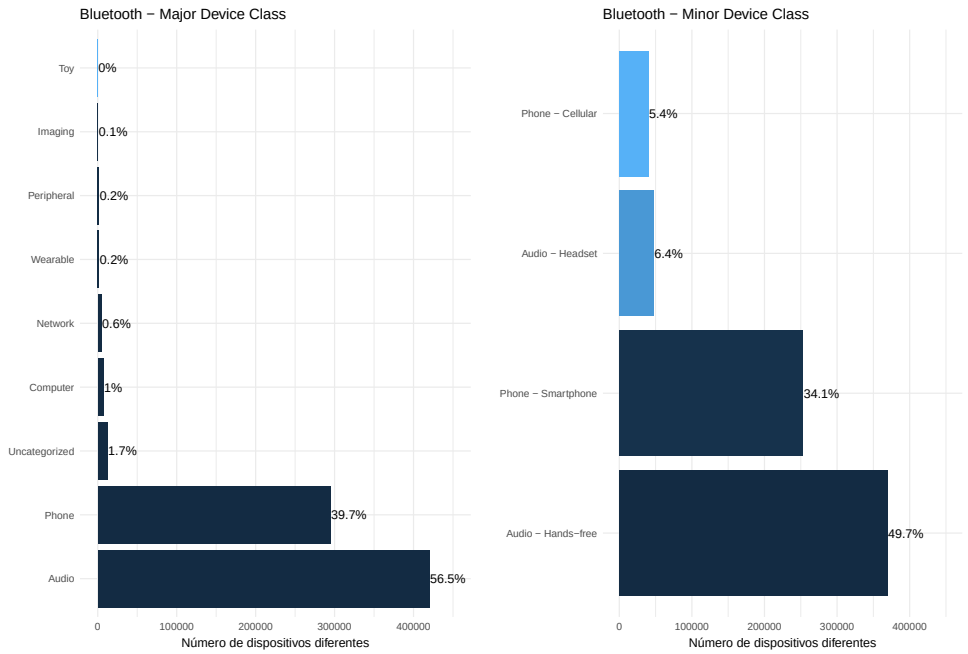


Figura 6.4
Tipos de dispositivos Bluetooth detectados según su naturaleza, determinada por su `Mayor device class` y su `Minor device class`.

La mayoría de dispositivos detectados, siendo más de la mitad, pertenecen a dispositivos de audio, que pueden ser de dos tipos distintos. Los dispositivos de manos libres (`hands-free`) y los `headset`, que es la denominación que se le otorga a los auriculares que incluyen micrófono.

Aproximadamente un 40 % de los dispositivos detectados son dispositivos clasificados como teléfonos, de los cuales sobre el total un 34.1 % son `smartphones` y un 5.4 % teléfonos `celulares` corrientes[2].

La detección de tantos dispositivos teléfonos entra en conflicto con el principio de conservación de batería que se impone en los smartphones, como se indicó en la Sección 4.2.5. Sin embargo, estos mecanismos de conservación de batería son anulados con el `smartphone` dispone de alimentación constante a través de su cargador. Esto nos hace pensar que la mayoría de los dispositivos

2 ↑Es decir, teléfonos móviles convencionales que no son considerados inteligentes.

Bluetooth detectados, se encuentran en vehículos conectados por un cargador a la batería del coche.

De igual manera, los dispositivos manos libres se encuentran alimentados por la batería del vehículo, por lo que suelen prescindir de mecanismos para el ahorro de batería. Así como, primando un diseño en el que se requiera la menor intervención del usuario, se encuentran siempre en modo de descubrimiento `General discoverable mode` que permite que sean detectados constantemente.

A pesar de la enorme cantidad de distintos tipos de dispositivos que emplean Bluetooth, el número de dispositivos detectados que son identificados como Juguetes (`Toy`), dispositivos de imagen como cámaras de vídeo y fotos (`Imaging`), periféricos como ratones, teclados o mandos de videoconsola (`Peripherical`) es prácticamente anecdótico. De igual manera, los `werables` que hacen en su mayoría uso de `Bluetooth LE` (Sección 4.2.1.2) son tan sólo un 0.2 % de los dispositivos detectados. Dispositivos de infraestructura de red (`Network`) son menos de un 1 % de los dispositivos detectados, de igual manera que los ordenadores (`Computer`). Estos dispositivos presentan una duración de estancia en las inmediaciones más elevada, como se puede observar en la Figura 6.5.

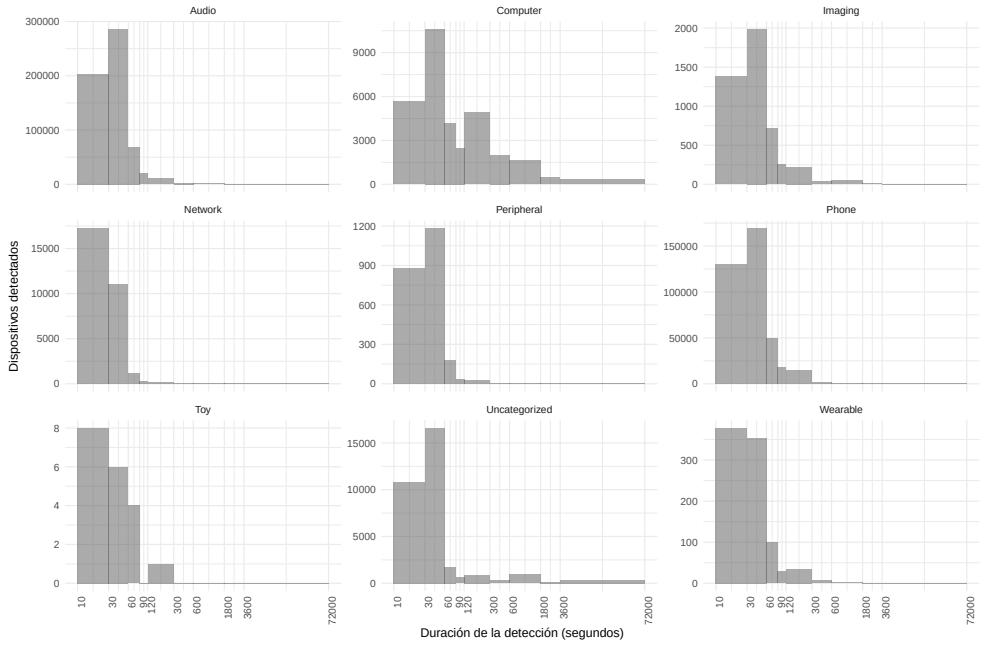


Figura 6.5
Duración de las detecciones según la naturaleza de dispositivo Bluetooth. El eje X se presenta en escala logarítimica para facilitar la legibilidad de la Figura.

Además, estos dispositivos suelen aparecer en los nodos de monitorización emplazados en zonas urbanas, donde los nodos pueden captar las comunicaciones emitidas por los dispositivos alojados en los edificios. La mayor afluencia de estos dispositivos se puede observar en la Figura 6.6.

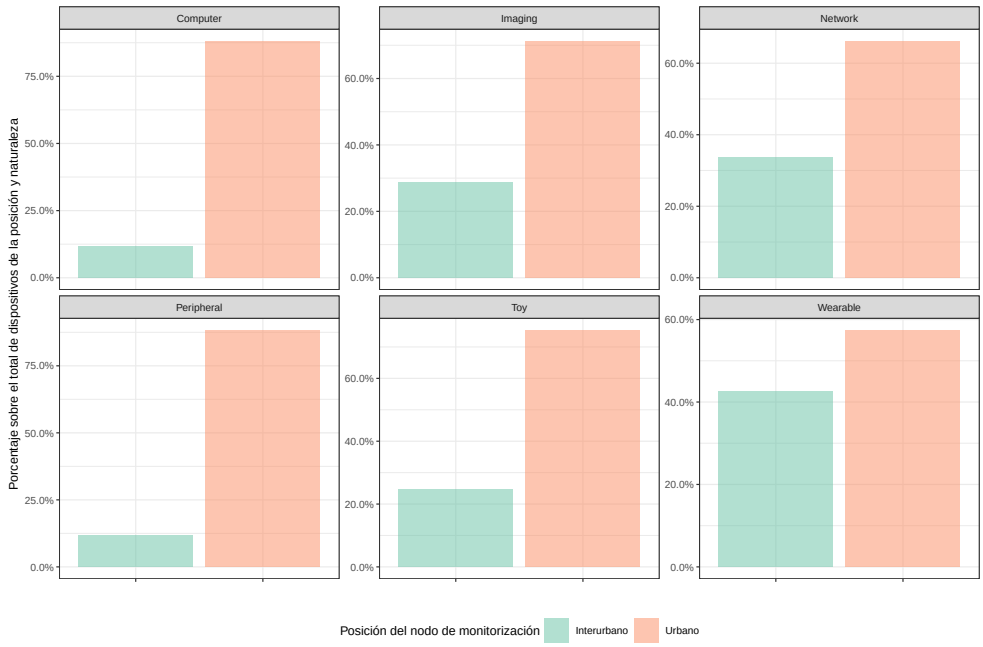


Figura 6.6
Naturaleza del dispositivo en función del punto de monitorización. Se ha normalizado el número de pasos en cada tipo de nodo para que su impacto en el porcentaje sea homogéneo aunque la magnitud sea distinta. Por tanto, el porcentaje muestra la proporción entre una y otra posición asumiendo que ambas posiciones han detectado el mismo número de dispositivo.

Conclusiones

El nodo de monitorización propuesto es capaz de detectar dispositivos bluetooth con pérdidas de 0.5 segundos cada 10 segundos. Debido a la clase de la tarjeta de red, su rango efectivo y a la velocidad esperable de los dispositivos circulando dentro de vehículos, no se prevé que se produzcan pérdidas significativas de dispositivos detectables.

Los dispositivos que se encuentran en un modo que permite su descubrimiento, son detectados por el nodo de monitorización en tiempos inferiores a $1ms$.

De los dispositivos detectados en el sistema en producción, el 56.5 % pertenece a un dispositivo manos libres. Cerca del 40 % pertenece a teléfonos que están configurados para permanecer siempre en un modo que permite su descubrimiento, lo cual tan solo suele ocurrir cuando disponen de una fuente de alimentación constante, como la que provee el cargador de un coche, que desactiva los sistemas de ahorro de energía.

La cantidad de dispositivos que no pertenece a estas categorías (3.8 % sobre el total), presenta tiempos de detección mayores y se observan principalmente en los entornos urbanos. Sin embargo, dado que el protocolo Bluetooth facilita información sobre la naturaleza del dispositivo, pueden ser fácilmente despreciados en los estudios relativos a la movilidad.

6.1.2 *Experimentos relativos a la captación WiFi*

Como se presentó en la Sección 4.3 el protocolo WiFi carece de mecanismos nativos para la detección de dispositivos en la inmediaciones. En su lugar, se emplea el modo monitor para capturar y analizar todo el tráfico de red generado en las inmediaciones del nodo de monitorización.

Se presentan varias cuestiones relativas a esta captación que son sujeto de estudio.

Experimentos relativos a la captación WiFi

Estudio 6.1.4: Determinación del tiempo requerido para la captura del nodo.

Según la arquitectura presentada en la Sección 5.6.1, existen tres elementos software independientes encargados de la monitorización de dispositivos WiFi: el sensor, el monitor y el notificador.

El sensor, es el encargado de capturar las tramas WiFi, y trasmitir aquellas que sean válidas[3] al monitor, para que extraiga la dirección MAC y actualice la ventana de detección. En la Figura 6.7 se han extraido instante procesamiento del sensor y el monitor durante 2 segundos de tiempo.

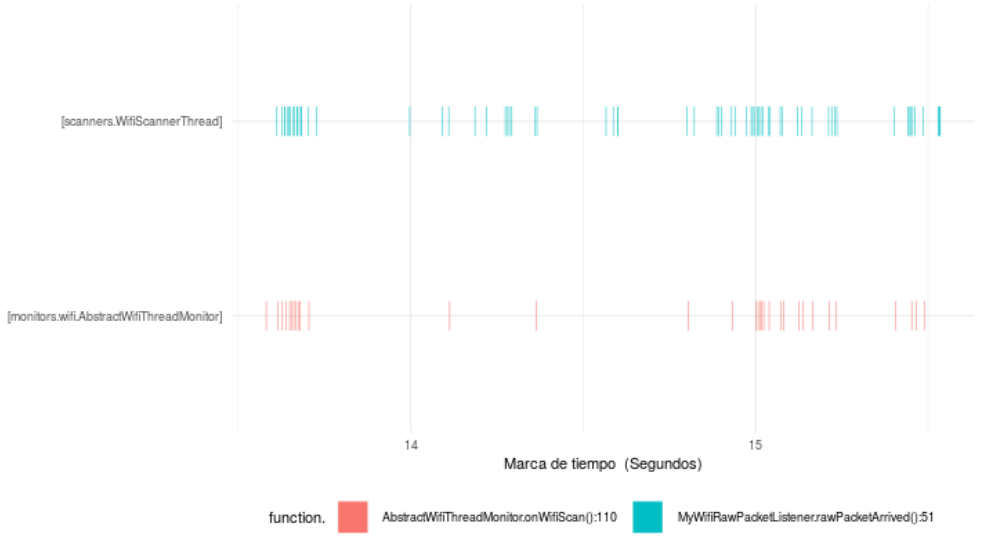


Figura 6.7
Tiempos de procesamiento del sensor y el monitor del sistema de monitorización WiFi. Como se aprecia, no todas las tramas detectadas por el sensor son enviadas al monitor.

Se observa que no todas las tramas capturadas en el sensor por el modo monitor de la tarjeta de red son enviadas al monitor. Como se presentó en la Sección 5.6.5.5.1, unicamente las tramas que son interpretables y por debajo de un umbral son enviadas al monitor, con el fin de ahorrar cómputo. Esto se observa en más detalle en la Figura 6.8.

3 ↑Entiendo como válidas aquellas que sean interpretables, no pertenezcan a comunicaciones de infraestructuras o se encuentren en los umbrales de la señal de recepción determinados.

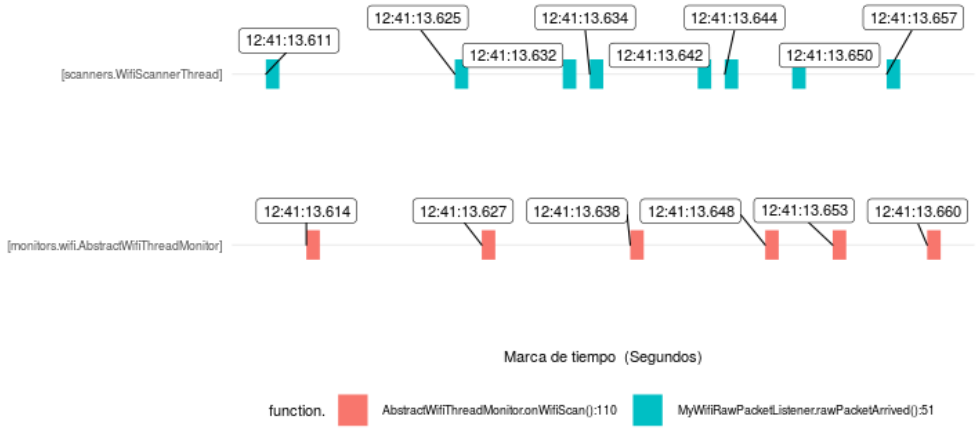


Figura 6.8
Detalle del procesamiento del sensor y el monitor del sistema de monitorización WiFi. Se ha representado 60*ms* de procesamiento.

El tiempo de procesamiento del sensor es inferior al 1*ms*[4] y al descartar tramas no interpretables resulta muy eficiente. Desde que la trama es capturada por el sensor hasta que es enviada al monitor transcurren entre 2 y 5 *ms*, por lo que es prácticamente inmediato.

Sin embargo, cabe la pregunta de cuantas tramas es capaz de procesar el sistema de monitorización propuesto.

Experimentos relativos a la captación WiFi

Estudio 6.1.5: Tramas por segundo procesables por el sistema

Como se ha presentado en la Sección 5.4.1 el bus USB donde se conectan las tarjetas de red tiene un ancho de banda de 480*mbps* (60*MBps*). La tarjeta de red (Sección 5.4.3) tiene un ancho de banda de 150*Mbps* (18.75*MBps*) según especificaciones del fabricante.

El tipo de tramas capturadas tienen tamaños que oscilan desde los 50*Bytes* al máximo teórico de 2304*Bytes*, pero se sitúan en torno a los 65*bytes* en la práctica (Tabla 6.3).

Tabla 6.3
Tamaño promedio de las tramas de red capturadas.

TIPO DE TRAMA	TAMAÑO
`Acknowledgemet`	50*Bytes*
`Probe Request`	150*Bytes*
`Probe Response`	300*Bytes*
`Beacon`	300*Bytes*
`Request-to-send`	50*Bytes*

4 ↑Un 1*m* es la unidad mínima de tiempo medible en el software desarrollado y por el sistema operativo, pues no se han añadido librerías adicionales de `Precision Time Protocol (PTP)`.

En entornos prácticos, el número de tramas procesadas por segundo oscila entre las $200 fps$[5] y las $800 fps$. Esto supone un tráfico capturado entre los $12 KBps$ y $50 KBps$ muy por debajo de la capacidad de la tarjeta de red.

En laboratorio empleando simuladores de tráfico[6] se ha logrado tráfico constante de $1\,300 fps$ ($85 KBps$) con picos puntuales de hasta $3\,000 fps$ ($200 KBps$).

En la práctica, ese flujo permanece más o menos constante, como se presenta en la Figura 6.9. El sistema ha capturado unos $1.42 GB$ diarios de tráfico de red, lo que supone un tráfico aproximado de $17.9 KBps$ de forma constante[7].

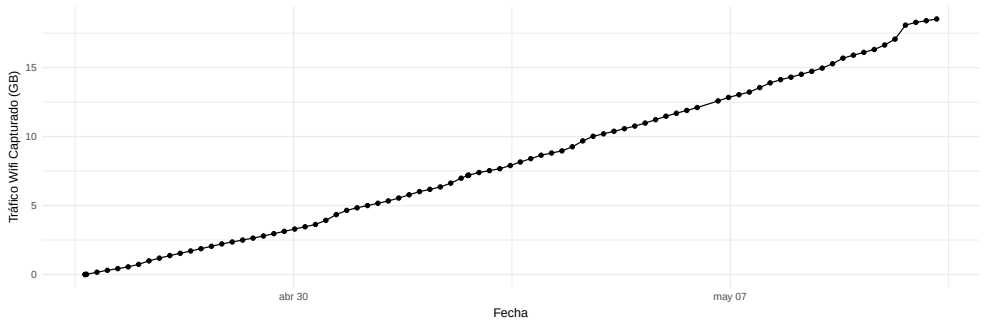


Figura 6.9
Cantidad de tráfico de red monitorizado por el sistema.

El sistema es capaz de lidiar con bastante tráfico de red, aunque aún presenta margen de mejora. Una mejor discriminación de que tramas de red pueden ser interpretables por el sistema permite reducir el procesamiento asociado a su monitorización.

Es neceario notar que estas pruebas han sido realizadas con nodos emplazados en zonas donde no existen redes WiFi en funcionamiento y que se tratan en su mayoría de zonas de tránsito, por lo que el tráfico capturado por el mono monitor es únicamente el relativo al tráfico de búsqueda de redes, no ha tráfico real de una red.

5 ↑Con fps como `frames per second` o tramas por segundo.
6 ↑Ostinato - https://github.com/pstavirs/ostinato.
7 ↑Lamentablemente, la adición del envío de esta información al servidor fue realizada al sistema de monitorización hace relativamente poco tiempo, por lo que se carece de información histórica de otros momentos en los que se han realizado pruebas de stress al sistema de monitorización.

Experimentos relativos a la captación WiFi

Estudio 6.1.6: Tramas WiFi generadas por dispositivos inteligentes

Una vez determinado que el sistema de monitorización propuesto es capaz de capturar un flujo constante de tráfico de red, es necesario cuestionarse cada cuanto tiempo es detectable un dispositivo que emplea WiFi, o lo que es lo mismo, cada cuando emite tramas WiFi los dispositivos inteligentes cuestionables de ser detectados. La Figura 6.10 presenta 1 minuto del tráfico de red capturado en las inmedicaciones de un nodo de monitorización en pruebas.

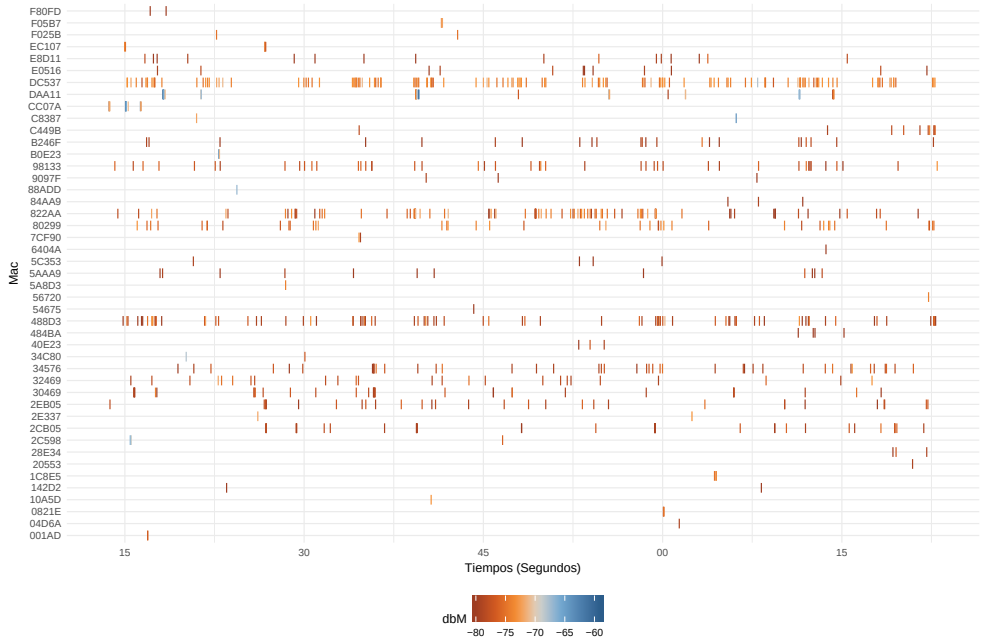


Figura 6.10
Instantánea del tráfico de red capturado por un nodo de monitorización durante 1 minuto. Se ha limitado el tráfico cápturado a ser recibido por debajo de $-80dBm$.

Los dispositivos que son detectados de forma reiterativa realizan sus comunicaciones en tiempos inferiores a los 10 segundos. Muchos otros dispositivos son detectados únicamente en un instante de tiempo, no volviendo a ser detectados en próximmos instantes de tiempo, se puede asumir, que o bien son dispositivos en movimiento o que espacian mucho en el tiempo su emisión de tramas.

Los dispositivos inteligentes disponen de modos de ahorro de energía que emplean sus sensores para determinar cuando el dispositivo está en reposo o cuando se encuentra en movimiento. La mayoría de fabricantes implementa estos mecanismos para aumentar la vida útil de sus dispositivos, siendo el mejor documentado el sistema `DOZE` empleado en los dispositivos `Android`[8]. En la Figura 6.11 se presenta el funcionamiento del sistema.

8 ↑El hecho de que el sistema de Android sea el más documentado, es debido a que es un componente que forma parte del `Android Open Source Project`, por lo que se tiene acceso al código. Sistemas cerrados como los de `iPhone` se saben que existen, pero no su funcionamiento exacto, del cual solo se puede elucubrar empíricamente.

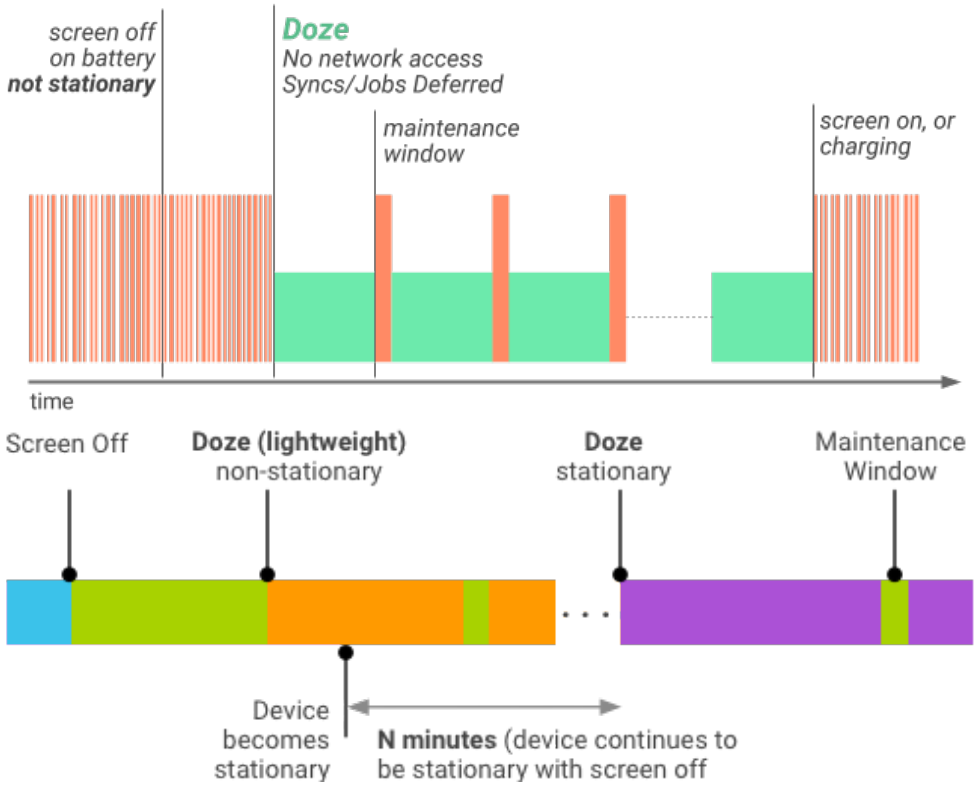


Figura 6.11
Esquema de funcionamiento del sistema de ahorro de energía DOZE empleado en dispositivos ANDROID. El dispositivo mantiene varios modos basado en el movimiento del dispositivo y el estado de la pantalla para apagar las comunicaciones del dispositivo, manteniendo ventanas espaciadas en el tiempo para habilitar nuevamente las comunicaciones.
Fuente: https://source.android.com/devices/tech/power/platform_mgmt

Mientras el dispositivo se encuentra en movimiento (`non-stationary`), el dispositivo tiene libre acceso a las comunicaciones. En al Figura 6.12 se presenta el tráfico de red capturado emitido por un único dispositivo a lo largo de 10 segundos. El dispositivo se encuentra con la pantalla apagada y no se encuentra conectado a ninguna red WiFi.

Aproximadamente cada 2 segundos, se detectan tramas de red emitidas por el dispositivo inteligente, a pesar de encontrarse con la pantalla apagada y sin conexión a ninguna red WiFi activa. El tipo de trama detectada es el empleado en la detección de redes WiFi.

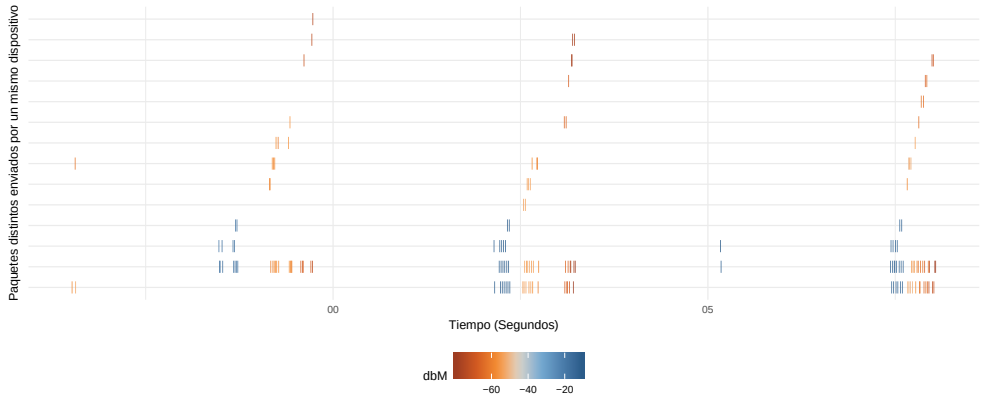


Figura 6.12
Tráfico WiFi capturado de un dispositivo en movimiento. El dispositivo no se encuentra conectado a ninguna red WiFi. Cada línea horizontal representa un tipo de paquete capturado, agrupando los que sean del mismo tipo y naturaleza en la misma línea horizontal aunque hayan sido emitidos en instantes distintos de tiempo.

En el momento que el dispositivo queda en absoluto reposo, como por ejemplo al posicionarlo encima de una superficie vertical, entran en funcionamiento los sistemas de ahorro de energía. Al detectarse el dispositivo con la pantalla apagada y en reposo, el tiempo habilitado para las comunicaciones incluida la búsqueda de redes se espacia enormemente, hasta superar los 10 segundos, como se recoge en la Figura 6.13, en lo que se conoce como ventana de mantenimiento o `Maintenance Windows` en el sistema `Doze`. El sistema de ahorro de energía asume que si el usuario no está prestando atención al dispositivo, no le supondrá ningún contratiempo atrasar las peticiones de nuevas comunicaciones.

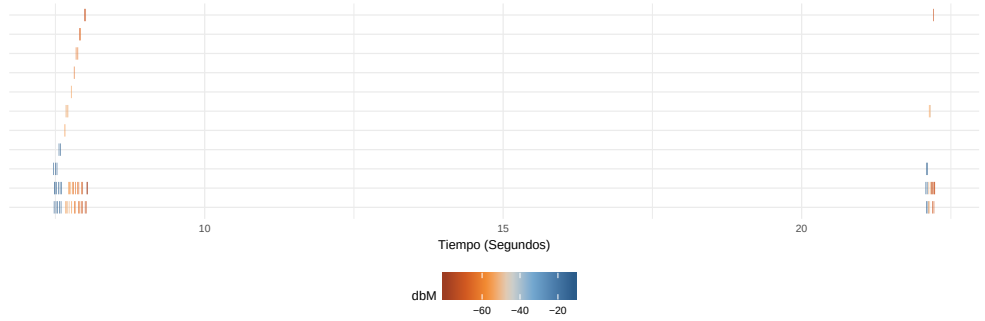


Figura 6.13
Tráfico WiFi capturado de un dispositivo en absoluto reposo, sobre una superficie horizontal estática. Entran en funcionamiento los modos de ahorro de batería y se espacian las comunicaciones y búsquedas de redes.

Una de las peculiaridades del sistema de búsqueda de redes WiFi de los dispositivos inteligentes, es que se realiza incluso aunque el dispositivo se encuentre con el WiFi apagado o en el modo avión. En ese caso, las peticiones se espacian mucho en el tiempo, del orden de 5 minutos con la pantalla apagada y en reposo, pero se activan automáticamente al encender la pantalla, mover el dispositivo o realizar cualquier desplazamiento o cualquier otra acción que requiera una interacción con el dispositivo. Esto es debido a

que los principales desarrolladores de sistemas operativos para dispositivos inteligentes, les interesa este comportamiento, pues la detección de redes WiFi es empleada por los dispositivos para mejorar su posicionamiento. La Figura 6.14 representa las tramas capturadas por un dispositivo en reposo en modo avión, con una frecuencia de búsqueda situada entre los 5 y 6 minutos.

Figura 6.14
Tráfico WiFi capturado de un dispositivo en reposo en modo avión.

Para evitar esto, los dispositivos inteligentes suelen tener un opción o varias muy recónditas para deshabilitarlo:

En los dispositivos Android, se tiene que activar el modo de localización por solo GPS, deshabilitar la búsqueda de redes WiFi en cualquier momento para mejora la ubicación, deshabilitar la interfaz WiFi y deshabilitar el modo Usar Wifi en suspensión.

En IOS, se tiene que deshabilitar los servicios de red WiFi en los ajustes de privacidad relativos al servicio de localización, deshabilitar la interfaz WiFi, deshabilitar la notificación de nuevas redes y deshabilitar las opciones de llamadas WiFi de la aplicación de teléfono.

Estas opciones, resultan demasiado concretas como para ser activadas sin intencionalidad por el grueso de usuarios. Además, con ella solo se impide que sea el sistema operativo el que realice la búsqueda de redes WiFi.

Sin embargo, multitud de aplicaciones solicitan el permiso de gestión de redes inalámbricas obteniendo permiso por defecto por parte de la inmensa mayoría de los usuarios. Solicitan la gestión de las redes inalámbricas pues emplean esta información de forma periódica para usos internos, basados principalmente en la detección o no de ciertas redes.

De esta forma, aplicaciones en apariencia tan simples como las pertenecientes a Starbucks, Burguer King o cualquier otra tienda que provea de conexión WiFi a sus clientes, estará configurada para detectar redes WiFi en la proximidad, pues es el mecanismo que emplean para determinar que el dispositivo se encuentra en uno de sus comercios, el ofrecimiento de una red WiFi con un determinado `SSID` (Ver Sección 4.3).

De esta forma, la única manera real de que un dispositivo inteligente no sea factible su detección por medio de la captación WiFi, es que se encuentre apagado. En cualquier otro escenario, requerirá una gestión plena y dedicada de la configuración del dispositivo inteligente, lejos del alcance de su mayoría de usuarios.

Experimentos relativos a la captación WiFi

Estudio 6.1.7: Rebote de tramas WiFi

Debido a que se capturan tramas propagadas por el aire por medio de WiFi, una misma trama emitida por un dispositivo es susceptible de ser capturada en varias ocasiones por el nodo de monitorización. Es por ello que en las Figuras del estudio anterior se presentaban varias detecciones del mismo tipo de trama en corto espacio de tiempo y distintas intensidades. Este fenómeno se puede observar en más detalle en la Figura 6.15, donde se observa como una única trama emitida ha sido detectada en más de 20 ocasiones en un periodo de algo más de 1 segundo en un entorno aislado y controlado.

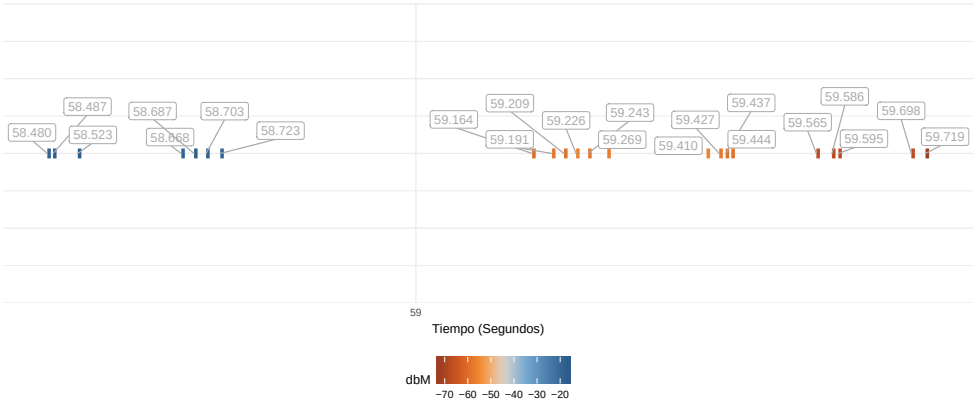


Figura 6.15
Detalle del efecto de rebote o reflexión de las tramas capturadas por WiFi.
Una misma trama de red emitida es detectada hasta 20 veces por el nodo de monitorización.

La naturaleza de los rebotes o refracciones de las comunicaciones Wi-Fi escapa del ámbito de estudio de esta tesis, sin embargo el sistema de monitorización propuesto tiene que lidiar con sus efectos.

Debido a que el sensor notifica al monitor las tramas que ha interpretado para que este actualice la ventana temporal del dispositivo (Sección 5.1.3), detectar varias veces la misma trama supone un incremento de la carga de trabajo del monitor, que tendrá que actualizar en multitud de ocasiones el instante de última detección del dispositivo, una por cada vez que la trama en cuestión es detectada de forma adicional.

Es deseable que el monitor sea capaz de determinar si la información que le facilita el sensor pertenece a un rebote de una trama del dispositivo ya detectada anteriormente o es una trama enviada nuevamente por el dispositivo. Se observa en la Figura 6.16 la cantidad de veces que la misma trama WiFi es detectada por un dispositivo inteligente en reposo (emitiendo una trama cada 10 segundos aproximadamente) en un entorno controlado.

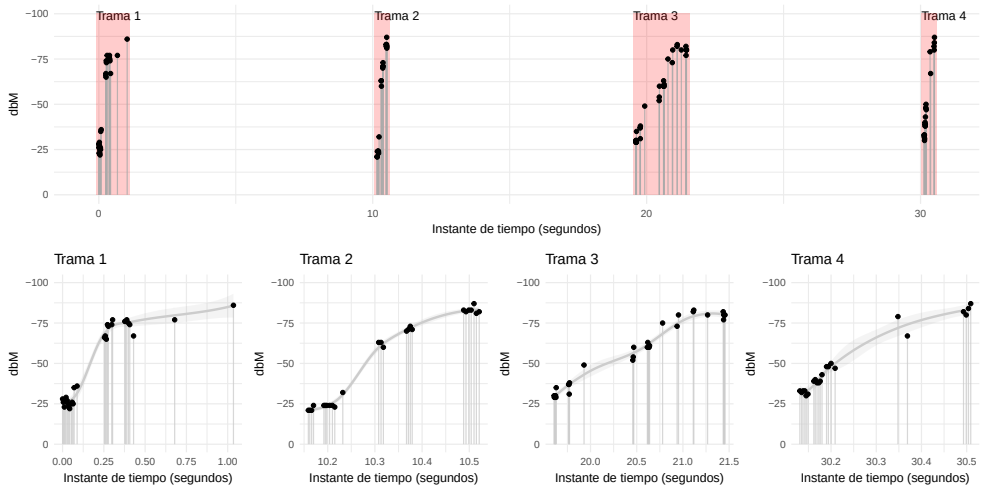


Figura 6.16
Efecto de rebote o reflexión de las tramas capturadas.
Para facilitar la interpretación, la gráfica de arriba muestra la información en la línea continua de tiempo, y las gráficas inferiores los valores preciosos de cada trama.

En promedio y entorno controlado, las tramas permanecen detectables en el medio hasta 1 segundo de tiempo después de la primera recepción, por lo que una de las posibilidades del monitor sería reducir la sensibilidad al segundo, para no considerar que tramas recibidas en instantes de tiempo inferiores al segundo. Sin embargo, tal y como es esperable, la intensidad de recepción proporcionada por el protocolo `radiotap` de las recepciones sucesivas de una misma trama empeora en cada recepción sucesiva. Por lo que otra posibilidad sería descartas las tramas que llegasen con una peor intensidad a la de la última detección.

Sin embargo, debido a que se espera que los dispositivos detectados se encuentren en movimiento, no puede ser esperable que captaciones posteriores se produzcan en mejores condiciones que las anteriores, y no por ello han de ser descartadas por el monitor. Además cualquiera de estas soluciones implica la inclusión de mecanismos software adicionales al monitor (Sección 5.6.5.5.2) que afectarían negativamente a su eficiencia[9]

Es por ello que de forma deliberada no se realiza ningún mecanismo adicional para evitar la captación y envío al monitor de las mismas tramas. Además, estos resultados anteriores son mediciones realizadas en entornos controlados. En la práctica, la cantidad de reincidencia de las tramas de red capturadas fruto de rebotes y refracciones es menor como se presenta en la Figura 6.17.

9 ↑No es objeto de esta tesis discutir como la inclusión de condicionales afecta negativamente al código de alto rendimiento.

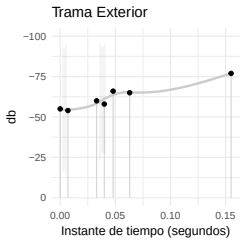


Figura 6.17
Efecto de rebote o reflexión de las tramas capturadas en entorno real.

Es por ello que si bien tiene que ser un fenómeno a considerar en la captación de comunicaciones inalámbricas, con el diseño funcional del sensor y monitor presentados en la Sección 5.6, no requiere ninguna atención excepcional.

Sin embargo, tendrá otras implicaciones en la interpretación de los pasos de dispositivos WiFi. Al contrario que en el caso de la detección Bluetooh donde un dispositivo detectado una única vez tendrá igual tiempo de inicio y fin de su enmarcación temporal, esto no ocurrirá en el caso del WiFi. O al menos, no es esperable. Debido a que aunque el dispositivo solo haya emitido una única comunicación, el nodo de monitorización puede haber capturado esa misma en varias ocasiones.

Experimentos relativos a la captación WiFi

Estudio 6.1.8: Tipo de dispositivos WiFi detectados

Si bien en contraposición de los dispositivos Bluetooth detectados en los que el propio protocolo proporciona información sobre la naturaleza del dispositivo, en el caso de los dispositivos WiFi la única información que se dispone sobre ellos es relativa al fabricante, proporcionada por su dirección MAC (Secciones 4.3.5 y 5.1.2).

Es por ello que aunque el estudio de dispositivos en el caso de WiFi disponga de menos información, sus resultados y análisis siguen resultando interesantes.

Uno de los riesgos de la captación de comunicaciones inalámbricas es que no únicamente dispositivos inteligentes las emplean y en el caso de WiFi, al requerir infraestructura, los dispositivos inteligentes son solamente una parte ínfima de los dispositivos a detectar.

Si bien los dispositivos de infraestructura son descartados debido a su carácter estático o perpetuo al permanecer mucho tiempo en las inmediaciones del nodo[10], es esperable que los dispositivos detectados pertenezcan en su mayoría de fabricantes de dispositivos inteligentes.

10 ↑Lo cual dependiendo del modo de funcionamiento del monitor, puede ocasionar que nunca sea notificado al servidor en el modo paso (Sección 5.6.1.3.1)

A fecha de escritura de la tesis se han detectado más de 18 millones de pasos WiFi, pertenecientes a más de un millón de dispositivos únicos[11] de los cuales ha sido posible determinar su fabricante en más de medio millón[12].

La Figura 6.18 recoge la propoción sobre los dispositivos WiFi detectados con fabricantes identificables de cada uno de los fabricantes.

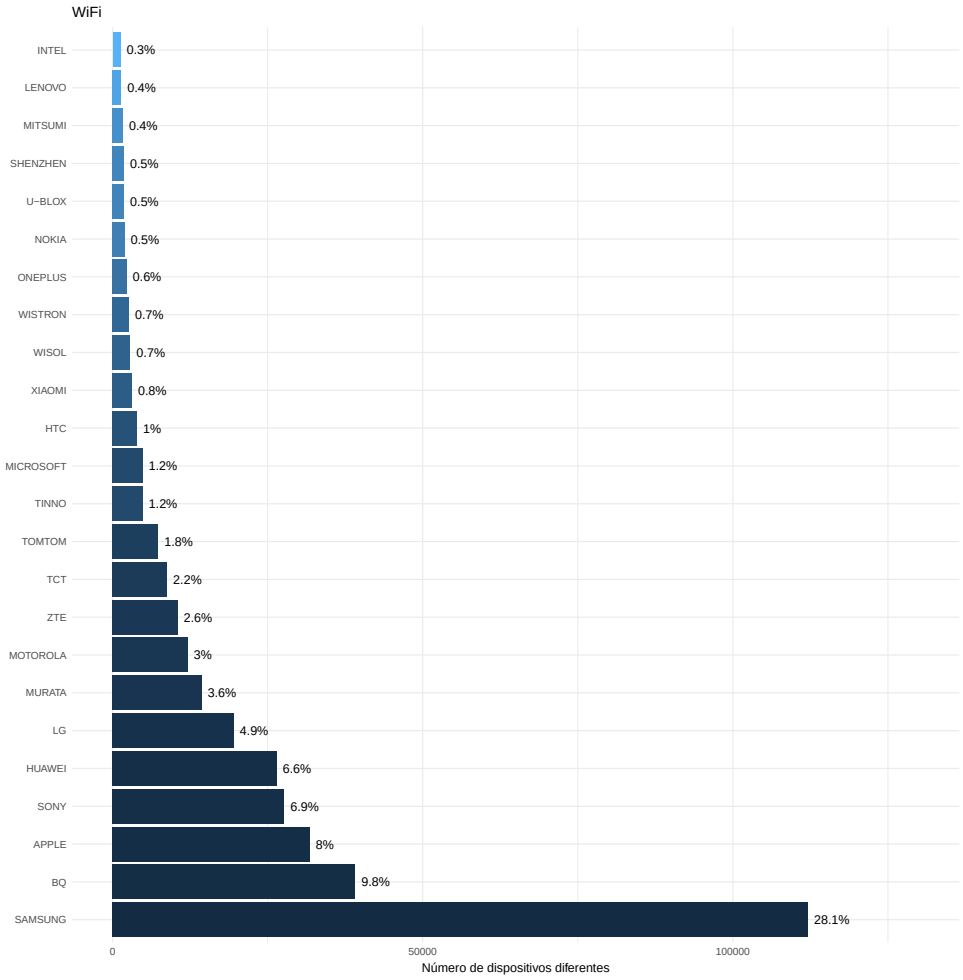


Figura 6.18
Proporción fabricantes de los dispositivos WiFi detectados identificables.

De entre los fabricantes detectados, la mayoría de ellos pertenecen a fabricantes de `smartphones` (SAMSUNG, BQ, APPLE, SONY, LG, MOTOROLA, ZTE, XIAOMI, MICROSOFT, HTC, XIAOMI, ONEPLUS, NOKIA) o dispositivos GPS emplazados en vehículos (TOMTOM). Destaca el porcentaje de dispositivos detectados pertenecientes a MURATA[13],TINNO[14] y WISOL[15],

11 ↑Por presentar las cifras absolutas 18 334 634 pasos de 1 097 518 dispositivos distintos.
12 ↑Esto es debido a las desactualizaciones del fichero que asocia los rangos de MACs con fabricantes, que no siempre muestra la información más actualizada posible. Lo cual en el escenario de dispositivos inteligentes donde se producen anualmente muchos lanzamientos de nuevos dispositivos, supone que la mayoría de los dispositivos de última generación no tienen su dirección MAC asociada con un rango-fabricante aún.
13 ↑https://www.murata.com/en-global/apps/mobile/smartphones
14 ↑http://www.tinno.com/home?lang=en
15 ↑http://www.wisol.co.kr/?ckattempt=3

sin embargo es necesario destacar que se tratan de algunos de los fabricantes de SoC más importantes del mundo, y muchas marcas minoritarias de smartphones relegan en ellos la fabricación de los componentes de telecomunicaciones, incluida la gestión de las direcciones MAC. De igual manera MITSUMI, por ejemplo, es la que provee las tarjetas de red a las videoconsolas portátiles más extendidas[16]

Además, resulta grato observar como la cantidad de fabricantes de dispositivos de infraestructura es mínimo, con ausencias de grandes fabricantes como CISCO, TP-LINK, NETGEAR o LINKSYS entre otros.

Experimentos relativos a la captación WiFi

Estudio 6.1.9: Intesidad RSSI de las comunicaciones detectadas

Cada nodo de monitorización que emplea captación WiFi puede ser configurado para descartar las tramas capturadas por debajo de una intensidad RSSI (proporcionada por la cabecera radiotap, Sección 5.1.2) determinada. El protocolo WiFi establece que las comunicaciones por debajo de los $-80dBm$ no resultan válidas para la transmisión, siendo inoperables a partir de $-100dBm$. Determinar a partir de que intensidad RSSI se descartan las tramas permite acotar (en cierta manera) el radio de acción aproximado del nodo de monitorización.

Si bien la relación entre la intensidad RSSI y la distancia está bien estudiada y modelizada [31, 159, 185, 316] normalmente se basa en la intensidad con la que un dispositivo móvil detecta los distintos puntos de acceso en una zona acotada geográficamente. En el caso del sistema de monitorización propuesto el modelo es al revés, es un nodo el que detecta un único dispositivo, por lo que no es posible la triangulación ni ningún mecanismo de posicionamiento preciso. La Figura 6.19 recoge la intensidad RSSI de detección mínima de una única trama emitida por un dispositivo inteligente en relación a la distancia al nodo de monitorización en distintos escenarios[17].

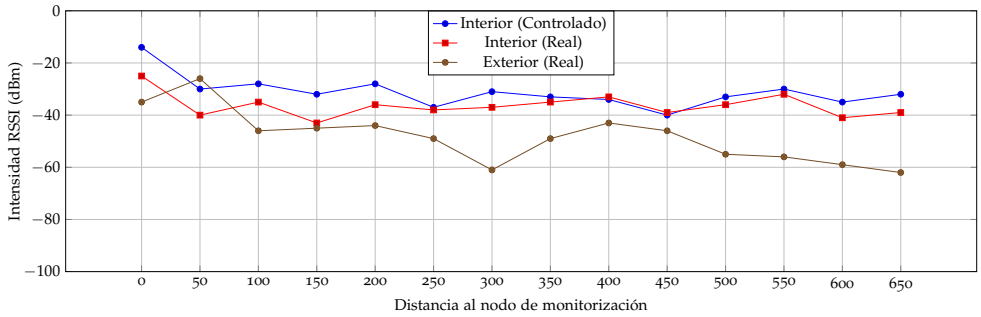


Figura 6.19
Relación entre la distancia al nodo de monitorización y la intensidad RSSI.

En entornos reales y distancias cortas, existen numerosos factores que influyen más que la distancia en la intensidad de recepción, como son los

16 ↑Incluyendo toda la familia de Nintendo DS o Nintendo Switch.
17 ↑Agradecer a Israel quien fue quien realizó la mayoría de las mediciones durante su beca de iniciación a la investigación, mostrándose aquí solamente una parte de las mismas.

obstáculos, la posición del dispositivo respecto al portador, la cantidad de dispositivos emitiendo, la presencia de puntos de acceso WiFi o las condiciones climáticas.

Si bien existe toda una línea de investigación a desarrollar que puede derivar de este aspecto, por ejemplo integrando varios nodos en una misma zona con el fin de posicionar mediante triangulación, no ha sido explotada en esta tesis.

En la Figura 6.20 se presentan los histogramas de las intensidades de primera y última detección de todos los pasos detectados por el sistema.

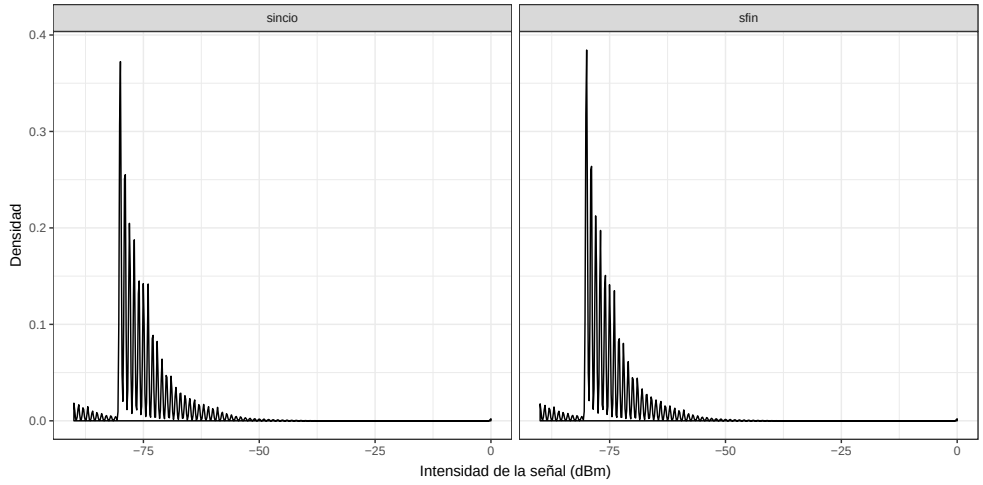


Figura 6.20
Densidad poblacional de las intensidades de captación de las tramas que determinan la ventana temporal de los dispositivos WiFi detectados por el sistema.

La mayoría de los pasos detectados se enmarcan entre los $-80dBm$ y $-50dBm$. El emplazamiento de cada nodo ha sido establecido intentando acotar de forma empírica el alcance, mediante la realización de mediciones.

Experimentos relativos a la captación WiFi

Estudio 6.1.10: Pruebas de stress y rendimiento

Las simulaciones de pruebas de stress del sistema de monitorización en el laboratorio se han visto superadas ampliamente una vez el sistema se enfrentó a escenarios reales.

En interiores el nodo de monitorización no ha tenido problemas en ser capaz de manejar más de 1 500 dispositivos simultáneos (Sección 5.1.5). En la Figura 6.21 se presenta la noche de uno de los nodos de monitorización colocado en interiores que más dispositivos llegó a detectar. El aforo máximo de la sala donde fue emplazado se marcaba en las 1 200 personas, con un aforo máximo de todo el local de 2 000 personas.

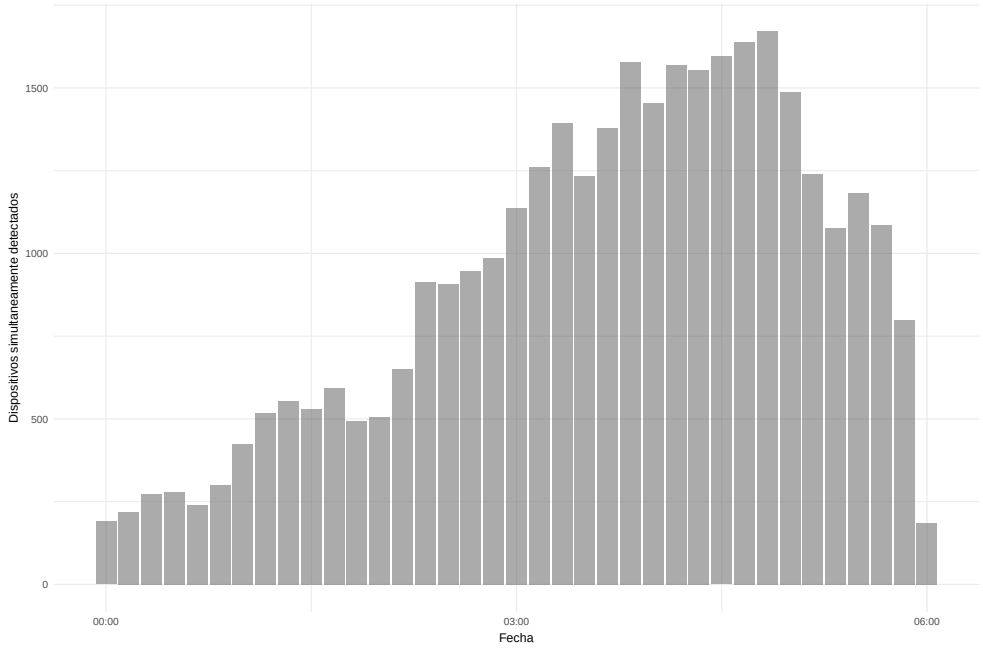


Figura 6.21
Máximo de dispositivos WiFi simultáneos manejados por el nodo de monitorización en interiores

En exteriores, el número de dispositivos máximos detectados de forma simultánea se encuentra cerca de los 800, que se recoge en la Figura 6.22 y pertenece a una manifestación, que serán estudiada en más detalle en los estudios de la Sección 6.3.4.

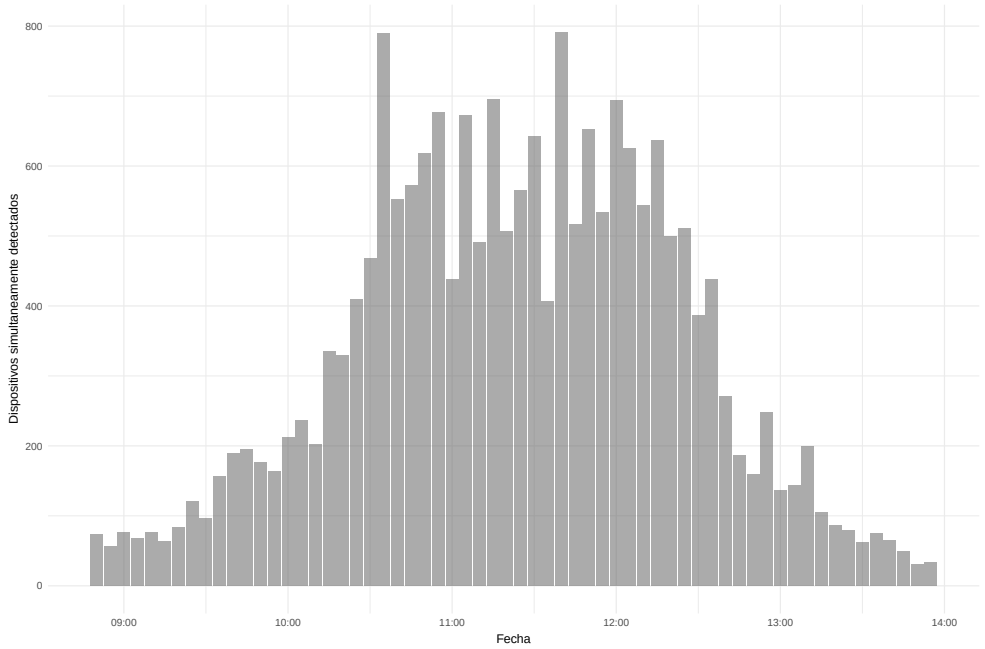


Figura 6.22
Máximo de dispositivos WiFi simultáneos manejados por el nodo en exteriores.

Debido a que estos escenarios reales sobrepasan a las pruebas de stress a los que fue sometido el nodo de monitorización en laboratorio, y la enorme cantidad de dispositivos detectados de forma simultánea, se evidencia que el prototipo de sistema de monitorización es capaz de lidiar con grandes cantidades de dispositivos simultáneos sin ningún tipo de impedimento.

Experimentos relativos a la captación WiFi

Estudio 6.1.11: Impacto en el sistema de las Macs de búsqueda aleatorias

En los últimos smartphones de Apple y en la nueva versión en desarrollo de Android [18] incorporan un mecanismo por el que la dirección MAC empleada en las búsquedas de redes WiFi no es la MAC Real.

Esto supone en principio un duro impacto negativo en el sistema, pues origina dos potenciales problemas. El primero, es que rompe la trazabilidad del dispositivo. El segundo, es que el número de dispositivos detectados se puede ver alterado si en cada búsqueda el dispositivo en corto tiempo emplea una MAC distinta.

Sobre la trazabilidad, según se ha observado la dirección MAC aleatoria se establece en el proceso de arranque del sistema operativo (IOS o Android) y no se cambia hasta que el dispositivo sufre un `hard reset` [90, 173, 279]. De esta manera, aunque la dirección MAC empleada en la búsqueda no sea la real, es siempre la misma hasta que el dispositivo no se reinicie completamente.

[90, 173, 279] How talkative is your mobile device?: an experimental study of Wi-Fi probe requests, A study of MAC address randomization in mobile devices and when it fails, Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms

18 ↑`https://www.androidpolice.com/2019/04/05/android-q-randomizes-mac-addresses-by-default-with-per-network-customization/`

Este tipo de reinicio habitualmente es el que se realiza cuando el dispositivo se queda sin batería por un largo periodo de tiempo. En dispositivo Android, incluso se ha documentado que ante un reinicio sin batería la mac asignada aleatoriamente es la misma.

De igual manera, como el dispositivo no cambia la dirección MAC de búsqueda hasta que se reinicia, empleará siempre la misma, por lo que no será detectado con distintas direcciones en cortos periodos de tiempo.

Aunque estos mecanismos de aleatorización buscan no permitir la trazabilidad del dispositivo, su aplicación más directa es no vincular la trazabilidad al tráfico de red de un determinado dispositivo. Así aunque un dispositivo con una MAC aleatoria se haya desplazado y haya sido detectado por varios puntos de acceso, cuando el dispositivo se conecte a uno de estos, hará uso de su dirección real. De esta manera, no se podrá vincular la dirección MAC a la que se le ha otorgado la dirección IP (y por tanto el tráfico de red asociado a esta) con la trazabilidad del dispositivo.

Aunque la aleatorización de las direcciones MAC puede suponer un contratiempo al sistema de monitorización propuesto, en su utilización actual por los dispositivos inteligentes no supone mayor contratiempo. La generación de una dirección MAC válida del rango del fabricante que no esté en uso por otro dispositivo, no es una acción trivial computacionalmente, por lo que no puede ser empleada para aleatorizar la dirección MAC de cada búsqueda.

Sin embargo, supone uno de los principales puntos de interés de la línea de investigación en el futuro, pues cualquier cambio puede influir en la naturaleza e implementación del sistema de monitorización propuesto.

Conclusiones

Debido a que WiFi no es un protocolo orientando a la detección de dispositivos, el nodo de monitoración realiza una captación de comunicaciones inalámbricas constantes. El sensor descarta las tramas y las procesa con tiempos inferiores a la precisión de medición ($1ms$).

El sistema de monitorización es capaz de procesar en laboratorio flujos constantes de tráfico de red de hasta 3000 tramas por segundo o $200KBps$. En entornos reales, en cambio, el flujo de red que se ha detectado es inferior a los $20KBps$, por lo que la eficiencia máxima del sensor, está por encima de los esperable en entornos reales. Esto es debido, a que los nodos de monitorización se emplazan en sitios donde el tráfico a capturar sea el de búsqueda de puntos de acceso, no a capturar tráfico de red con potencial ancho de banda.

Los dispositivos inteligentes tienen sistemas de ahorro de batería que les hacen interrumpir las comunicaciones cuando el dispositivo se encuentra en reposo. Para dispositivos en movimiento, el tiempo de búsqueda es inferior a los 5 segundos en los casos estudiados. Al detectar el dispositivo que se encuentra en reposo, se espacia esta búsqueda hasta los 10 segundos, incrementándose a lo largo del tiempo. Incluso desactivando la red WiFi y activando el modo avión, un dispositivo en reposo continuará buscando puntos de acceso cercanos.

Si bien el rebote de las tramas es un factor a tener en cuenta, debido a la capacidad de soportar flujos de tráfico más elevados al real, no requieren mayor consideración.

Los dispositivos WiFi detectados pertenecen a fabricantes que realizan dispositivos inteligentes, siendo el porcentaje de dispositivos de infraestructura detectado por debajo del 0.1 %.

Si bien la intensidad `RSSI` de detección puede ser empleada en el futuro, a las distancias que opera el nodo de monitorización, los factores externos influyen más que la distancia, no pudiéndose emplear para determinar de forma precisa la posición del dispositivo detectado.

Finalmente, en escenarios prácticos el nodo de monitorización ha demostrado ser capaz de lidiar con el tráfico de red de más de 1 500 dispositivos inteligentes WiFi de forma simultánea.

No existen por tanto impedimentos para la detección de los dispositivos inteligentes por medio de la captación de sus comunicaciones WiFi y su procesamiento en tiempos cercanos al real por un dispositivo de bajo coste como el presentado como prototipo y herramienta de estudio de esta tesis.

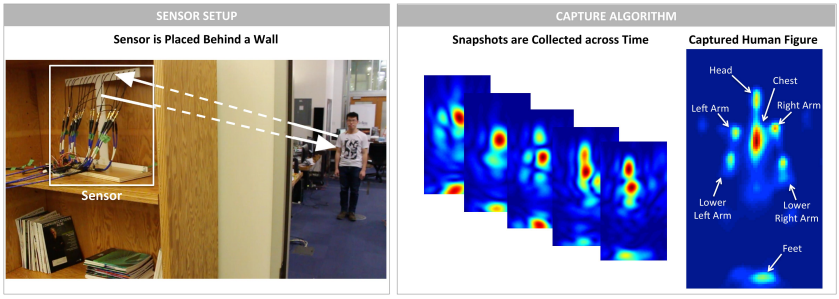
6.1.3 *Comparativa entre nodos emplazados en la misma zona*

Para estudiar la fiabilidad del nodo de monitorización en la captación de comunicaciones inalámbricas, se emplazan 3 nodos en la misma zona y se comparan los resultados obtenidos entre los distintos nodos.

Si bien está estudiado que variaciones muy cortas entre las antenas en la banda de frecuencia empleadas por Bluetooth y WiFi pueden variar tanto en la detección o no de ciertas tramas como en su intensidad [2], es deseable a nivel de sistema de monitorización los resultados sean coherentes entre distintos nodos emplazados en la misma zona.

[2] See through walls with WiFi!

Para ello se emplazan las antenas de captación de 3 nodos distintos en un corto espacio[19]. Se configura cada nodo para que no identifique a los otros nodos, de forma que no existan interferencias entre ellos. La Figura 6.23(b) recoge la disposición de las antenas cuando se realizó la medición WiFi. Tanto las distancias como distribución están basados en estudios que emplean las comunicaciones WiFi para ver detrás de paredes [2] como se recoge en la Figura 6.23(a)



(a) Visión a través de paredes en la investigación de Fadel Adib.

(b) Emplazamiento de las antenas en la misma distancia en laboratorio.

Figura 6.23
Para la realización de la comparativa se realiza un supuesto similar al empleado para determinar objetos detrás de paredes, por lo que se espera que el tráfico capturado sea distinto en cada antena, pero que los resultados a nivel de sistema de monitorización no lo sean.
Figura (a) Fuente: See through walls with WiFi! [2]

El objetivo del estudio es comprobar que el sistema de monitorización propuesto es confiable en cuanto a sus resultados obtenidos.

19 ↑5 centímetros, por lo cual 10 centímetros de extremo a extremo.

Comparativa entre nodos emplazados en la misma zona

Estudio 6.1.12: Comparativa varios nodos Bluetooth

En el caso del Bluetooth el resultado es del 100 % de dispositivos detectados, pues es la ventaja de que el protocolo provea de mecanismos para la búsqueda de dispositivos. En el escenario donde se han realizado las pruebas, debido al corto alcance aparente del Bluetooth, todos los dispositivos eran detectado prácticamente al unísono cuando se encontraban en las inmediaciones o se activaba un modo que permitiese su descubrimiento.

Sobre la variación de los instantes de tiempo de detección, resultaban inferiores al segundo en la totalidad de las mediciones.

Comparativa entre nodos emplazados en la misma zona

Estudio 6.1.13: Comparativa varios nodos WiFi

En el caso del WiFi, a pesar de la escasa distancia entre las antenas se encuentran variaciones en cuanto a las tramas de red detectadas, debido a que se está capturando tráfico de red en modo monitor.

La Tabla 6.4 recoge el número pasos totales detectadas por el nodo y el número dispositivos detectados.

Tabla 6.4
Comparativa entre nodos emplazados en mismo sitio en la captación WiFi.

NODO	PASOS	DISPOSITIVOS DISTINTOS
A	335	50
B	359	59
C	346	54

El nodo A ha sido el que menos pasos y dispositivos ha detectado, sin embargo los 50 dispositivos detectados en el nodo A han sido detectados en los nodos B y C. Si bien el nodo B y C han detectado más pasos y dispositivos, se considera que es debido a la posición relativa de las antenas.

El nodo B ha capturado 24 pasos más que el nodo A, de los cuales 21 pertenecen a 9 dispositivos no detectados por el nodo A. El nodo C ha capturado 9 pasos más que el nodo A, 4 de los cuales pertenecen a 4 dispositivos distintos.

El nodo B respecto al nodo C ha capturado 13 pasos más, pertenecientes a 4 dispositivos distintos[20].

Debido a que las tramas de red capturadas no se almacenan, no es posible determinar si son las mismas las que han sido detectadas por los distintos nodos. Pero si es posible estudiar su instante de captura y su ventana temporal. En la Figura 6.24 se han representado los pasos (Sección 5.1.3) de los 50 dispositivos detectados a lo largo del tiempo. Cada dispositivo ha sido identificado por los 3 últimos digitos hexadecimales de su dirección MAC.

20 ↑Existe 1 dispositivo detectado por C que no ha sido detectado ni por A ni por B.

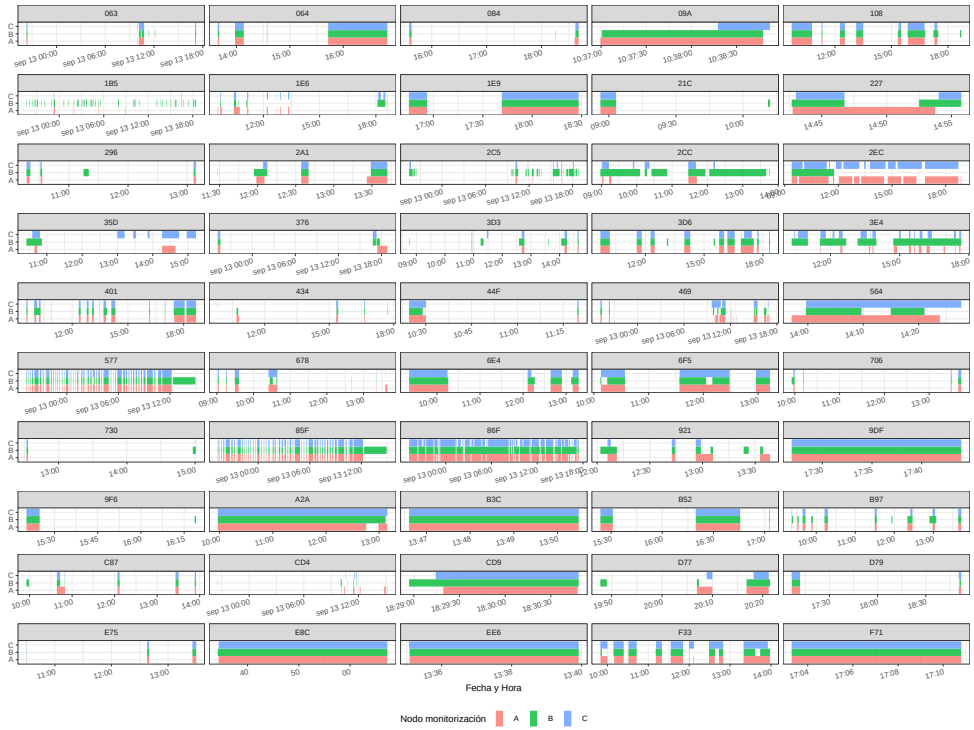


Figura 6.24
Comparativa entre distintos nodos capturando tráfico WiFi.

Los resultados de monitorización entre los 3 nodos resultan prácticamente idénticos, salvo en el caso de los dispositivos `35D`, en que el nodo B no lo detecta durante un largo periodo de tiempo, y el dispositivo `2CC` que el dispositivo A no lo detecta.

Debido a que los nodos de monitorización están configurados para descartar las tramas que lleguen con una intensidad `RSSI` menor de $-80dBm$, se estudia la intensidad con la que se han recibido las comunicaciones inalámbricas capturadas que determinado el instante de detección del dispositivo y por tanto su paso. Este estudio se recoge en la Figura 6.25.

Se observa que existe una correspondencia directa entre las discrepancias de la duración o existencia del paso, con la intensidad de señal `RSSI` con la que el paso ha sido detectado. De esta manera, por ejemplo en el dispositivo `35D` se observa que tanto el nodo A como el nodo C lo están detectando por valores muy cercanos a los $-80dBm$, por lo que es posible que el nodo B no lo detecte debido a que las tramas capturas estén llegando con valores inferiores a estos $-80dBm$ configurados.

Sin embargo, cuando los valores de intensidad `RSSI` de las comunicaciones capturadas se encuentran en el mismo rango, la duración y existencia de los pasos entre los 3 nodos distintos estudiados es prácticamente perfecto, con discrepancias en los tiempos inferiores al segundo[21].

21 ↑La sincronización de los relojes de los nodos de monitorización se aborda en las Secciones 5.4.6 y 5.5.4

Figura 6.25
Comparativa de la intensidad RSSI entre distintos nodos capturando tráfico WiFi.

Conclusiones

En el caso de Bluetooth, al ser un protocolo orientado a la detección de dispositivos tiene una tasa de correspondencia del 100 % entre los tres nodos.

En el caso de WiFi, esta correspondencia no es perfecta, debido a que las tramas cercanas al umbral de filtrado pueden ser consideradas en alguno de los nodos, mientras que los otros la estén descartando. En caso contrario, existe una correspondencia casi exacta entre los resultados de monitorización de los 3 nodos estudiados.

De esta forma, aunque la captación de las comunicaciones inalámbricas pueda no coincidir en nodos emplazados en la misma zona, los resultados de monitorización si resultan coincidentes, salvo en ocasiones en el que la comunicación se capte con intensidades RSSI muy cercanas al valor umbral.

No existen motivos detectados que hagan desconfiar de la veracidad de los resultados de la captación de las comunicaciones inalámbricas.

6.1.4 *Influencia del posicionamiento del nodo y sus antenas*

La posición del nodo de monitorización y sus interfaces de red de captura resulta crítica, pues emplazalo en zonas aisladas o que no permitan capturar suficiente tráfico de red puede hacer peligrar la viabilidad del sistema de monitorización para ese escenario.

Si bien el concepto de caja de Faraday no es ajeno a la comunidad científica, el desconocimiento general de la atenuación del metal en las comunicaciones inalámbricas ha requerido en ocasiones tener que demostrar empíricamente lo que en ocasiones resultaba evidente.

Influencia del posicionamiento del nodo y sus antenas

Estudio 6.1.14: Emplazamiento a pie de calle o en semáforo

En este estudio se compara los resultados de la modificación de la posición de emplazamiento de un nodo en un entorno urbano. La Figura 6.26 muestra la ubicación sobre plano del nodo de monitorización.

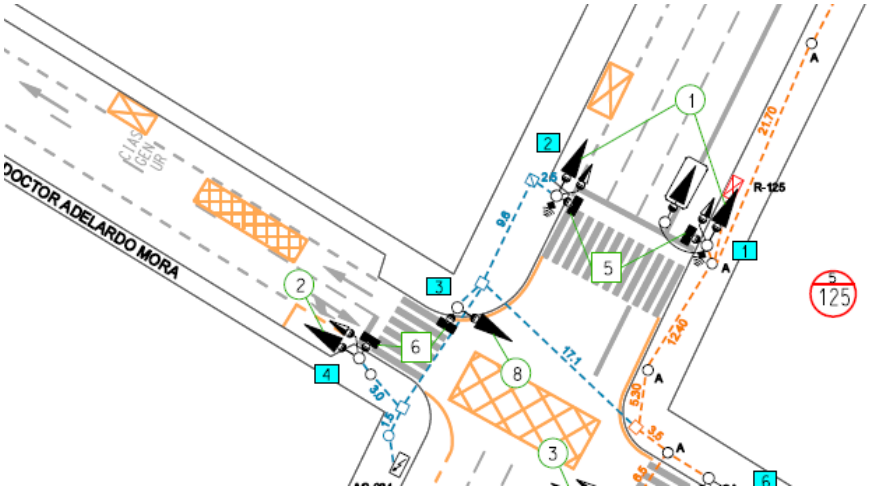


Figura 6.26
Emplazamiento del nodo de monitorización. Originairamente emplazado en el armario 1 (representado por el rectángulo azul 1) y posteriormente ubicado en el interior del semáforo (representado por el rectángulo verde 5).

Originariamente el dispositivo fue emplazado dentro de un armario de conexión metálico (Figura 5.38) y no ofrecía buenos resultados de monitorización, tal y como resultaba evidente.

Al ser emplazado dentro de un semáforo (Figura 5.39), pero con la frontal orientada hacia la calzada de plástico, los resultados de monitorización fueron mucho más satisfactorios, como muestra la Figura 6.27 obteniéndose un incremento del número de dispositivos detectados de 10 veces más.

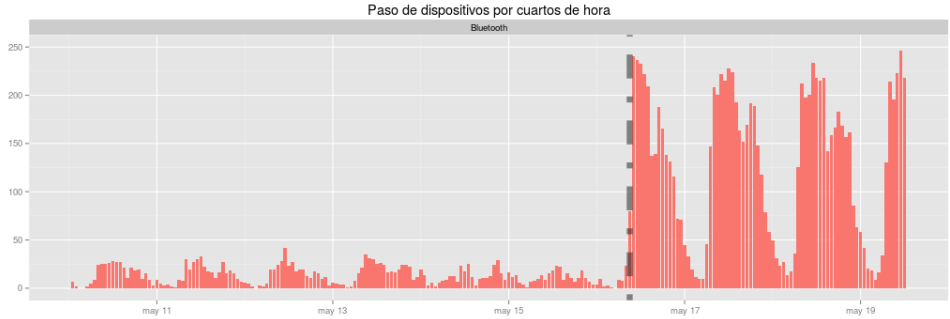


Figura 6.27
Pasos de dispositivos detectados en armario frente a semáforo. La línea punteada marca el instante de tiempo en el que se realizó el cambio, siendo la izquierda el emplazamiento dentro del armario y la derecha la detección dentro del semáforo.

Si bien en ocasiones convencer de cual es el emplazamiento óptimo del nodo de monitorización no resultaba fácil, su influencia la cantidad de datos obtenibles es más que evidente.

Influencia del posicionamiento del nodo y sus antenas

Estudio 6.1.15: Emplazamiento en carretera

En escenarios interurbanos la proximidad a la carretera resulta mandataria. En este estudio se comparan 3 escenarios posibles de emplazamiento del nodo de monitorización en carretera.

Los nodos `1010` y `1020` se encuentran emplazados dentro de edificios a pie de carretera (menos de 10 metros) donde se almacenan los equipos de telecomunicaciones de la Dirección General de Tráfico. Si bien se encuentran cerca de la carretera, se posicionan en un lateral de esta. Además, están cerrados por una puerta metálica de varios milímetros de grosor, y sus paredes de hormigón tienen un ancho de unos $20cm$.

Los nodos `1110`,`1120`,`1130` y `1140` se encuentran emplazados dentro de un armario de mantenimiento sobre la calzada. El armario es metálico y dispone de ventiladores para la refrigeración que funcionan por electroimanes.

Finalmente los nodos `1150`, `1160` y `1170` se encuentran emplazados en un puente de señalización sobre la carretera, como se muestra en la Figura 5.41.

Un mal emplazamiento de un nodo de monitorización puede ocasionar que este no detecte nada, no debido a que no estén circulando dispositivos inteligentes detectables, sino porque la comunicación inalámbrica ni tan siquiera es capturada por el nodo.

La Figura 6.28 compara las magnitudes de pasos de vehículos detectado por ambos sensores por los distintos nodos de monitorización emplazados en distintos escenarios.

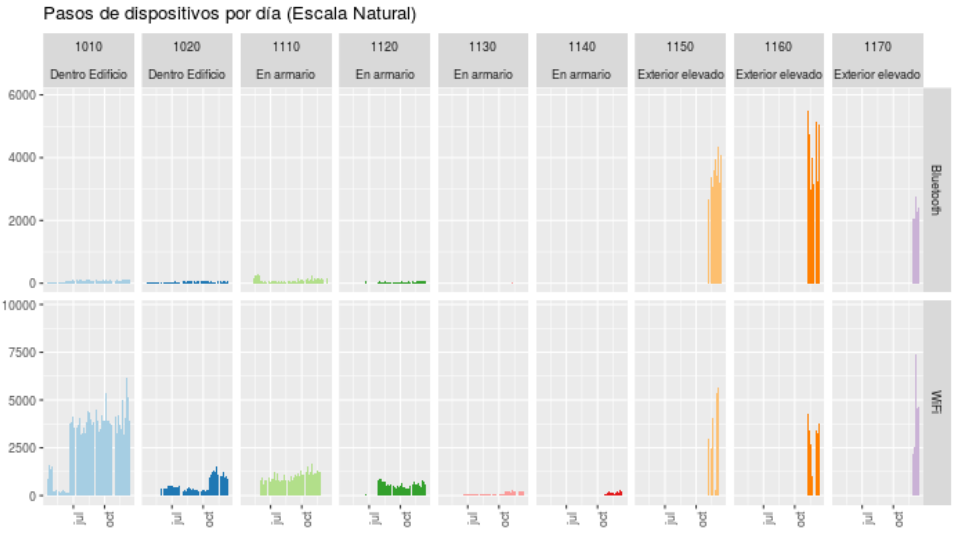


Figura 6.28
Comparativa de la posición de los nodos en carretera

Si bien WiFi parace comportarse un poco mejor, esto es debido a la mejor antena y captación que ofrece el prototipo. Sin embargo, el emplazamiento sobre plataforma elevada a pie de carretera en Bluetooth consigue detectar magnitudes considerables de vehículos.

Si bien puede ser discutido que la gráfica presentada es de distintos periodos de tiempo y emplazamientos. El nodo 1010 y el nodo 1160 se encuentran emplazados en el mismo punto kilométrico con una distancia entre ellos inferior en línea recta sobre el plano a los 4 metros. Sin embargo, la cantidad de dispositivos detectados es de varios cientos más, como se presenta en la Figura 6.29

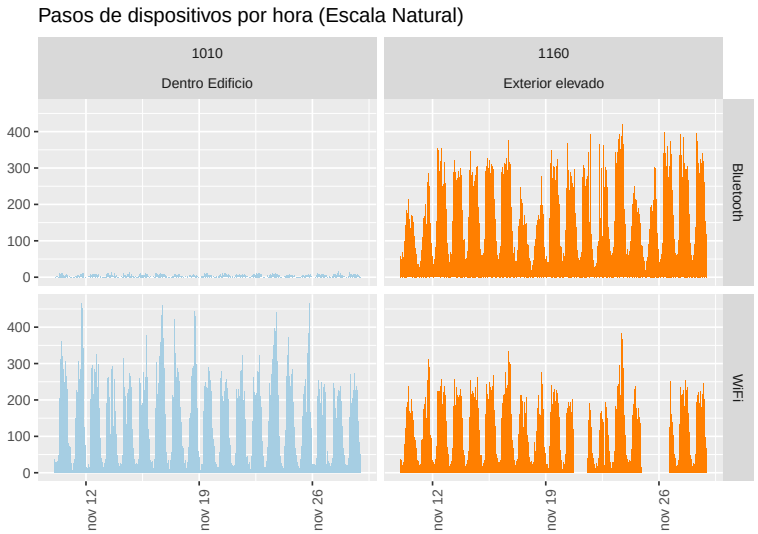


Figura 6.29
Comparativa entre dos nodos emplazados en el mismo punto kilométrico.

Resulta reseñable como en el caso de WiFi del nodo 1010 supera en ocasiones a los valores

del nodo 1160. Esto es debido a las sensibilidades de los nodos están configuradas de forma distinta, descartando el nodo 1010 las tramas por debajo de los $-100dBm$ y el nodo 1160 las tramas por debajo de $-80dBm$. De no ser asi, el nodo 1010 no detectaría prácticamente nada. Y el nodo 1160 se acota únicamente al tráfico real circulando por debajo suya.

Conclusiones

Si bien en el estudio de la posición de los nodos y por tanto sus antenas podría ser mucho más extendido, esta sección viene unicamente a demostrar que las comunicaciones inalámbricas de los dispositivos inteligentes existen allá donde estos se mueven, y que su no detección en la mayoría de las ocasiones abordadas en la aplicación del sistema de monitorización de esta tesis, son debido a malos emplazamientos del nodo, no debido a la invalidez del sistema.

El correcto emplazamiento de los nodos de monitorización resulta imprescindible para el funcionamiento del sistema, pues emplazarlo en una zona electromagnéticamente aislada puede impedir la captación de las comunicaciones inalámbricas emitidas por los dispositivos inteligentes.

Sin embargo, frente a las alternativas existentes, resulta menos restrictivo debido a que no requiere ser instalado sobre la calzada como los tubos neumáticos (Sección 3.4.1) o las espiras magnéticas (Sección 3.4.2). Ni requiere una visibilidad perfecta como en el caso de los sitemas basados en imágenes de vídeo (Secciones 3.4.3, 3.5.2 y 3.5.3)

Debido a que la implantación de los nodos no requiere de obra civil, el nodo puede ser emplazado y puesto en funcionamiento en minutos. Esto hace factible cambiar la posición del nodo en caso de que su ubicación actual no esté obteniendo buenos resultados. Lo cual puede variar sustancialmente los resultados de la captación, como se ha presentado en los estudios de esta sección.

6.2 HIPÓTESIS II: SOBRE EL ESTUDIO DE LA MOVILIDAD DE LOS DISPOSITIVOS INTELIGENTES Y LA ADECUACIÓN AL MOVIMIENTO DE PERSONAS Y VEHÍCULOS

Un vez demostrada la viabilidad de la captación de las comunicaciones inalámbricas emitidas de forma inadvertida por los dispositivos inteligentes en la Sección anterior, en esta se aborda la monitorización de los dipositivos y su aplicación a la detección de personas y vehículos.

Se presenta en primer lugar los resultados de una encuesta realizada con el fin de conocer los hábitos de empleo de los dispositivos inteligentes en la conducción y el empleo de las distintas interfaces inalámbricas del dispositivo por parte de los usuarios.

Se exponen distintos escenarios en los que se han realizado los distintos estudios y análisis. Con el fin de evitar repetir estudios en este documento[22], se presentará una única cada estudio, con independencia de que haya sido repetido en más de un escenario.

22 ↑Ya de por si bastante extenso

6.2.1 *Encuesta sobre el uso de los dispositivos móviles y comunicaciones inalámbricas durante la conducción.*

Con el fin de determinar el uso de las redes inalámbricas empleadas por los dispositivos móviles por parte de los usuarios, se realiza una encuesta online que es responda por más de 600 personas.

Encuesta sobre el uso de los dispositivos móviles y comunicaciones inalámbricas durante la conducción.

Estudio 6.2.1: Información general sobre la población encuestada

Al lanzarse en un entorno universitario, se cuestiona la edad de los implicados en la encuesta (Figura 6.30).

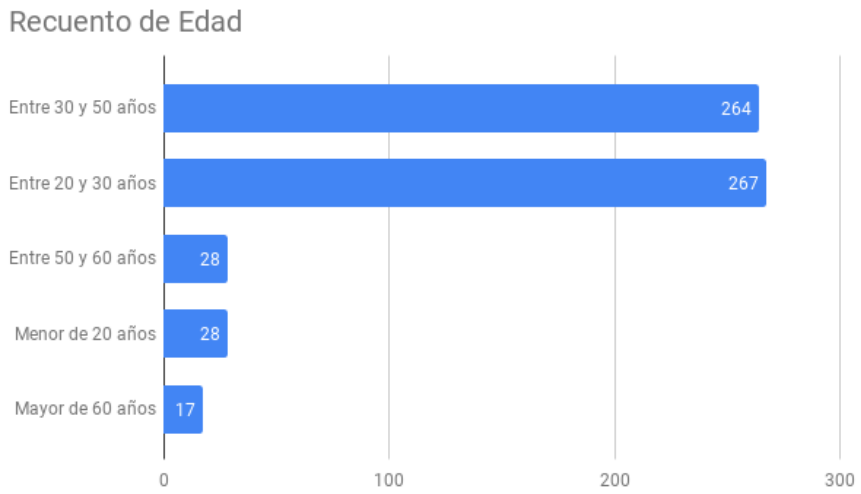


Figura 6.30
Edado de los encuestados.
El grueso de la población encuestada se encuentra entre los 20 y 50 años de edad.

Se cuestiona también el sexo de los encuestados, si conocen el sistema operativo de su móvil o el fabricante del mismo. Sin embargo, estos resultados no resultan relevantes para el objetivo de esta encuesta en la tesis.

Encuesta sobre el uso de los dispositivos móviles y comunicaciones inalámbricas durante la conducción.

Estudio 6.2.2: Cuestiones relativas al uso de manos libres

Debido a que el objeto principal de estudio de la encuesta es sobre el uso de los dispositivos móviles durante la conducción, se le cuestione a la población encuestada si son conductores habituales (Figura 6.31)

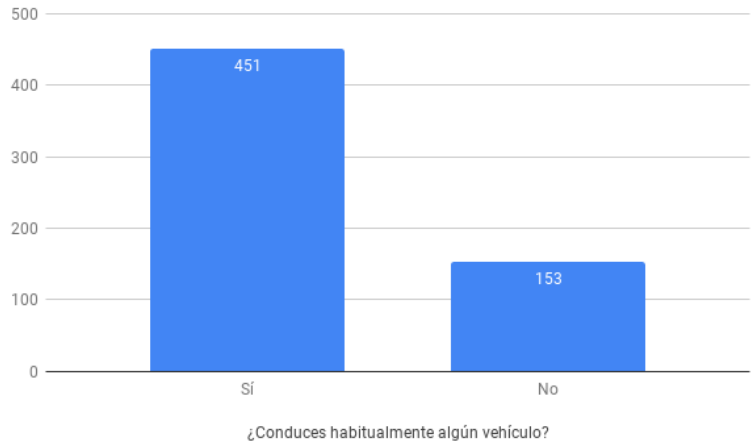


Figura 6.31
Conducción habitual de los encuestados.
Cerca del 75 % de los encuestados se declaran conductores habituales.

A la población encuestada se le pregunta si sus vehículos disponen de un dispositivo "manos libres"(Figura 6.32), pues resulta más fácil plantear esta cuestión que cuestionar directamente si sus vehículos disponen de conexión Bluetooth.

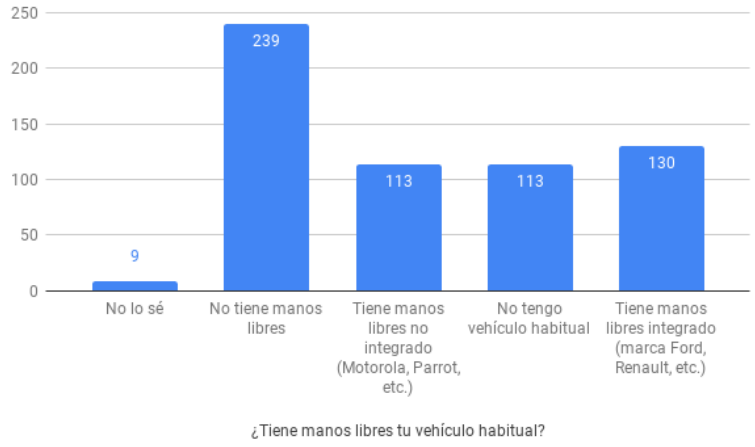


Figura 6.32
Disponibilidad de manos libres por los encuestados.

Se condiciona el resultado de esta pregunta únicamente a aquellos que han declarado ser conductores habituales (Figura 6.33).

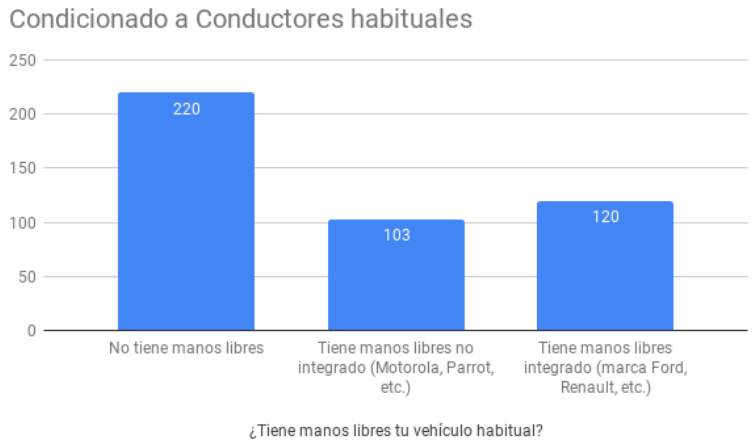


Figura 6.33
Disponibilidad de manos libres por los encuestados que se declaran conductores habituales

Según los resultados el 50 % de los encuestados que se declaran conductores habituales disponen de manos libres en su vehículo. Estudiando los rangos de edad (Figura 6.34), se refleja que los encuestados entre 30 y 50 años tienen manos libres en el 57 % de los casos, frente al 43 % de los encuestados entre 20 y 30 años.

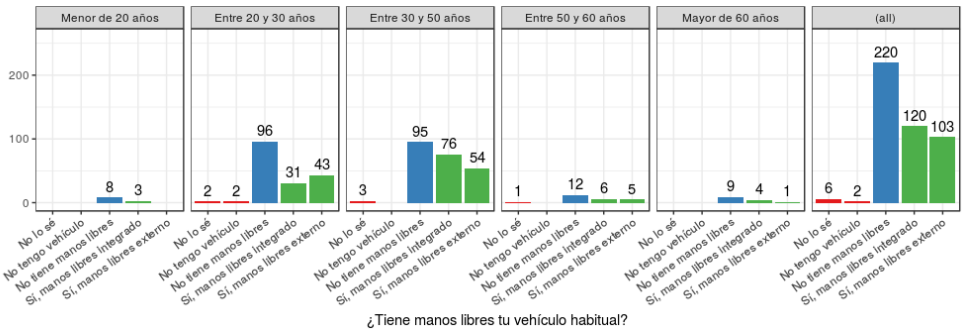


Figura 6.34
Influencia de la edad de los encuestados en la disponibilidad de dispositivo manos libres en sus vehículos.

Se le cuestiona a la población si hacen uso del manos libres cuando conducen, condicionado a únicamente los conductores habituales (Figura 6.35). Cerca del 50 % nunca usan dispositivos de manos libres mientras conducen, mientras que de los restantes, la mitad sólo lo usa algunas veces.

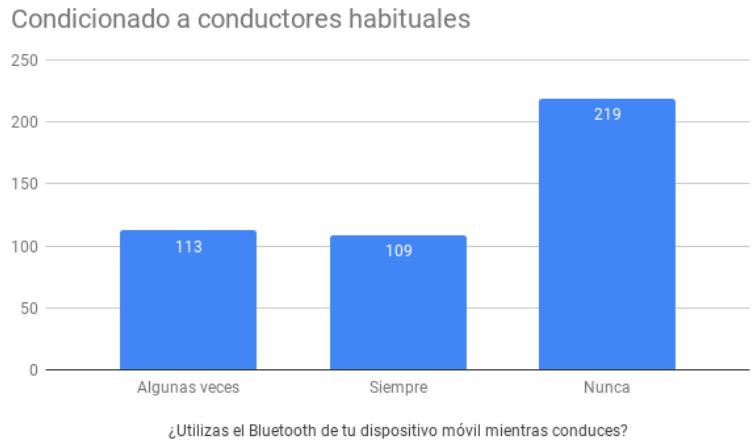


Figura 6.35
Uso de manos libres por los conductores habituales encuestados.

Debido a que no todos los encuestados han manifestado disponer de manos libres, se condiciona la respuesta a aquellos que disponen de manos libres, con el fin de ampliar el estudio (Figura 6.36). Sólo un 10 % de los encuestados, pese a disponer de manos libres en su vehículo, no hace uso de él. El 46 % de los encuestados que disponen de dispositivo manos libres en su vehículo hacen siempre uso de él.

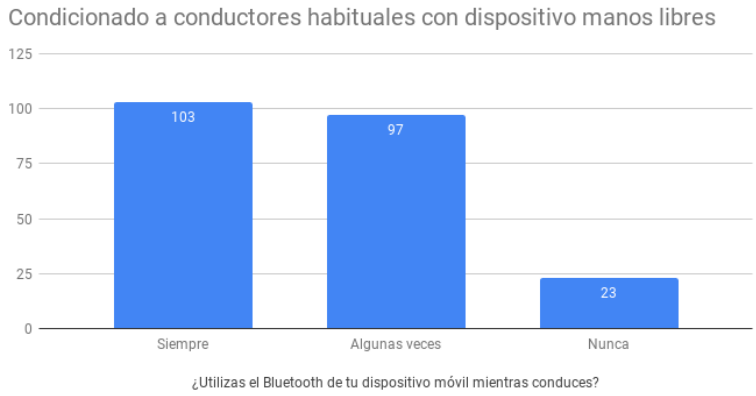


Figura 6.36
Uso de manos libres por los conductores habituales poseedores de dispositivos manos.

El uso de manos libres entre los conductores parece bastante arraigado entre sus poseedores. El grueso de conductores que carecen de dispositivo manos libres han resultado ser el perfil más joven encuestado, lo que parece evidenciar que su empleo no está reñido con la edad de los conductores, sino con factores inherentes a la edad[23].

23 ↑Tal vez factores económicos, o la antigüedad o gama baja de la flota conducida por los más jóvenes.

Encuesta sobre el uso de los dispositivos móviles y comunicaciones inalámbricas durante la conducción.

Estudio 6.2.3: Cuestiones relativas a las comunicaciones inalámbricas

Adicionalmente, se les cuestiona a la población sobre que uso hacen de las comunicaciones inalámbricas de sus dispositivos móviles, concretamente sobre si activan y desactivan constantemente las interfaces.

En el caso del Bluetooth (Figura 6.37), un 75 % de los encuestados declara apagar el Bluetooth cuando no lo está empleando. Dato que resulta sorprendete

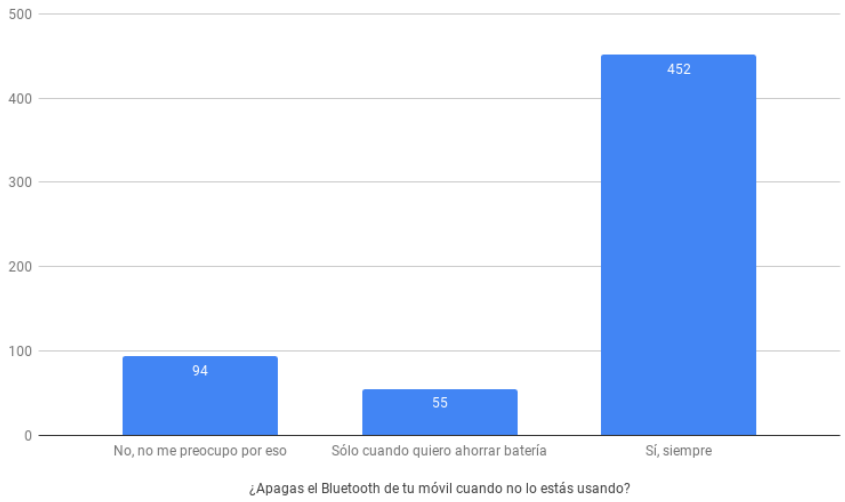


Figura 6.37
Hábitos con el Bluetooth de los encuestados relativos a su activación y desactivación.

En el caso del WiFi (Figura 6.38), aproximadamente tan solo 1 de cada 3 encuestados declara apagar el WiFi cuando no lo está empleando. Otro declara apagarlo sólo cuando necesita ahorrar batería. Y el otro restante declara no preocuparse por eso.

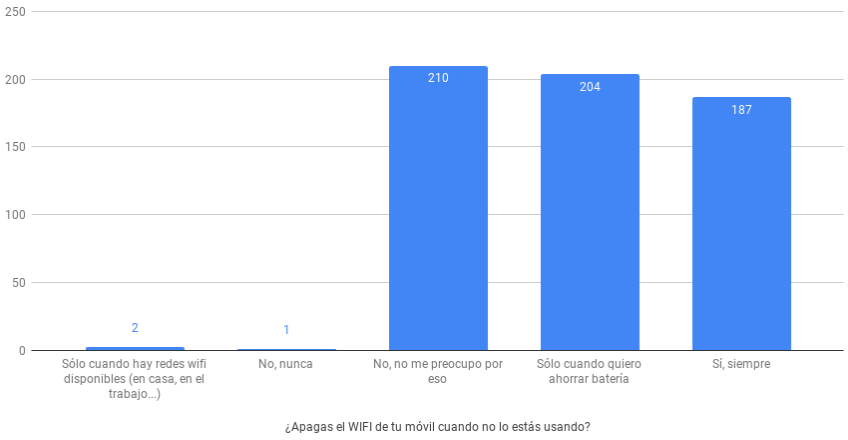


Figura 6.38
Hábitos con el WiFi de los encuestados relativos a su activación y desactivación.

Tabla 6.5
Hábitos combinados de la gestión de las conexiones Bluetooth y WiFi de los encuestados.

WiFi \ BT	No, no me preocupo por eso	Sí, siempre	Solo cuando quiero ahorrar batería
No, no me preocupo por eso	63	137	10
Sí, siempre	11	169	7
Solo cuando no hay redes WiFi disponibles	0	0	0
Solo cuando quiero ahorrar batería	20	146	38

Para los encuestados, la gestión de la conexión Bluetooth resulta más importante que la de WiFi. Los posibles factores que expliquen este comportamiento son que asocien a la conexión Bluetooth un mayor consumo energético o un uso directo y gestionado con un dispositivo concreto. Frente a la libertad de WiFi, que consideran que no se está empleando si no hay se está conectado a una red concreta.

Estos resultados contrastan con la implantación masiva de dispositivos que hacen uso de Bluetooth LE, como pulseras cuantificadores, smartwatches o auriculares. Sería deseable que una nueva versión de la encuesta realizada en el futuro, arrogase mejores resultados en cuanto a la gestión de Bluetooth.

Conclusiones

La implantación de conexiones Bluetooth en los dispositivos manos libres de los vehículos es importante en la flota actual. Cerca de la mitad de los encuestados que se declaran conductores habituales dispone de un dispositivo manos libres en su vehículo. De estos, solamente el 10 % no hace un uso del dispositivo de manos libres, pese a disponer de él.

Tan solo el 30 % de los encuestados declara apagar siempre su conexión WiFi cuando no la está usando, frente al 33 % que lo hacen únicamente cuando necesita ahorra batería y el 35 % que no se preocupa por hacerlo. Sin poner en duda la veracidad de respuesta de los encuestados,

6.2.2 *Análisis de la congestión del tráfico urbano*

La congestión de las vías de tráfico es uno de los principales problemas en las urbes europeas (Sección 2.2). Debido a que el sistema propuesto es capaz de enmarcar las estancias de los vehículos y mediante del empleo de dos o más nodos se pueden obtener los tiempos de desplazamiento reales requeridos para desplazar entre las zona de ambos nodos, se propone su empleo para el análisis de la congestión del tráfico.

Gracias a la colaboración con el Área de Movilidad del Ayuntamiento de Granada, se emplazan dos nodos de monitorización en una calle muy concurrida por los vehículos de la Ciudad de Granada, que recoge mucho tráfico del norte de la ciudad ya que es la artería muy céntrica de la ciudad que permite desplazarse entre distintos barrios de la misma. En la Figura 6.39 se recoge la posición de los nodos.

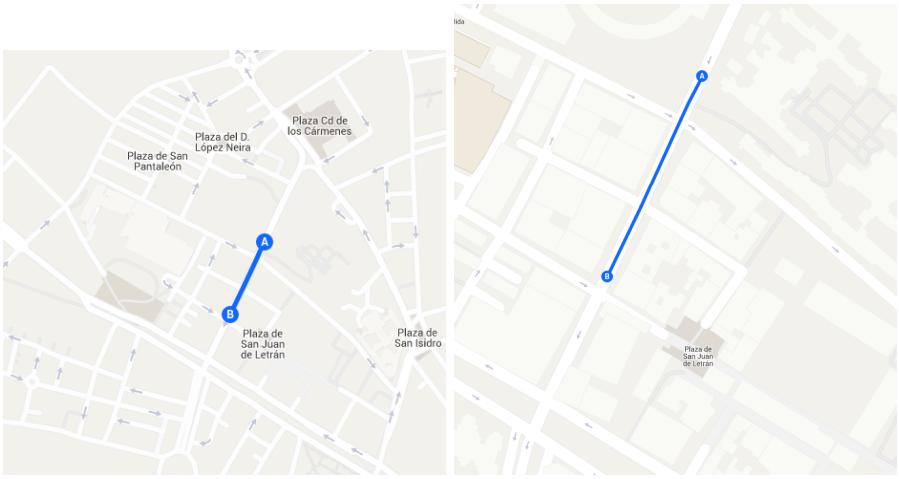


Figura 6.39
Localización de los 2 nodos en la calle congestionada

El ayuntamiento facilita información histórica sobre mediciones que han realizado en dicha calle con aforadores portátiles[24] ofreciendo información de Enero a Abril de 2015 sobre el paso de vehículos. Los datos son obtenidos por una espira magnética (Sección 3.4.2) y se presentan en la Figura 6.40.

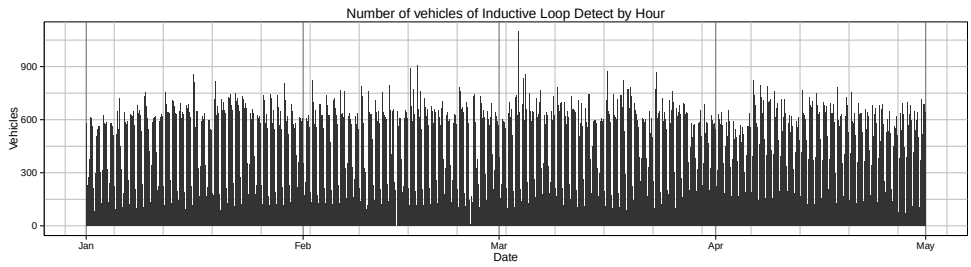


Figura 6.40
Vehículos detectados por la espira magnética por hora.

24 ↑Lamentablemente, los cuales no pudieron ser emplazados durante durante el tiempo de funcionamiento, lo cual requirió que se realizasen otra serie de análisis derivados.

Los datos históricos del ayuntamiento se comparan con los obtenidos por el sistema propuesto mediante la captación de comunicaciones Bluetooth entre los dos nodos durante un periodo del 11 de Enero al 29 de Febrero de 2016. El número de vehículos detectado se presenta en la Figura 6.41.

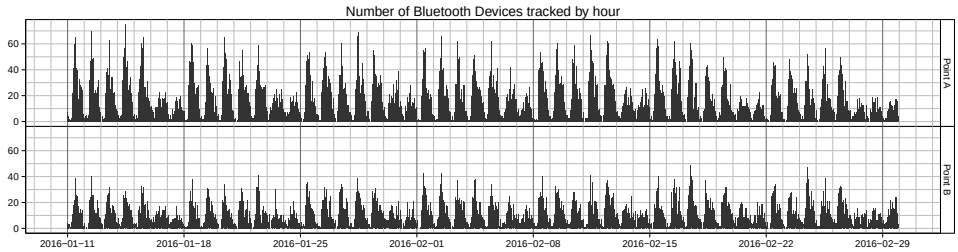


Figura 6.41
Vehículos detectados por la captación Bluetooth

Debido a que no se tratan de datos de las mismas fechas, se tienen que emplear otros factores para el análisis del sistema propuesto con el fin de validarlo.

Análisis de la congestión del tráfico urbano
Estudio 6.2.4: Análisis del flujo de tráfico

Si bien el sistema propuesto no es un sistema exhaustivo, que indique el número de vehículos exactos circulando, se basa en la variación respecto a lo que es normal. Debido a los componentes periódicos (Sección 5.11.1) de los flujo de tráfico (Sección 3.2.2) se realiza una correlación entre la influencia del día de la semana y la hora del día con el flujo de tráfico capturado por ambos sistemas. La Figura 6.42 presenta la variación e influencia del día de la semana y hora del día en el número de vehículos detectados por ambos sistemas.

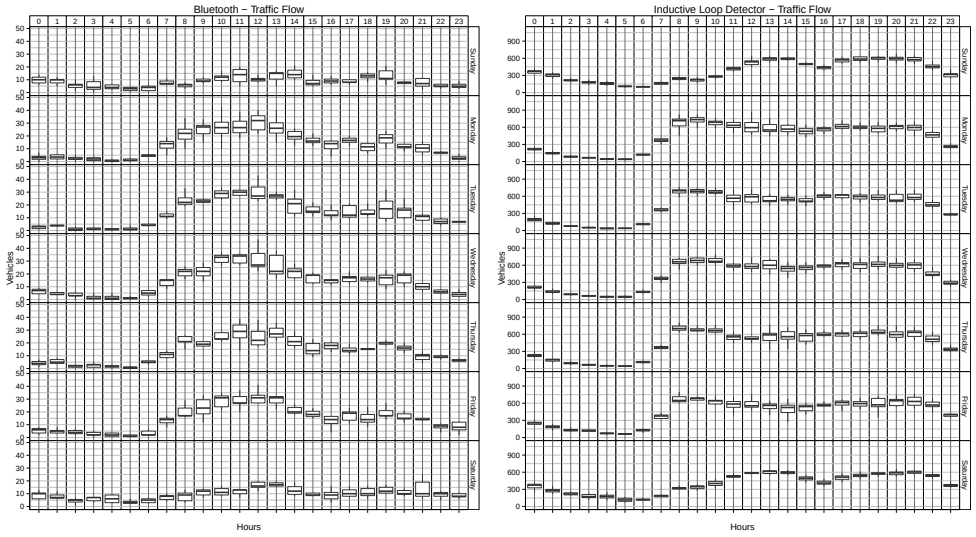


Figura 6.42
Variación del flujo de tráfico del sistema propuesto y los datos del ayuntamiento agrupados por día de la semana y hora.

Si bien la similitud entre ambas gráficas es evidente, se necesitan mayor evidencias para poder indicar que ambos sistemas arrojan resultados similares.

Una regresión lineal simple indica que ambas magnitudes están correlacionadas con factor $R^2 = 0.6389$, lo cual no resulta muy significativo, aunque arroja indicios de que puede existir relación.

Para estudiar la relación entre ambas magnitudes influidas por los factores periódicos, se emplea el Test de Granger [106]. El test de Granger plantea una hipótesis estadística para determinar si una serie temporal X puede predecir los valores de otra serie temporal Y, lo cual implica que existe una causalidad de Granger entre ambas series[25].

[106] Testing for causality: A personal viewpoint

Los resultados del test obtienen un p-value de 0.0002355 que es mejor que el umbral de confianza. Esto arroja una gran evidencia estadística de que la variación de los dispositivos Bluetooth detectados en una zona permite determinar la variación del tráfico real circulando.

Es necesario notar, que este test está preparado para no verse influenciado por la periocidad de la serie, sino que trabaja sobre ella con la magnitud. Aunque en las figuras se han agrupado las series para una mejor representación, este tratamiento no ha sido realizado durante los test, por lo que no se puede alegar a la periodicidad de las series la causalidad de Granger.

Análisis de la congestión del tráfico urbano

Estudio 6.2.5: Análisis del flujo de tráfico

Para determinar la congestión de la vía mediante el empleo del aforador, se emplea el tiempo total que la espira magnética ha estado activa dividido por el número de vehículos detectados. De esta forma, la métrica aproxima el tiempo que ha estado detenido en promedio los vehículos sobre la espira. Sin embargo esta métrica resulta poco precisa, debido a que se basa en velocidad puntuales no a la velocidad de crucero. Debido a que las vías controla el tráfico por medio de semáforos y la espira se activa únicamente con vehículos encima de ellas, los vehículos pueden estar detenidos (por una congestión de la vía) pero que ninguno se encuentre exactamente encima debido a la distancia entre los vehículos.

En el sistema de monitorización propuesto, la saturación de la vía viene determinado por el tiempo requerido para circular de forma directa entre nodos del sistema (Secciones 5.1.6 y 5.10.2.5). Conociendo la distancia entre nodos, es posible convertir este tiempo en velocidades de crucero (Sección 3.2.4).

Al igual que se ha realizado en el anterior estudio, se calcula la correlación de ambas medidas con los factores periódicos de la serie influidos por el día de la semana y la hora del día. La Figura 6.43 recoge el tiempo requerido

25 ↑En otras palabras, el test permite determinar si los factores que causan variabilidad en la misma periodicidad de una serie se reflejan en otra serie en los mismos momentos periódicos en otra serie distinta. Granger ganó el premio Nobel de Economía con dicho test, aplicándolo como ejemplo al precio del Barril del petróleo.

para recorrer la distancia entre ambos nodos y el tiempo de activación de la espira.

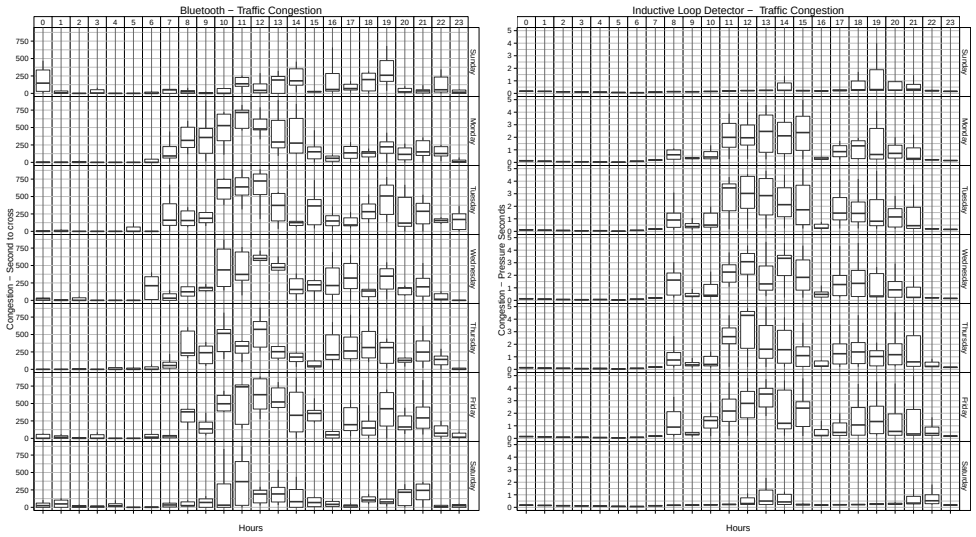


Figura 6.43
Variación de la congestión de tráfico del sistema propuesto y los datos del ayuntamiento agrupados por día de la semana y hora.

Un método simple como una regresión lineal entre ambas magnitudes obtiene un valor $R^2 = 0.4662$, lo cual no resulta significativo. El test de Granger, en cambio, consigue un p-value de 0.5844×10^{-11}, lo cual arroja una evidencia estadística de la causalidad de Granger entre ambas series.

Conclusiones

Tanto el flujo de tráfico real como los indicadores de congestión empleados resultan muy correlacionados con el número de dispositivos Bluetooth detectados evidenciado por una causalidad de Granger.

En el caso del flujo de tráfico, las variaciones que afectan al tráfico de vehículos son afectadas en igual medida pero distinta magnitud en el número de dispositivos bluetooth detectados. En otras palabras, aquello que hace que haya menos vehículos en carretera influye de manera proporcional al número de dispositivos bluetooth detectables.

En el caso de los indicadores de congestión, los resultados fueron análogos al flujo. Con la salvedad de que la métrica obtenida resultaba más útil y atractiva a las autoridades, debido a ser más fácilmente interpretable que el tiempo de activación dividido por vehículo de la espira magnética, al ofrecer el tiempo real requerido para el desplazamiento de los vehículos circulando.

Es por ello que se concluye, que según las pruebas y experimentos realizados, no existen indicios para no poder inferir que la captación de comunicaciones Bluetooth no resulta un indicativo al menos igual de bueno que una espira magnética, igualando los resultados de esta aunque con distinta magnitud.

6.2.3 *Análisis de la predilección del giro en tráfico urbano*

Ampliando la red de sensores, es posible estudiar la predilección de giros de una calle concreta. Se dispone de los sensores desplegados en la Figura 6.44 en la ciudad de Granada.

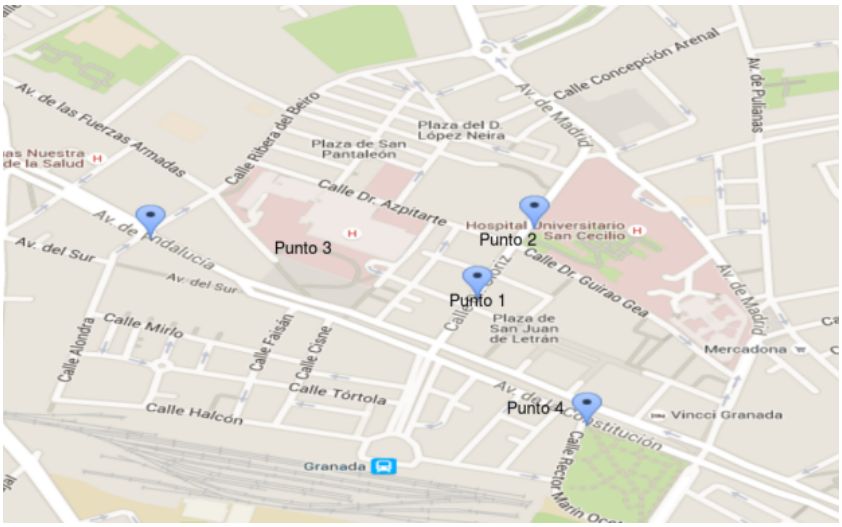


Figura 6.44
Localización de los 4 nodos en el tramo a estudiar la predilección del giro

Se pretenden estudiar la predilección de giro de los conductores entre los puntos $2 \rightarrow 1$, y determinar si algún factor periódico influye en que realicen el recorrido $2 \rightarrow 1 \rightarrow 3$ o que realicen el giro contrario, es decir, $2 \rightarrow 1 \rightarrow 4$.

En estos estudios se hará uso de las matrices origen destino presentadas en la Sección 5.11.2 y sus normalizaciones.

Se estudia durante un periodo de tres meses los datos obtenidos por los 4 nodos. Se presentan tanto los datos relativos a los dispositivos Bluetooth y los dispositivos WiFi. Debido a que se considera que los vehículos hacen uso de comunicaciones Bluetooth, se prevé que el comportamiento de estos dispositivos sea más restrictivo que en el caso del WiFi, que se aplica a los dispositivos portados por las personas, que disponen de mayor facilidad de movimiento.

Análisis de la predilección del giro en tráfico urbano

Estudio 6.2.6: Predilección de giro - Tráfico producido

Con los datos obtenidos se estudia la predilección del giro del recorrido $2 \rightarrow 1$. Para ello se presenta la matriz origen destino representada en al Figura 6.45 normalizada para mostrar el tráfico producido por cada nodo.

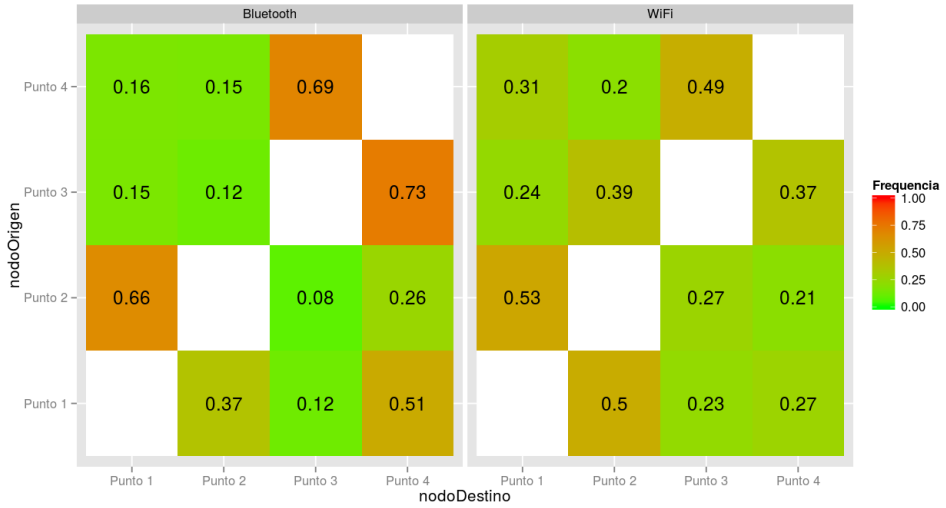


Figura 6.45
Matriz de origen destino normalizada para mostrar el tráfico producido.

El 66 % del tráfico producido en el puno 2 termina en el punto 1. Solo un 8 % llega directamente al punto 3 sin ser detectado anteriormente por ninguno de los otros nodos. Un 26 % llega directamente al punto 4.

Un 37 % del tráfico producido en el punto 1, vuelve al punto 2[26]. Un 51 % gira para ser detectado posteriormente por el punto 4. Tan solo un 12 % gira para ser detectado posteriormente en el punto 3.

Existe una clara tendencia en el punto 2 *de girar hacia la izquierda* para dirigirse hacia el punto 4. De forma de que la mitad de los dispositivos Bluetooth detectados presentan este comportamiento.

En el caso de WiFi, no existen tendencias tan marcadas. Además, de que se impone más la cercanía entre los nodos, ya que los dispositivos WiFi se considera que son portados por las personas y estas tienen libertad de movimiento.

Sin embargo, estos factores son absolutos. Para estudiar si existe alguna influencia periódico, se estudian las matrices origen destino de los distintos días de la semana y horas del día.

26 ↑Lo cual resulta posible por el parking existente en la Plaza de San Juan de Letrán que dispone de un paso subterráneo.

Análisis de la predilección del giro en tráfico urbano

Estudio 6.2.7: Influencia del día de la semana en el Tráfico producido

Se presentan las matrices origen destino de los cuatro nodos de cada uno de los días de la semana, con el fin de estudiar si el comportamiento del tráfico es influido por este factor.

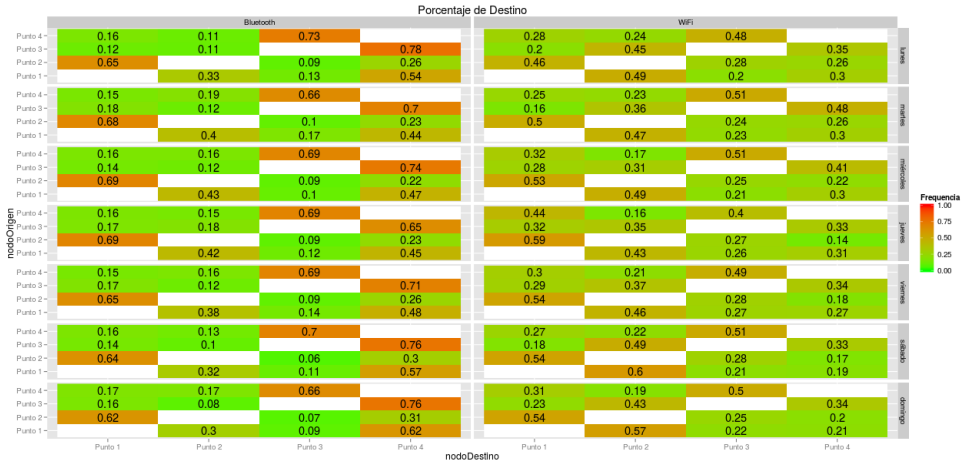


Figura 6.46
Matriz de origen destino normalizada para mostrar el tráfico producido por día de la semana.

La intepretación de las matrices puede resultar tedioso. Es por ello que se presenta la Figura 6.47 donde se representan gráficamente los valores de las matrices.

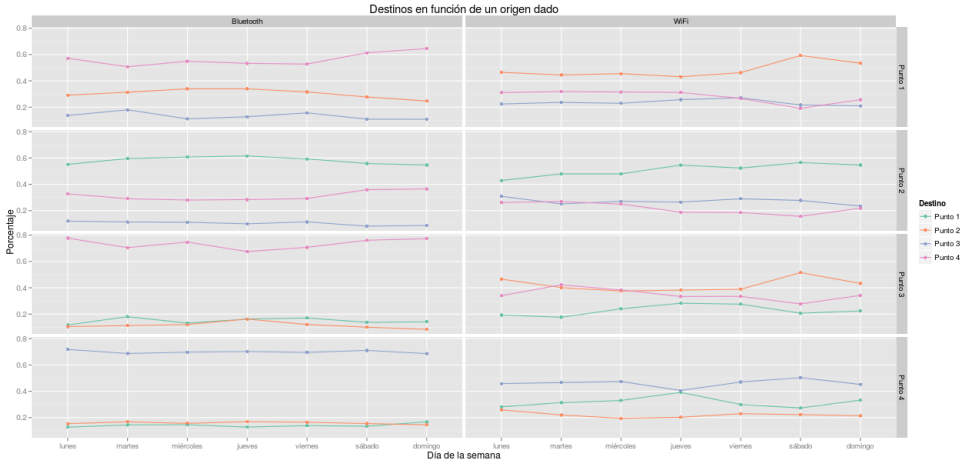


Figura 6.47
Evolución del tráfico producido por día de la semana.

Se observa que en caso de Bluetooth no existen cambios en la proporción del tráfico producido por ninguno de los nodos. En vista de estos resultados, se puede inferir que no se ha encontrado evidencia alguna de que la predilección de giro en la calle estudiada se vea influenciada por el día de la semana.

Análisis de la predilección del giro en tráfico urbano

Estudio 6.2.8: Influencia de la hora en el Tráfico producido

Se estudia también la influencia de la hora del día en la predilección de giro. Se obvia presentar las 24 matrices, presentando la Figura 6.48.

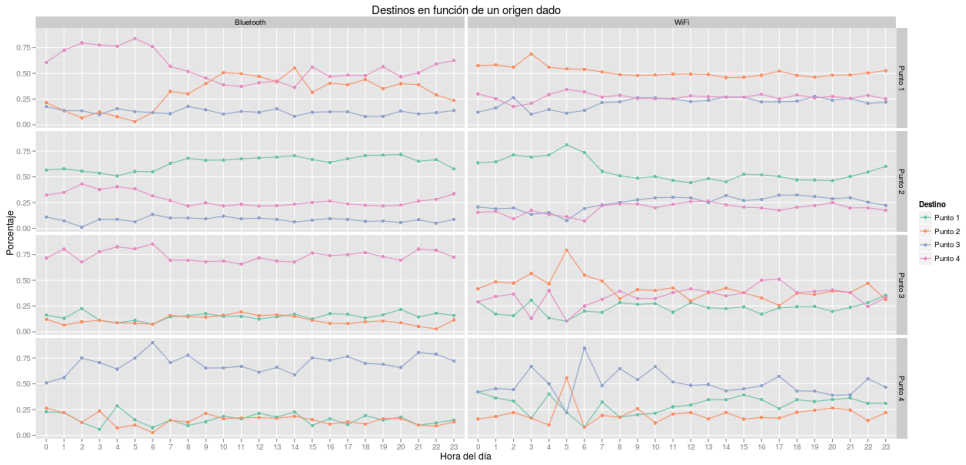


Figura 6.48
Evolución del tráfico producido por día de la semana.

En este caso para el punto 1 que es el punto de estudio, si se observan variaciones significativas respecto a la hora del día. Desde las 9:00 a las 15:00, la mayor parte del tráfico producido opta por girar hacia el punto 3, pero el porcentaje de giro hacia el punto 4 es similar. A partir de las 15:00 se cambian las tornas, pasando a ser el giro hacia el punto 4 el predominante. Esta diferencia se acrecienta con el paso de la hora, siendo muy significativa durante la madrugada[27].

Como era de esperar, en el caso del WiFi existen menos tendencias marcadas, debido a como se ha indicado anteriormente, la mayor libertad de movimiento del que disponen estos dispositivos.

Conclusiones

Con el sistema propuesto es posible realizar estudios sobre predilección de rutas y giros que tengan presente los factores periódicos, como el día de la semana y la hora del día. Los sistemas actuales, se basan únicamente en la propia calzada, pero con el sistema propuesto es posible realizar este tipo de análisis para varios puntos de una misma ciudad, por ejemplo distintos barrios. Dado que las administraciones carecían de la información que le estamos ofreciendo, no se puede comparar. Sin embargo, consideran que los resultados obtenidos obedecen al comportamiento fruto de la observación directa del tráfico, aunque de la influencia de la hora del día nunca habían tenido evidencias tan claras.

27 ↑Donde también el numero de dispositivos detectados es menor.

6.2.4 *Análisis de tráfico Interurbano en vías de alta capacidad I*

Gracias a la colaboración con la Dirección General de Tráfico (DGT), ha sido posible estudiar la viabilidad del sistema para el análisis del tráfico interurbano. La DGT cuenta en sus vías principales de sistemas de monitorización de tráfico basadas tanto en imágenes de vídeo (Sección 3.4.3) como en espiras magnéticas 3.4.2. De estas últimas, nos han cedido los datos para poder realizar una comparación con los datos obtenidos por el sistema de monitorización propuesto.

La DGT permitó la instalación de 6 nodos de monitorización emplazados en vías de alta capacidad a lo largo de la geografía andaluza. La Figura 6.49 recoge los nodos y sus identificadores.

Figura 6.49
Emplazamiento de los nodos para el estudio del trafico interurbano a lo largo de la geografía andaluza.

Los nodos de monitorización son emplazados en las proximidades de las carreteras, dentro de los edificios de mantenimiento de la DGT[28]. En la Figura 6.50 se recogen las distancias de los nodos a las carreteras donde han sido emplazados. En los casos de que un nodo tuviese acceso a más de una carretera distinta, se emplazado la antena de captación hacia la carretera deseada y se empleado aislante[29] para focalizar la captación.

Estos nodos era mandatario que fuesen instalados en estas zonas concretas ya que eran zonas donde existían aforadores a escasos metros del emplazamiento, con el objetivo de comparar ambas fuentes de datos.

28 ↑Las implicaciones de este hecho, han sido estudiadas en la Sección 6.1.4. Si bien no resultaban las más optimas y han sido mejoradas a lo largo de la implantación del sistema de monitorización.

29 ↑Capas y capas de papel aluminio.

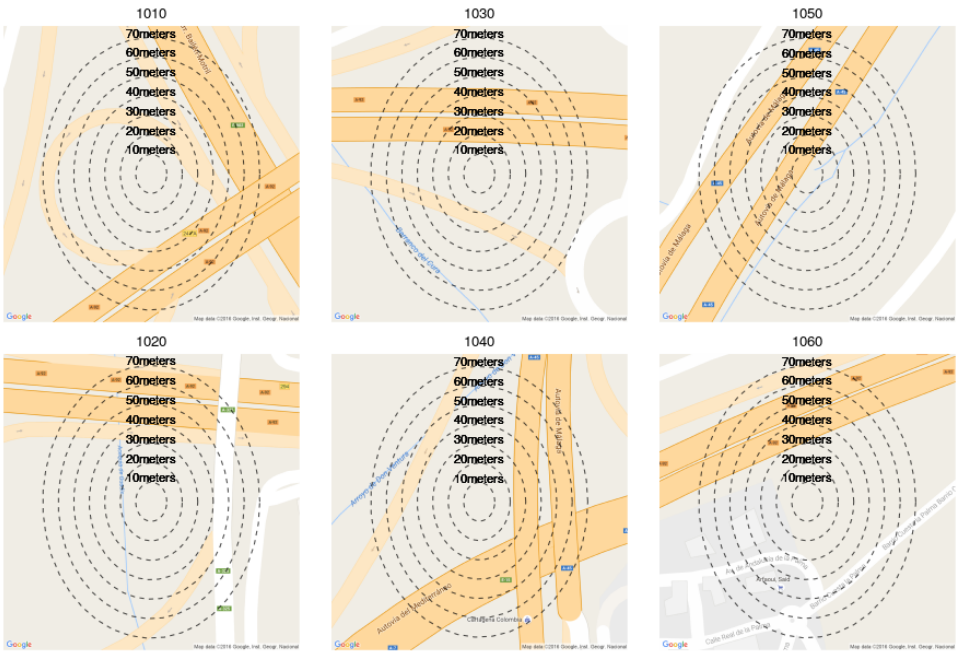


Figura 6.50
Distancias de los nodos a las carreteras en metros.

Análisis de tráfico Interurbano en vías de alta capacidad I

Estudio 6.2.9: Comparativa entre sistemas de monitorización

Si bien el sistema de monitorización presentado no es exhaustivo, puede ser deseable obtener un factor de conversión entre dispositivos detectados y vehículos reales. De esta forma, se puede determinar el porcentaje de vehículos circulando que emplean algún dispositivo Bluetooth.

Se proponen 5 criterios para el cálculo de este ratio:

- *Ratio absoluto*: ratio entre el número de vehículos totales detectados por los aforadores dividido entre el número de dispositivos detectados detectados por el sistema de monitorización propuesto.
- *Ratio por media*: ratio entra la media del número de vehículos por hora detectados entre los aforadores y la media de dispositivos por hora detectado por el sistema de monitorización propuesto.
- *Ratio de mediana*: Debido a que la media no es una métrica muy tolerable a valores extremos, igual métrica que la anterior pero empleando como estadístico la mediana.
- *Ratio por media por hora*: este método calcula un vector de ratio de conversión dependiente de la hora del día, empleando como estadístico la media.
- *Ratio por mediana por hora*: Igual que el método anterior, pero empleando la mediana como estadístico.

En el nodo 1010 donde se realizaron las mediciones, se determinó que el ratio absoluto de captura era de 17.23 coches por cada dispositivo Bluetooth

detectado. Sin embargo, una de las observaciones realizadas, es que en términos absolutos se detectaban más dispositivos Bluetooth por la noche que durante el día. La DGT nos indicó que podría ser debido a que los conductores que se ven obligados a conducir por la noche suelen ser profesionales de la conducción y que por tanto, sus vehículos están mejor acondicionados. Por ello se realizó un ratio de conversión basado en la hora del día, que arrojó mejores resultados. En la Tabla 6.6 se presentan las métricas de error de la magnitud de los aforadores con la obtenida por el sistema de monitorización aplicado el ratio de conversión.

Tabla 6.6
Métricas de error de la comparativa.

	RATIO	MAE	MAPE	MSE	RMSE
Ratio absoluto	17.230	60.373	36.265	**7371.579**	**85.857**
Ratio por media	20.792	72.323	43.443	11216.070	105.905
Ratio de mediana	18.2	62.454	37.515	8060.136	89.778
Ratio por media por hora	–Dependiente–	66.791	40.120	9890.978	99.453
Ratio por mediana por hora	–Dependiente–	**58.326**	**35.035**	7617.613	87.278

El sistema no resulta perfecto en la conversión debido sobre todo al comportamiento del sistema durante las horas de la noche y los valores extremos. Esto se puede evidenciar de forma más clara en la Figura 6.51 donde se presentan los valores obtenidos por el aforador de la DGT y el valor obtenido por el sistema aplicado el ratio de conversión de mediana por hora.

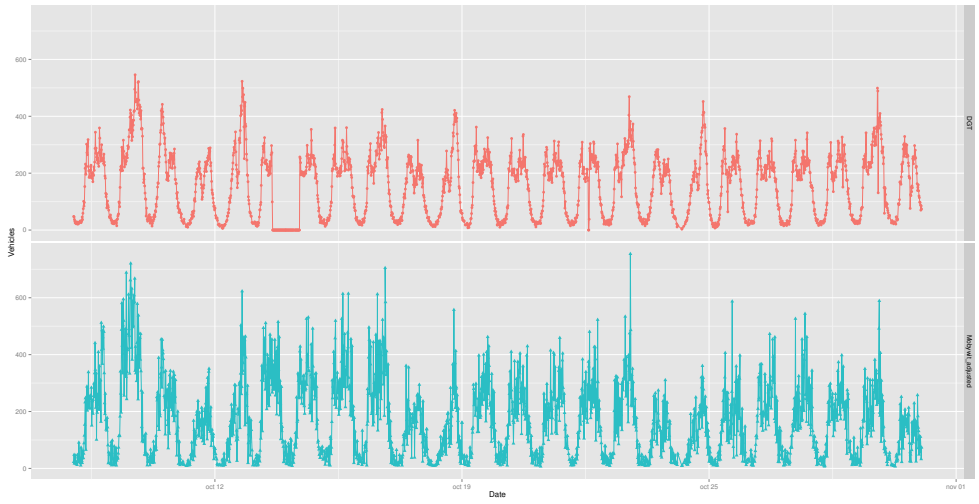


Figura 6.51
Comparativa entre series del aforador y el sistema propuesto. Arriba (en rojo) los valores obtenidos por la DGT. Abajo (en azul) los valores del sistema de monitorización propuesto aplicado el factor de conversión por mediana por hora.
Nota: Debido a un error del sistema de la DGT no había datos disponibles de su sistema durante el 14 de Octubre.

En el momento de realización de estos estos estudios, el sistema no resultaba preciso para la cuantificación del volumen de tráfico.

Análisis de tráfico Interurbano en vías de alta capacidad I

Estudio 6.2.10: Análisis de las magnitudes del tráfico

Debido a que el sistema es capaz de identificar de forma unívoca a cada dispositivo, es posible detectar por varios nodos a un mismo dispositivo. De esta forma, es posible elaborar matrices origen destino (Sección 5.11.2) que muestren la interacción de los dispositivos detectados entre los distintos nodos de la red.

Estudiar la variabilidad de estas matrices a lo largo del tiempo, permite determinar si el comportamiento de los dispositivos es constante e invariante, o al contrario, se ve influenciado por factores periódicos.

En la Figura 6.52 se presentan las matrices de origen destino de los nodos influidas por el día de la semana y la hora del día en términos de frecuencia. La hora del día se ha agrupado siguiendo bloques de 4 horas.

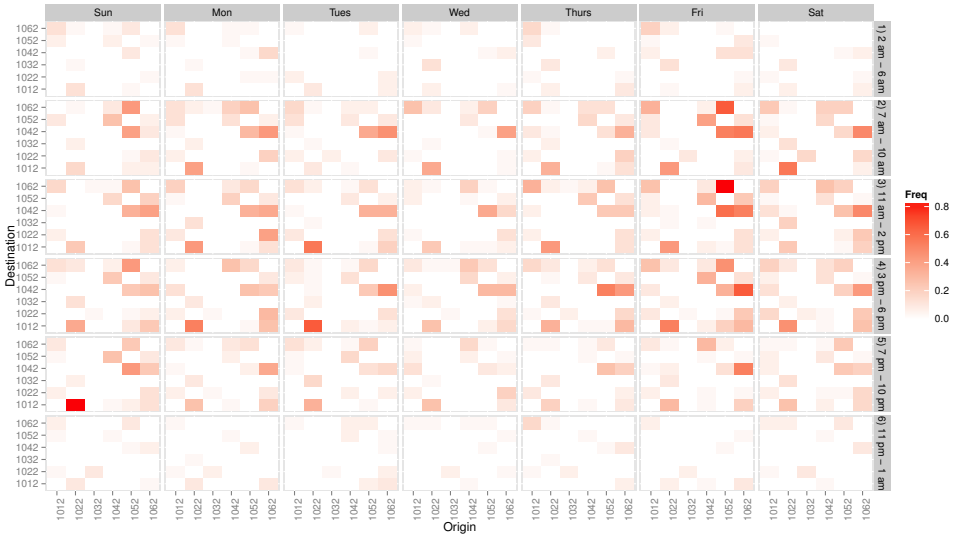


Figura 6.52
Frecuencia de tráfico entre nodos

Conclusiones

Si bien en el estudio de la posición de los nodos y por tanto sus antenas podría ser mucho más extendido, esta sección viene unicamente a demostrar que las comunicaciones inalámbricas de los dispositivos inteligentes existen allá donde estos se mueven, y que su no detección en la mayoría de las ocasiones abordadas en la aplicación del sistema de monitorización de esta tesis, son debido a malos emplazamientos del nodo, no debido a la invalidez del sistema.

6.2.5 *Análisis de tráfico Interurbano en vías de alta capacidad II*

La red de sensores presentada en la Sección 6.2.4 es mejorada acercando los nodos sensores a la carretera[30], disponiendo de una mejor posición y emplazamiento, como se ha presentado en la Sección 5.4.7.

Este acercamiento se representa en la Figura 6.53 donde se puede comparar las distancias de los nodos 1010 y 1020 del anterior proyecto con los nuevos nodos emplazados

(a) 1010 (b) 1020 (c) 1110

(d) 1120 (e) 1130 (f) 1140

(g) 1150 (h) 1160 (i) 1170

Figura 6.53
Posición de los nodos de monitorización acercados a la carretera.

Debido a que no era objetivo del proyecto, los nodos no se colocaron cerca de aforadores de la DGT con los que poder realizar comparativas exhaustivas.

30 ↑Gracias al empleo de comunicaciones móviles 3G para el envío de la información, permitiendo dejar de emplear la red cableada de la DGT.

Análisis de tráfico Interurbano en vías de alta capacidad II

Estudio 6.2.11: Magnitud del tráfico y comparación con datos oficiales

Se presentan en la Figura 6.54 las magnitudes de cada nodo expresado en dispositivos detectados a la hora.

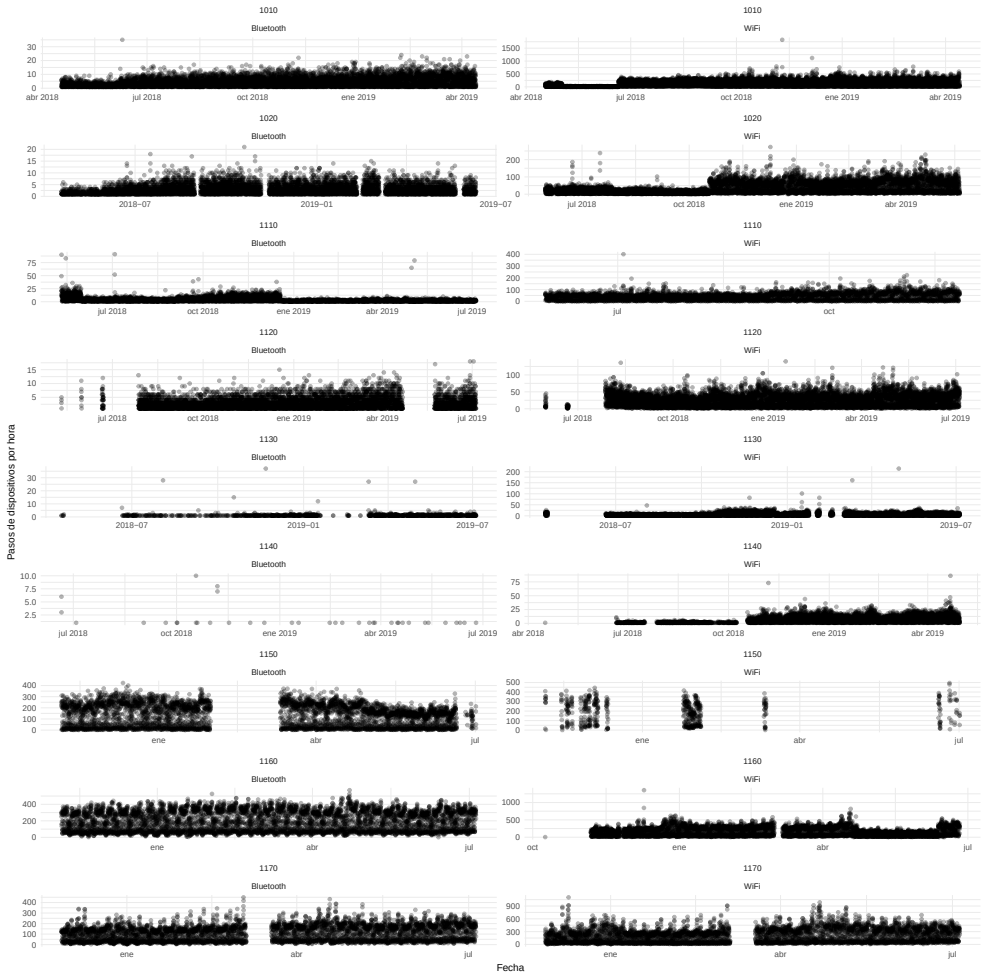


Figura 6.54
Magnitud de dispositivos/hora de los nodos sensores

Debido al pésimo emplazamiento, los nodos 1120, 1130 y 1140 no obtienen datos interesantes de analizar. Los nodos 1150, 1160 y 1170 si resultan interesantes de analizar, moviéndose en rangos de valores similares, alcanzando un valor entorno a los 300 y 400 dispositivos a la hora en el caso de Bluetooth.

Si bien no es posible una comparación directa[31], se puede visualizar los datos que recoge la DGT sobre tramos en la misma carretera, como se presenta en la Figura 6.55.

31 ↑Los datos no han sido solicitados oficialmente a la DGT y a los procedimientos de extracción automática empleados con anterioridad han dejado de funcionar debido a cambios en la web de donde se obtenían.

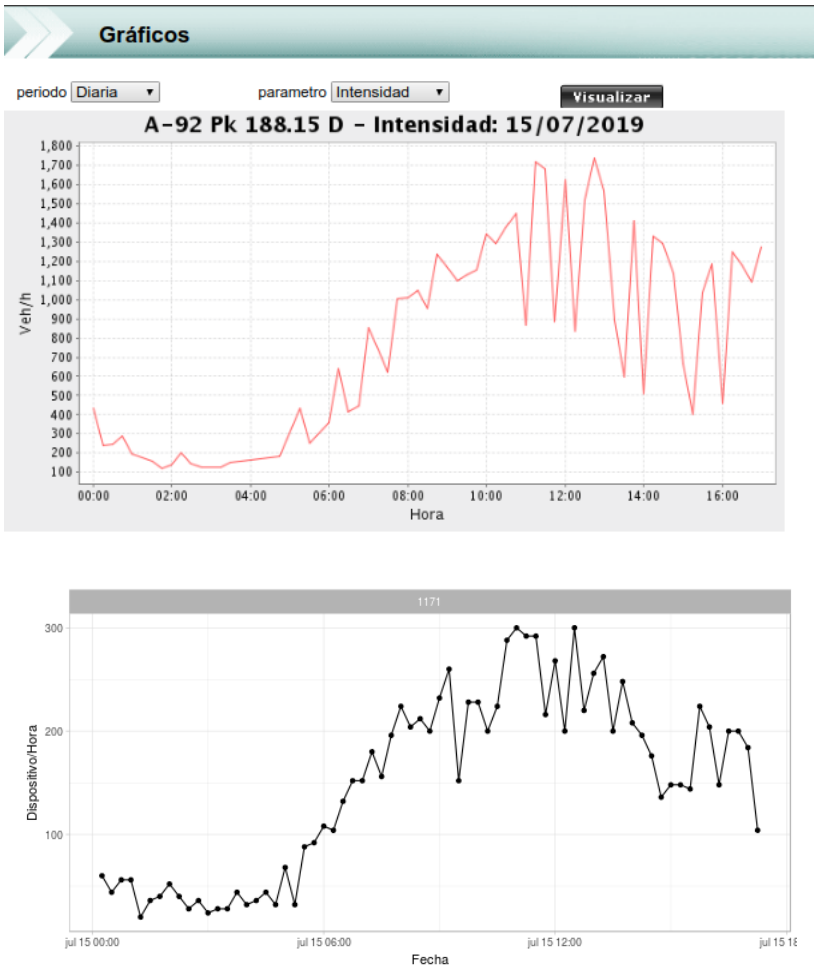


Figura 6.55
Datos oficiales de la misma carretera donde se encuentra ubicado el nodo 1170, pero en distinto punto kilométrico.
Es necesario notar que no existe una correspondencia directa ya que ambos punto se encuentran separados por unos 11km con numerosos puntos de salida e incorporación de tráfico..

El número de dispositivos Bluetooth detectados respecto al número de vehículos reales se presenta en un ratio aproximado de 1 de cada 6, lo que mejora de forma significativa los datos del Estudio 6.2.4, separados dos por dos años de tiempo[32].

32 ↑Aunque parte de la mejora ha sido el acercamiento a carretera

Análisis de tráfico Interurbano en vías de alta capacidad II

Estudio 6.2.12: Tiempos de viaje

Los nodos 1160 y 1170 se encuentran colocados en la $A-92$, en los puntos kilométricos 240 y 177 respectivamente, por lo que se encuentran separados por 63 kilómetro de vía.

Resulta factible obtener tiempos de desplazamiento entre nodos y estudiar la variabilidad de la velocidad entre dichos puntos.

Se presenta en la Figura 6.56 la velocidad de crucero de los dispositivos Bluetooth para haber sido detectados por ambos nodos en los tiempos en los que han sido detectados.

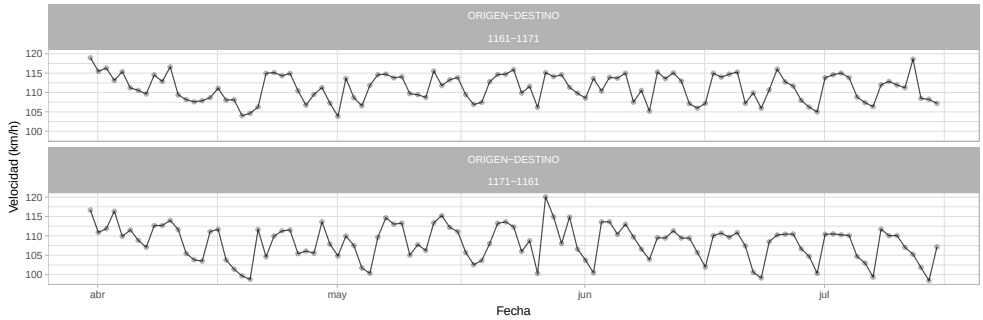


Figura 6.56
Velocidad de crucero promedia de los dispositivos Bluetooth detectados

Se puede contrastar como los fines de semana la velocidad media de los vehículos se reduce de forma significativa, lo cual se presenta en la Tabla 6.7.

Tabla 6.7
Velocidad de crucero promedia de los dispositivos Bluetooth detectados por día de la semana.

Día	lunes	martes	miércoles	jueves	viernes	sábado	domingo
km/h	111.8518	113.0502	112.8709	111.6471	107.5037	106.5568	106.1671

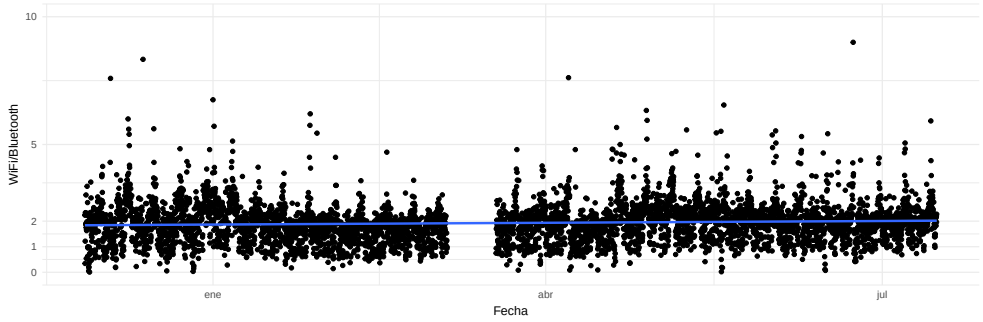
Esta velocidad está derivada del tiempo de viaje, no de la velocidad instantánea, por lo que provee de un MOE muy efectivo y estudiado para determinar el impacto real de los percances en el tráfico y la eficacia de los planes de emprendidos.

Análisis de tráfico Interurbano en vías de alta capacidad II

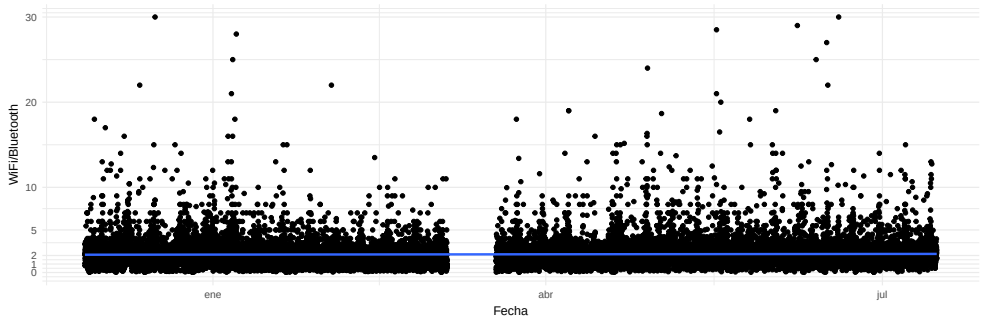
Estudio 6.2.13: Aproximación al número de ocupante por vehículo

Una de las magnitudes del tráfico de la cual no dispone la DGT ninguna métrica es la que aproxima el número de ocupantes por vehículo.

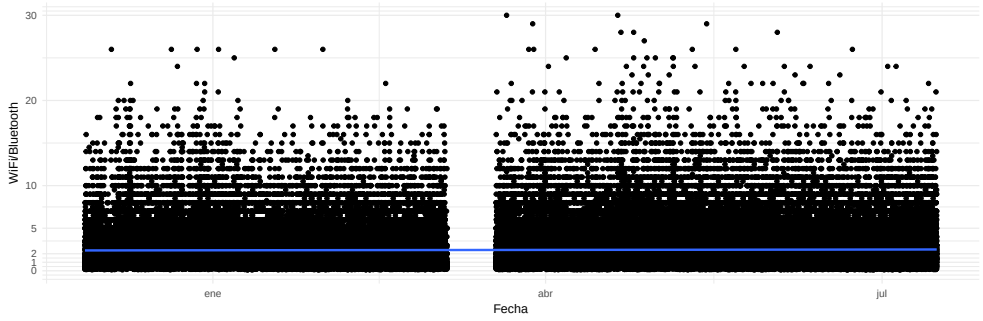
Supuesto un escenario ideal en el que cada vehículo cuenta con una conexión Bluetooth y un conductor que porta un dispositivo inteligente, siendo las comunicaciones de ambos elementos detectables por el sistema, el sistema podría ser empleado para determinar el número de dispositivos inteligentes por vehículo detectado, y de tal manera aproximar el número de ocupantes por vehículo.



(a) 60 minutos de muestreo



(b) 10 minutos de muestreo



(c) 1 minuto de muestreo

Figura 6.57
Ratio del número de dispositivos WiFi por dispositivo Bluetooth en distintos intervalos.

Si bien la métrica no es precisa, arroja valores coherentes en ventanas de tiempo grandes. En la Figura 6.57 se presenta el número de dispositivos

WiFi por dispositivo Bluetooth detectado a la hora. Si bien existen escenarios en los que el ratio es inferior a 1, los valores parecen acotados entre 1 y 10 dispositivos WiFi por cada dispositivo Bluetooth detectado, con un valor medio aproximado en torno a 2, lo que implicaría en un supuesto ideal que por cada vehículo son detectados dos ocupantes.

Si se reduce la ventana de muestreo, el aumento se hace más significativo y se ve más influenciado por valores extremos como se presenta en la Figura 6.57, donde existen casos de madrugada donde se llegan a detectar hasta 30 dispositivos WiFi por dispositivo Bluetooth detectado[33].

Esta métrica es explotable en los distintos factores periódicos, previamente estudiados en otros escenarios, como el día de las semana o la hora del día. Se recoge en la Figura 6.58 los rangos de valores del ratio en función del día de la semana y hora del día.

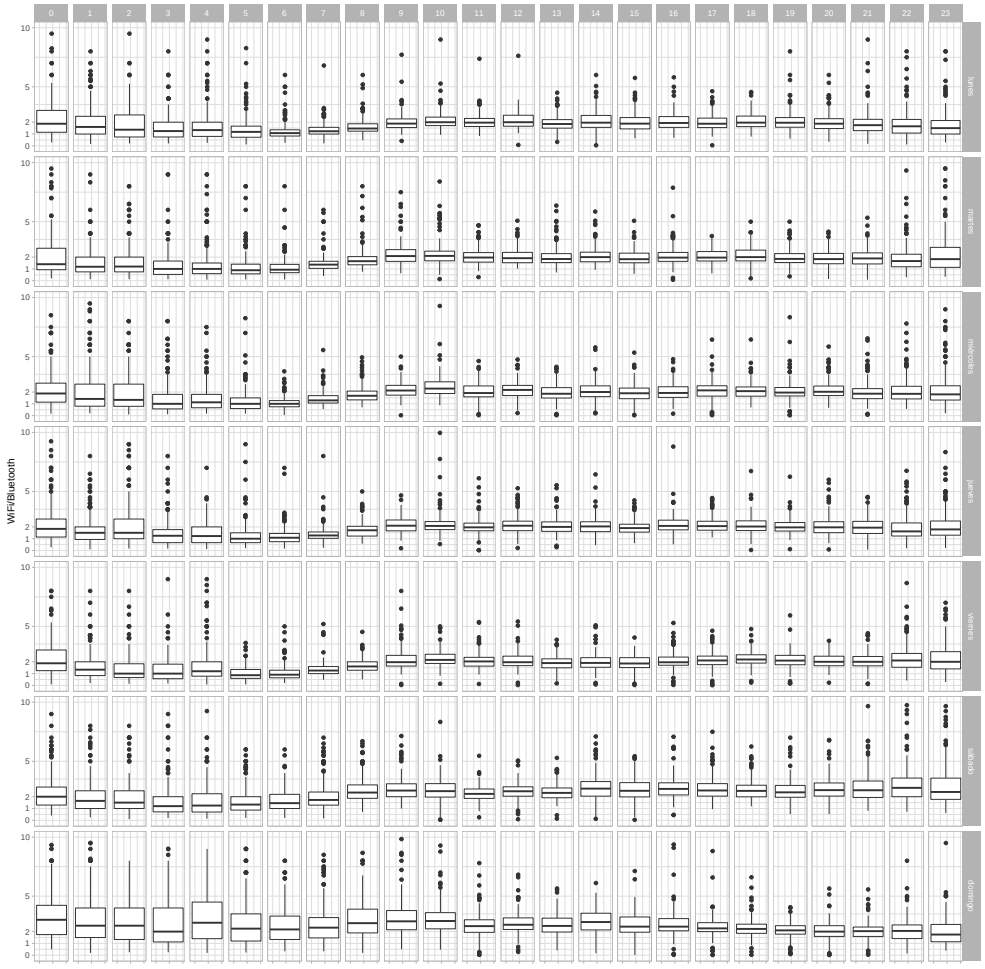
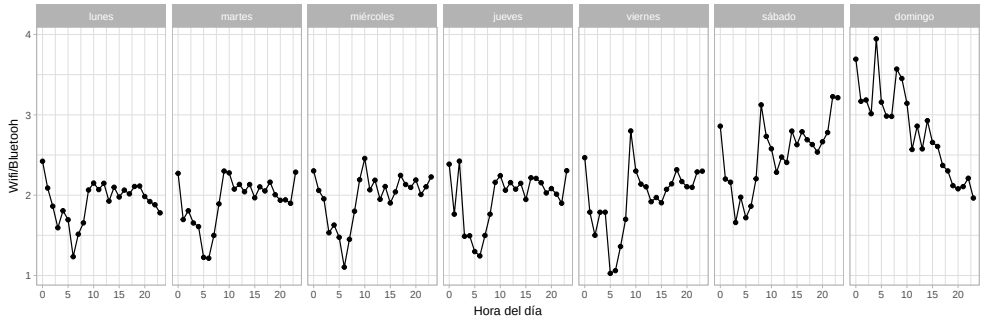


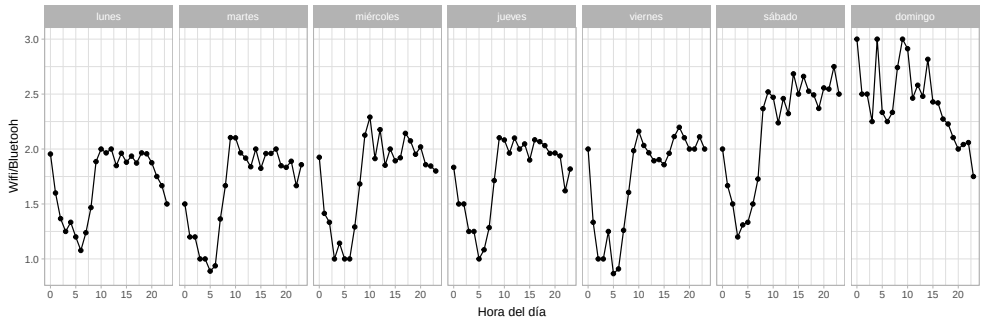
Figura 6.58
Influencia de los factores periódicos en el ratio de dispositivos WiFi por dispositivo Bluetooth.

33 ↑No se descarta que los datos sean reales y se trate de autobuses que transportan a un alto número de personas y estén influyendo en el número de dispositivos detectados por vehículo.

Para facilitar la interpretación de la Figura anterior, se presenta la Figura 6.59 donde se han representado los valores medios y promedios del ratio en función del día de la semana y la hora.



(a) Valor medio



(b) Valor promedio

Figura 6.59
Valores medio y promedios del ratio influido por factores periódicos.

Se aprecian similitudes de valores entre los días de la semana, mostrando un ratio cercano a 1 durante las horas de la noche, incrementándose de forma paulatina a lo largo de la mañana hasta aproximarse hasta un ratio de 2. Los fines de semana este ratio se sitúa más de medio por encima en los valores promedios y hasta 1.5 puntos en los valores medios.

Conclusiones

El número de dispositivos Bluetooth detectables en las carretas se ha visto incrementado en los nuevos nodos sensores emplazados. Este tema será nuevamente abordado en la Sección 6.2.10.

Debido a la proximidad de los nodos, es posible determinar tiempos de viaje, que doten de información sobre el tiempo requerido para realizar los desplazamientos.

Se presenta una aproximación para la estimación del número de ocupantes por vehículo, asumiendo que la totalidad de los vehículos fuesen detectados por Bluetooth y todos los ocupantes fuesen detectados por WiFi. Posterior-

mente dicha aproximación es estudiada en el entorno actual, con un sistema de monitorización no exhaustivo.

Si bien es aventurado decir que el sistema de monitorización basado en la captación de comunicaciones presentado en esta tesis provee en la actualidad de un mecanismo para inferir el número de ocupantes por vehículo, las variaciones que ofrece el ratio parecen seguir un comportamiento justificable por la compartición de vehículos en distintos momentos del día, como los fines de semana.

Debido a que las organizaciones no disponen de ningún método para el estudio del número de ocupantes por vehículo no resulta viable realizar ningún estudio comparativo.

Sin embargo, no se encuentran evidencias de que el método planteado pueda resultar inviable en el futuro, supuesto que el sistema de monitorización se torne exhaustivo por la renovación de la flota de vehículos y la proliferación de los dispositivos inteligentes hasta abarcar la totalidad de la población.

En la actualidad, con valores de detección aproximados de 1 de cada 6 vehículos, no es posible inferir en la cuantificación de la ocupación. Sin embargo la fluctuación de los valores del ratio presentado se ven fuertemente influidos por los valores periódicos estudiados, tal y como sería esperable en una magnitud relacionada con el número de ocupantes por vehículo.

Sin embargo, es deseable que esta aproximación sea estudiada en el futuro[34] como se presentará en la Sección 7 de Conclusiones.

34 ↑Pero queda fuera del ámbito de esta tesis.

6.2.6 *Movilidad de personas en edificios: Discoteca*

Se emplazan 5 nodos en un centro de ocio nocturno ubicado dentro de un centro comercial. El nodo A y B pertenecen a salas de acceso reservado. El nodo C pertenece a la sala principal. El nodo D pertenece a la entrada del recinto. El nodo E pertenece a la terraza (exterior).

Movilidad de personas en edificios: Discoteca

Estudio 6.2.14: Cuantificación de los visitantes

En la Figura 6.60 se representan el número total de dispositivos por cada nodo sensor.

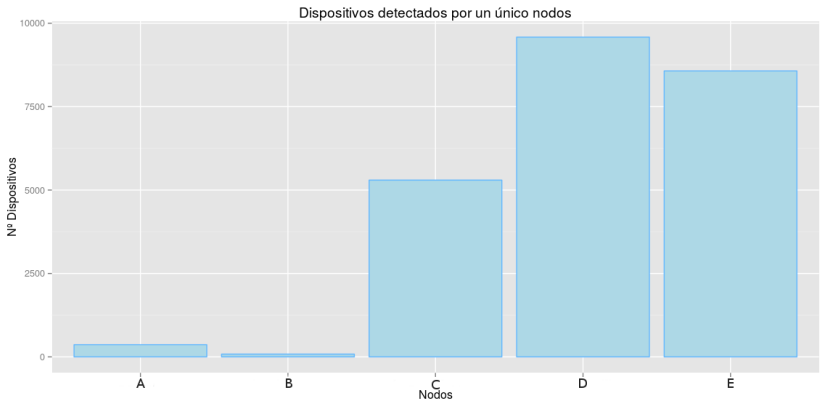


Figura 6.60
Numero de dispositivos WiFi únicos detectados por cada nodo a lo largo de una noche.

Debido a que los nodos sensores se colocan muy cerca de sitios de paso de un gran número de personas al tratarse de un centro comercial, se hace un estudio de reincidencia, que se resume en la Figura 6.61 donde se cuantifican los dispositivos que han sido detectados por número de nodos sensores en los que ha sido detectado.

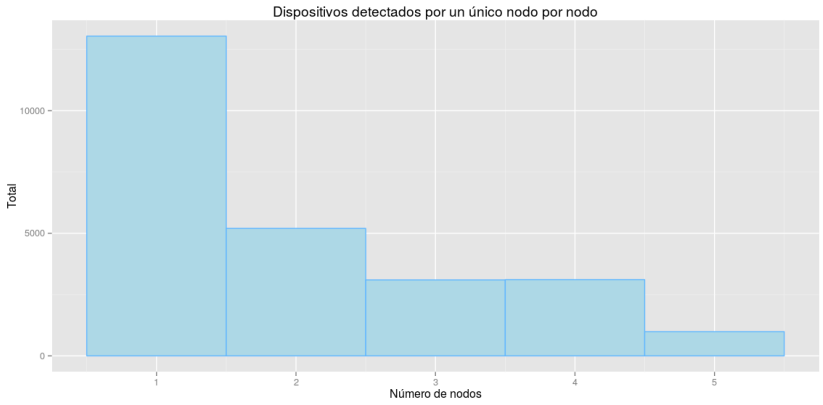


Figura 6.61
Dispositivos reincidentes en varios nodos.

De esta forma, es posible imponer que el número de dispositivos que se estimen en la sala principal (nodo C), hayan tenido que ser detectados con anterioridad en la Entrada del local (nodo D). De esta forma es posible determinar de forma aproximada cuando dispositivos (y por tanto personas) habían entrado en la recinto, formando una Serie Temporal como la mostrada en la Figura 6.62.

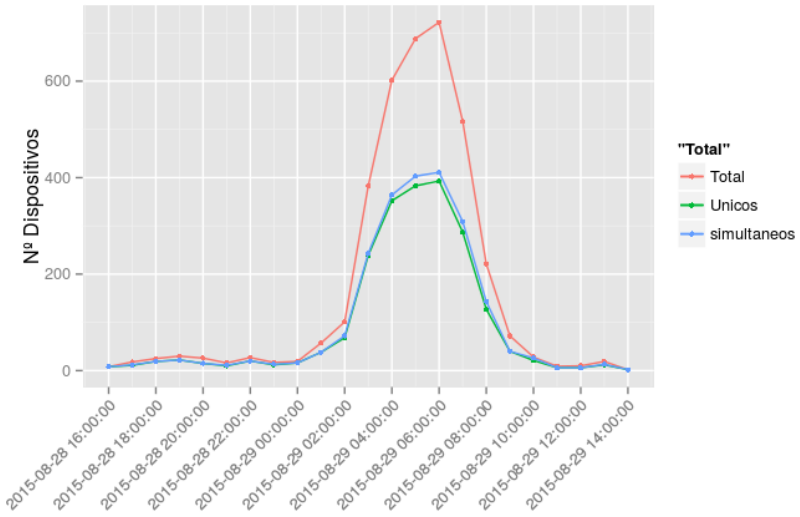


Figura 6.62
Numero de dispositivos estimados en la Sala del nodo C.

Movilidad de personas en edificios: Discoteca

Estudio 6.2.15: Densidades de visitantes - Mapas de calor

El poder emplazar varios nodos en un recinto más o menos acotados, se pueden realizar estudios de la densidad de los dispositivos detectados. Por ejemplo, en la Figura 6.63 se recogen dos momentos concretos del escenario de modificación, variando las salas que están disponibles para el acceso de los visitantes. Si bien en este escenario, el número de nodos es limitado, las posibilidades en escenarios con mayor cantidad de nodos permiten de forma inmediata determinar cuales son las zonas con mayor afluencia de gente.

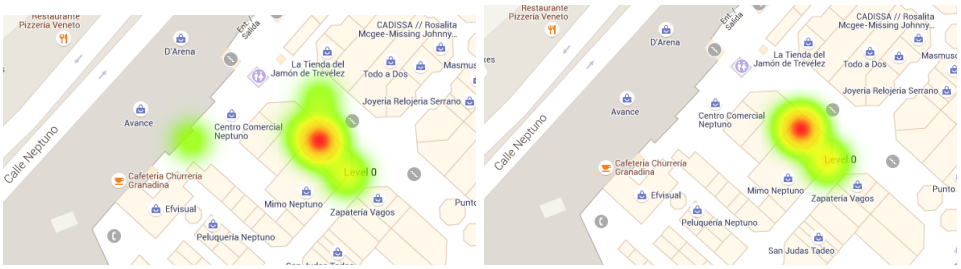


Figura 6.63
Mapas de calor de los dispositivos detectados en cada nodo sensor. En la Figura de la izquierda, se encuentran habilitadas las zonas VIP y el acceso al exterior. En la Figura de la derecha, estas zonas no están habilitadas al público.

Movilidad de personas en edificios: Discoteca

Estudio 6.2.16: Trazabilidad de estancias en distintas salas

Debido a la trazabilidad de los dispositivos (Sección 5.1.6) y al cálculo de dispositivos simultáneos (Sección 5.1.5) es posible reconstruir en que salas ha estado el dispositivo detectado durante su visita. En la Figura 6.64 se presentan el itinerario de dos dispositivos a lo largo de la noche.

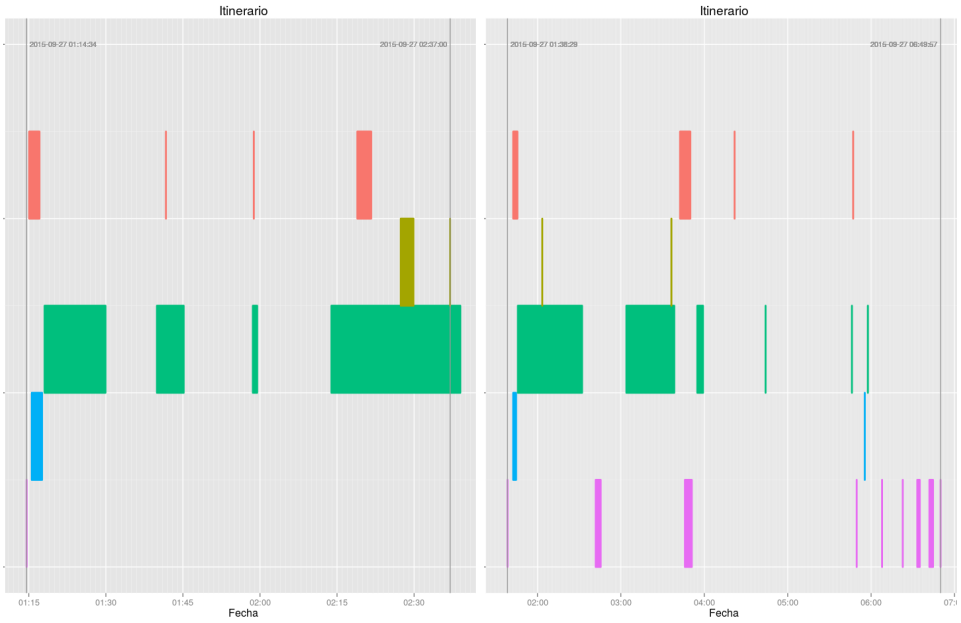


Figura 6.64
Itinerarios de dos dispositivos a lo largo de la noche.

Esto mismo puede ser realizado para todos los dispositivos que hayan sido detectados a lo largo de la noche, determinado cuanto tiempo han permanecido en cada sala así como cuanto tiempo ha durado su estancia en las instalaciones.

En la Figura 6.65 se presentan los itinerarios de estancias de todos los dispositivos detectados en una noche[35] de verano.

Se han descartado los dispositivos que han sido detectados en un único nodo sensor del sistema o que han sido detectados antes de la hora de apertura del negocio al público.

35 ↑Esta noche difiere de la noche presentada en el anteriores estudio, es por ello que las magnitudes son distintas.

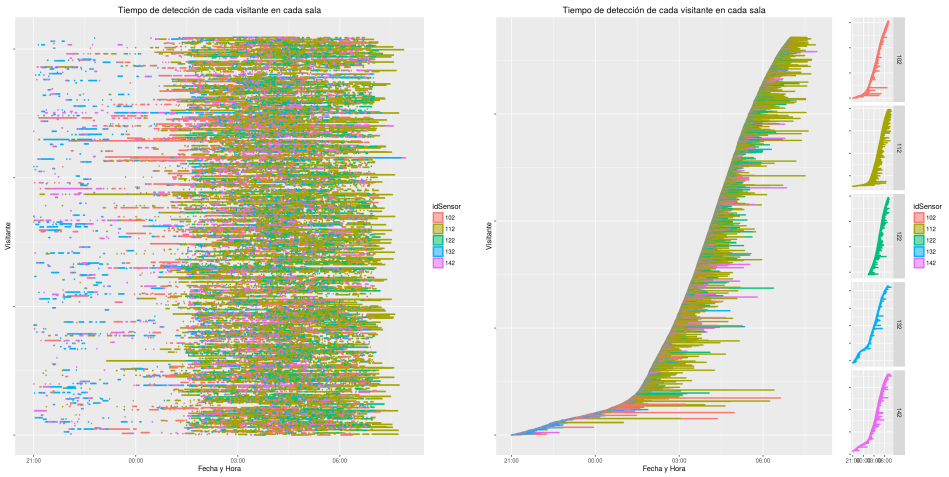


Figura 6.65
Itinerarios de todos dispositivos a lo largo de la noche. Para facilitar la visibilidad, se presenta una ordenación de los dispositivos en función de su hora de entrada y una descomposición por sala donde ha sido detectado dicho dispositivo.

Conociendo la primera y última detección del dispositivo, es posible determinar su tiempo de estancia en las instalaciones, como se presenta en la Figura 6.66.

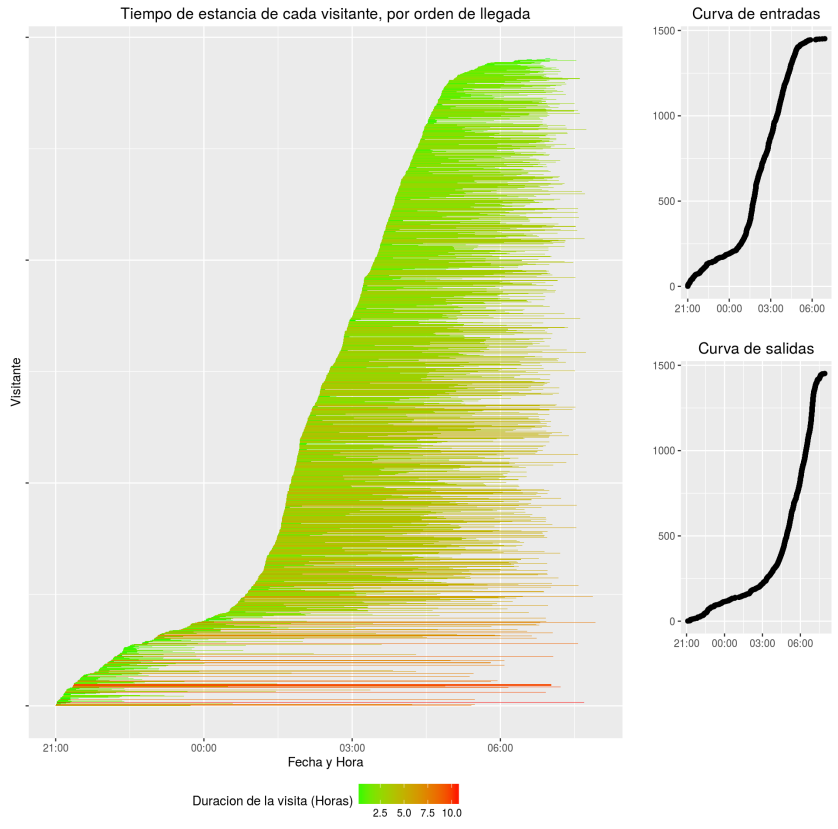


Figura 6.66
Tiempo de estancia de los dispositivos detectados

Con esta información resulta muy sencillo determinar en que momento se ha producido la mayor cantidad de entrada de gente y en que momento se ha empezado a ir la mayor parte de los asistentes. En la Figura **??** se presentan las horas de entrada y salida críticas de dicha noche.

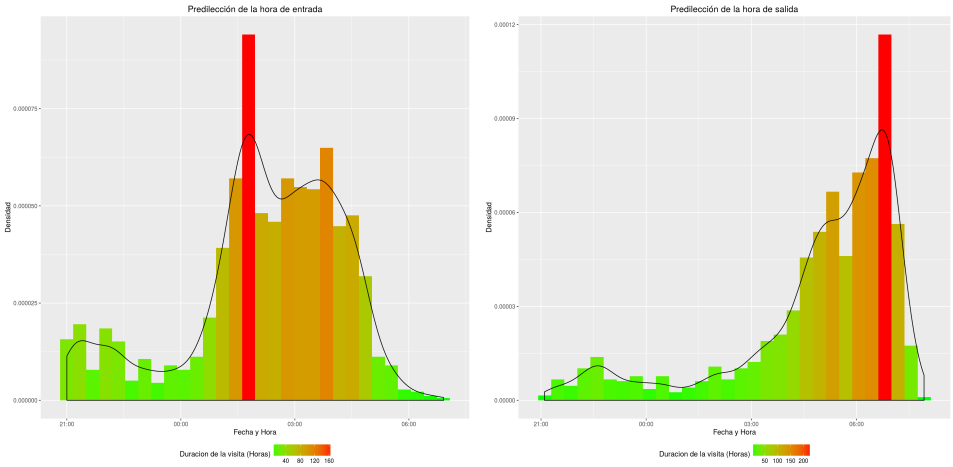


Figura 6.67
Horas de entrada y salida con mayor afluencia de gente

Resulta reseñable como a pesar de que las instalaciones abren a las 21:00, la mayor cantidad de gente entra pasadas las 1:30 de la madrugada. Respecto a la salida, se incrementa pasadas las 3:00 de la madrugada produciéndose el desalojo entre las 6 y 7 de la mañana.

Conclusiones

Si bien los dueños del local no facilitaron información sobre el aforo exacto durante las noches monitorizadas, la interpretación que hicieron del informe entregado sostenía que los resultados obtenidos por el sistema resultaban coherentes con sus apreciaciones.

La información sobre las horas críticas de entrada y salida fue muy bien recibida, porque les daba una evidencia empírica de algo que sospechaban los trabajadores pero no podrían demostrar. Emprender acciones cerca de las 3 de la mañana puede permitir alargar la estancia de los visitantes. Y eso puede ser medido nuevamente con el sistema propuesto y analizarlo para determinar si dichas acciones han tenido algún impacto real.

Todos los datos aquí presentados son muestras de los hábitos de comportamiento de los visitantes. En la Sección 6.2.6 se presentarán los resultados de realizar algoritmos de aprendizaje y extracción de patrones para extraer conocimiento sobre este escenario.

6.2.7 *Movilidad de personas en edificios: CITIC*

Un nodo de monitorización es emplazado en el CITIC[36] durante un periodo de pruebas. Se presentan aquí algunos resultados obtenidos en función del tiempo de trabajo de los trabajadores de dicho edificio.

Pese a ser conocida la línea de investigación por la mayoría del personal del edificio, el nodo de monitorización resulta totalmente discreto y nadie realiza en ningún momento ninguna apreciación ni manipulación del mismo. En la Figura 6.68 se presenta la localización del nodo, situado junto a la máquina del café, y situado enfrente de la puerta principal del edificio.

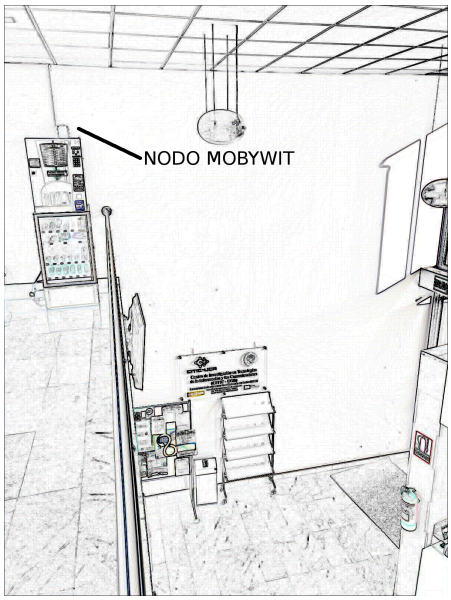


Figura 6.68
Emplazamiento del nodo de monitorización en el CITIC.

Sirven este estudio de demostración de otro escenario distinto de aplicación del sistema de monitorización propuesto, para determinar la duración de las visitas de los dispositivos detectados.

36 ↑Centro de Investigación en Tecnologías de la Informacion y Comunicación de la Universidad de Granada

Movilidad de personas en edificios: CITIC

Estudio 6.2.17: Hora de entrada y salida habitual de los trabajadores

Se toman datos de una semana, y se filtran para determinar únicamente que dispositivos pertenecen a trabajadores. Para ello se determina una jornada laboral de al menos 4 horas y acudir al edificio al menos dos días. De esa semana. Se presenta en la Figura 6.69 la predilección de hora de entrada y salida.

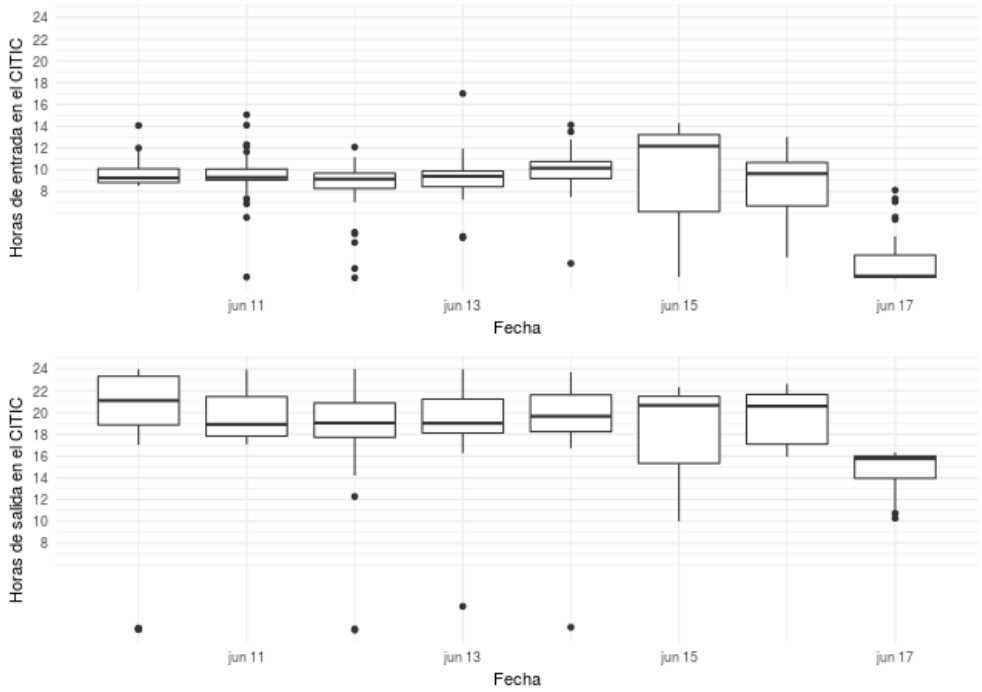


Figura 6.69
Predilección de la hora de entrada y salida de los trabajadores del CITIC a lo largo de una semana. El día 15 y 16 de junio son sábado y domingo.

Aunque la hora de entrada más frecuente se encuentra cerca a las 10 de la mañana, la hora de salida habitual es superior a las 20 de la noche. La Duración de las jornadas de trabajo de los trabajadores del CITIC se presenta en la Figura 6.70.

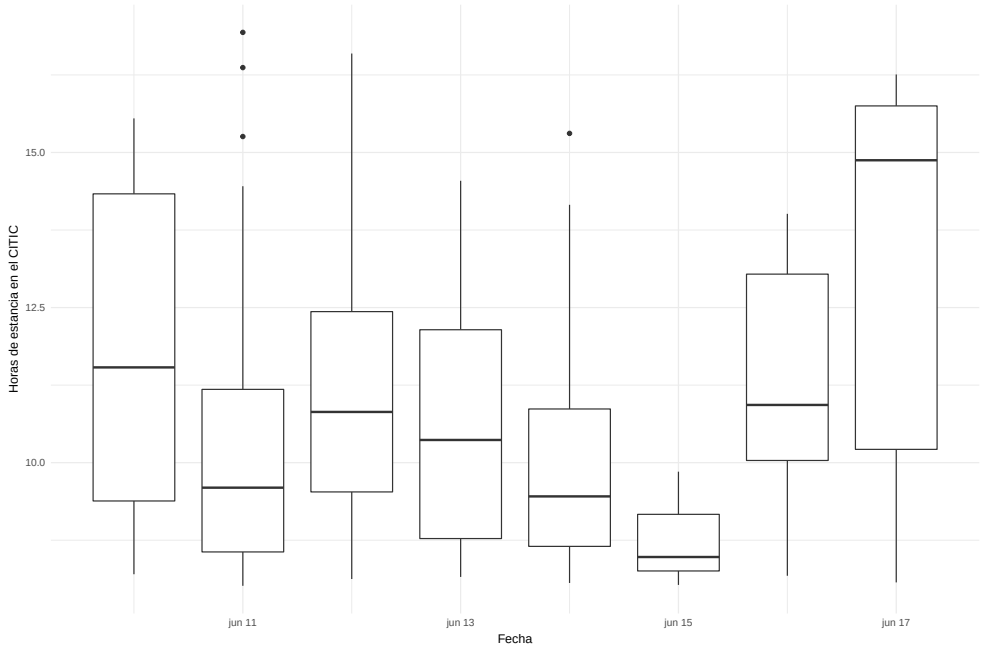


Figura 6.70
Duración de la jornada de Trabajo de los trabajadores del CITIC a lo largo de una semana. El día 15 y 16 de junio son sábado y domingo.

La mayoría de los estudiantes de doctorado y trabajadores del CITIC, permanecen en el edificio más de 10 horas diarias.

Conclusiones

El nodo de monitorización permanece totalmente inadvertido incluso ante personal investigador que conoce la línea de investigación. En ningún momento se constata ninguna sospecha.

6.2.8 *Movilidad de personas en edificios: ETSIIT*

El sistema permite controlar la estancia en edificios aunque disponga de varias puertas de acceso. Se emplazan tres nodos de monitorización en las distintas entradas de la ETSIIT[37] para estudiar el comportamiento de las personas en edificios con múltiples accesos.

Figura 6.71
Emplazamiento del nodo de monitorización en la ETSIIT.

El nodo 42 se encuentra situado en el aparcamiento subterráneo al que solamente tienen acceso los profesores y que dispone de un aforo muy limitado.

El nodo 52 se encuentra emplazado en la puerta principal exterior que lleva directamente hacia el aulario, cafetería y la zona de recreo y ocio del edificio.

El nodo 62 se encuentra en la puerta exterior del edificio principal, que permite el acceso a los despachos de profesores, comedores, cafetería, biblioteca y secretaría.

37 ↑Escuela Técnica Superior de Ingenierías en Informática y Telecomunicaciones de la Universidad de Granada.

Movilidad de personas en edificios: ETSIIT

Estudio 6.2.18: Predilección de puerta de entrada y salida

Se toman los datos de un día a modo de muestra, y se determina las puertas de primera entrada y última salida de la estancia. Se recoge en la Figura 6.72 los nodos que determinan la puerta de origen y la puerta final de la estancia del día.

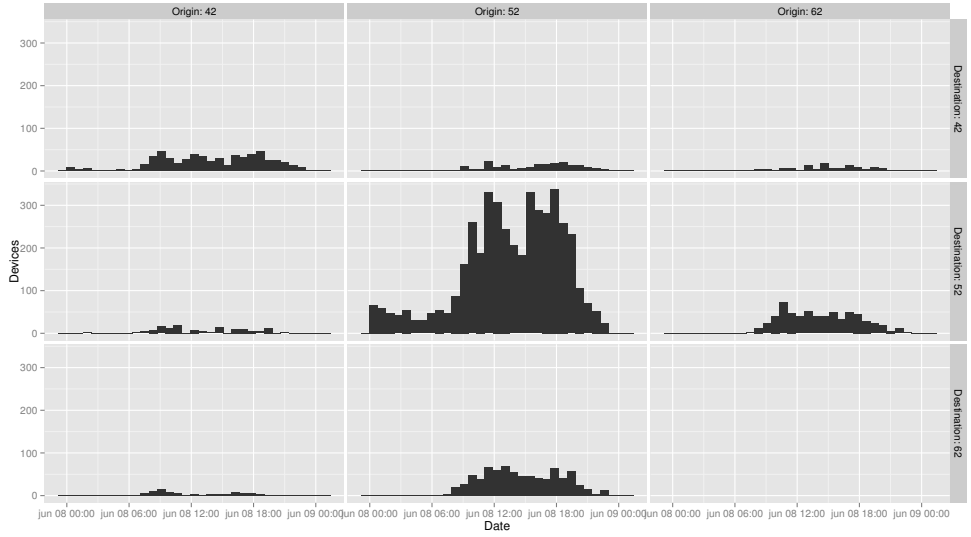


Figura 6.72
Predilección de la puerta de entrada y salida en base a los nodos de primera y última detección

Como era esperable, la mayoría de los dispositivos han entrado y salido por la misma puerta, salvo en el caso del nodo 62[38] El caso del aparcamiento arroja indicios de que hay gente que llega y sale por otra puerta (han venido con alguien), o el caso contrario, llegan andando (y entran por las otras puertas) pero se van en el coche de otra persona.

Conclusiones

Debido a que el sistema de monitorización propuesto permite la trazabilidad de los dispositivos detectados, es posible emplearlo para el control de entradas y salidas aunque el edificio disponga de varias puertas.

38 ↑Si bien no es objetivo de esta tesis realizar un estudio exhaustivo sobre el comportamiento humano, las indagaciones que se han realizado al respecto de este caso encuentran dos justificaciones. O bien la proximidad del nodo 52 al nodo 62 ha determinado que la última detección de la estancia pertenezca al nodo 52 aunque hayan pasado también por el nodo 62. O a un factor propiamente humano. Los alumnos preguntados al respecto, confiesan que para la salida normalmente no entran al edificio principal porque están deseando irse a su casa. Pero que si les resulta habitual llegar un par de horas antes a la escuela para ir a la cafetería a desayunar o a la biblioteca a estudiar. Eso puede ser la causa de la baja cantidad de dispositivos que entran y salen por puerta 62.

6.2.9 *Movilidad de personas en las calles: Anomalías*

Cualquier evento ocurriendo en las inmediaciones de los nodos que congregue masas de personas, supondrá una anomalía respecto al comportamiento normal.

Se presentan a continuación, algunos comportamientos anómalos producidos por eventos externos que han sido detectados por el sistema de monitorización propuesto.

Movilidad de personas en las calles: Anomalías

Estudio 6.2.19: Anomalía en el tránsito de personas por la calle por un concierto

Uno de los nodos emplazados del sistema se encuentra en las inmediaciones de la Plaza de Toros de Granada, lugar que alberga numerosos eventos multitudinarios a lo largo del año. En la Figura 6.73 se presenta como un concierto influye en el número de dispositivos detectados por un nodo cercano.

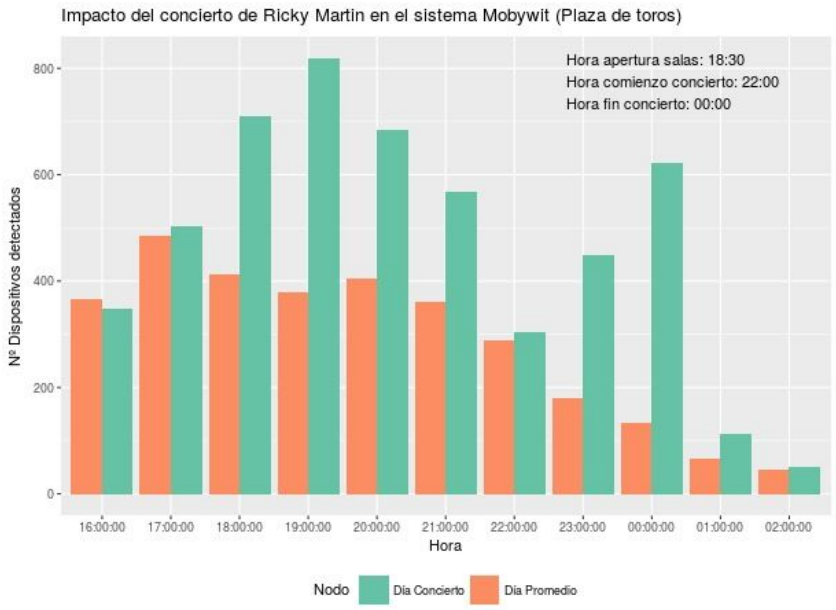


Figura 6.73
Dispositivos detectados durante un concierto en las inmediaciones del nodo comparado con los valores esperados para ese día.

Un evento, como este concierto, puede ocasionar que se doble el número de personas transitando por las calles cercanas al lugar donde tenga lugar.

La inexistencia de un sistema que cuantifique las personas transitando por la calle impide realizar ninguna comprobación adicional de la fidelidad del sistema.

Movilidad de personas en las calles: Anomalías

Estudio 6.2.20: Anomalía en el tránsito de personas por una manifestación

La ocurrencia de una manifestación en las inmediaciones de uno de los nodos demostró la capacidad el sistema para ser capaz de reaccionar con inmediatez al aumento significativo del número de personas transitando.

En la Figura 6.74 se presenta el número de dispositivos detectados durante una manifestación cercana a dos de los nodos del sistema, mostrándose un aumento significativo del número de dispositivos en las inmediaciones.

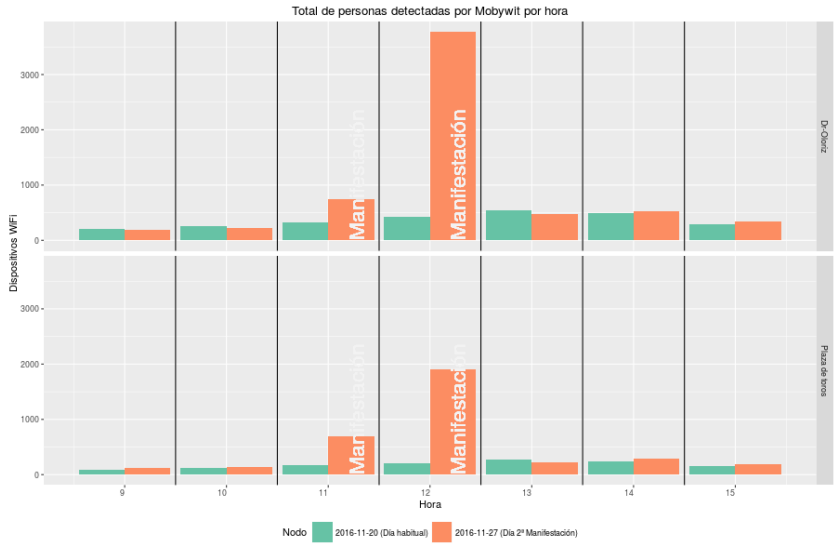


Figura 6.74
Dispositivos detectados durante una manifestación en las inmediaciones del nodo comparado con los valores del mismo día de la anterior semana.

La ocurrencia de este evento propició el tratamiento de las anomalías descripto en la Sección 5.12.5 y será abordado empíricamente en la Sección 6.3.4.

Conclusiones

Las anomalías detectadas en el sistema en cuanto a la afluencia de un mayor número de dispositivos de los esperados ha sido siempre justificada por causas que implican una mayor cantidad de personas en las calles, como los dos casos aquí presentados.

Aunque se carece de mecanismos que permitan cuantificar el número de personas, al no disponer de sistemas alternativos al propuesto, las variaciones obtenidas siempre han resultado coherentes con el escenario donde se han realizado las mediciones.

6.2.10 *Evolución del número total de dispositivos detectados*

Una vez en funcionamiento el sistema en producción, cabe la pregunta de si el número de dispositivos detectados a lo largo de mucho tiempo muestra algún tipo de tendencia.

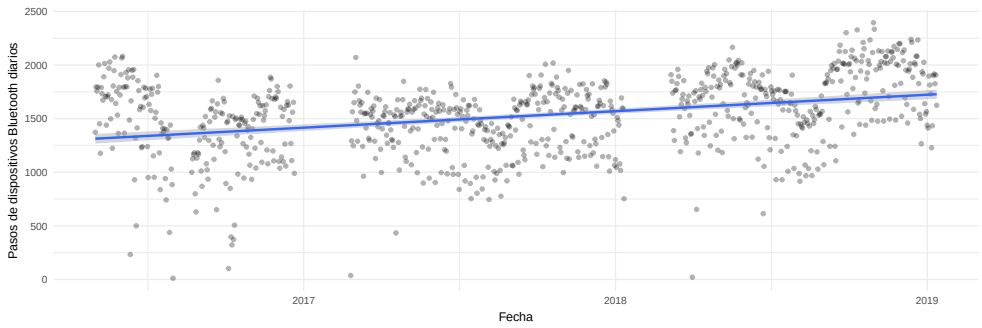
Factores como el deshabilitado de búsqueda de redes WiFi o la utilización de Bluetooth LE (Sección 4.2.1.2) en lugar de Bluetooth BR/EDR puede influir en el número de dispositivos detectables.

Un decrecimiento muy significativo del número de dispositivos detectados puede indicar la existencia de alguno de estos factores, que perjudiquen la viabilidad del sistema de captación propuesto.

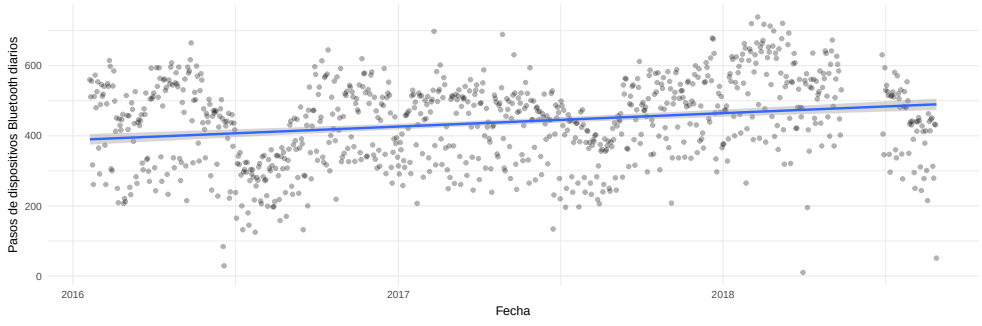
Evolución del número total de dispositivos detectados

Estudio 6.2.21: Evolución del número de dispositivos Bluetooth

En el caso de los dispositivos Bluetooth, se corre riesgo de que los vehículos y los dispositivos manos libres y multimedia que se detectan en su interior usen Bluetooth LE o mecanismos para permutar entre los distintos modos de descubrimiento (Sección 4.2.3.2).



(a) Interurbano



(b) Urbano

Figura 6.75
Evolución a largo plazo de la tendencia del número de dispositivos Bluetooth diarios detectados.

Se presenta en la Figura 6.75 las series temporales diarias de dos entornos (interurbano e urbano) a lo largo de los años de monitorización[39].

39 ↑Existen periodos de tiempo vacíos debido a errores del nodo, desconexiones, interrupciones de alimentación, manipulaciones externas. Los dos nodos mostrados fueron emplazados

El crecimiento en ambos escenarios es similar, entorno a un 5 % más de dispositivos Bluetooth detectados. Este crecimiento resulta proporcional en el resto de escenarios estudiados, aunque el periodo de tiempo sea inferior.

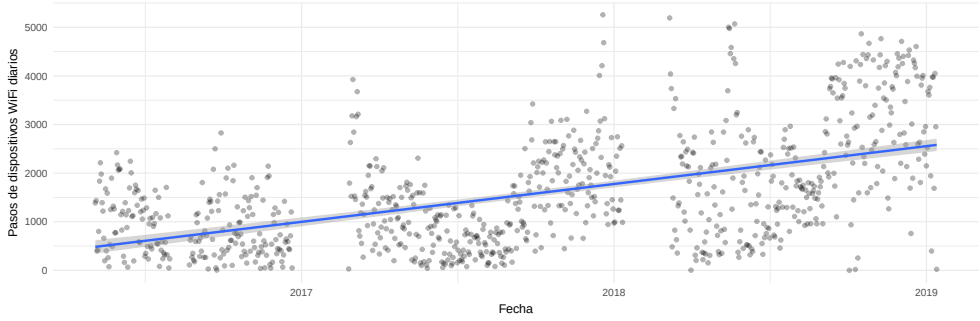
Según la DGT[40] este crecimiento parece justificarse con el rejuvenecimiento de la flota de vehículos y la incoporación de mayor cantidad de dispositivos inteligentes en los mismos.

Sin embargo, aunque no es posible presentar justificaciones causales más precisas sobre el crecimiento, no se encuentran evidencias de que el número de dispositivos Bluetooth detectados en las carreteras se haya visto influenciado por ningún factor externo que hagan peligrar la integridad del sistema. Si bien es factible el debate de si el sistema de monitorización por captaciones Bluetooth llagará a ser un sistema exhaustivo en el transcurso de los años debido a la renovación de la flota de vehículos.

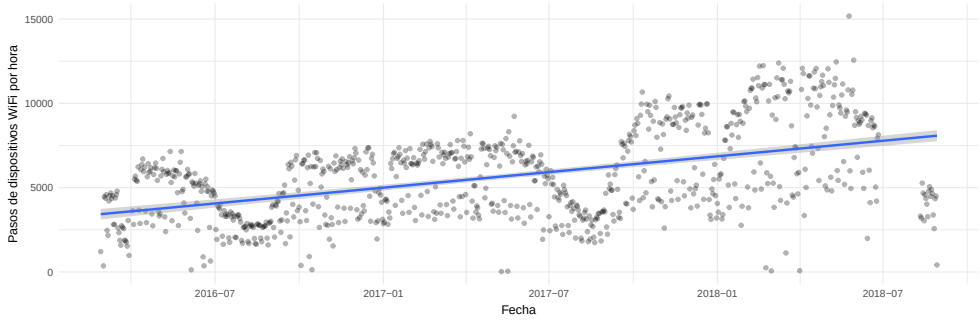
Evolución del número total de dispositivos detectados

Estudio 6.2.22: Evolución del número de dispositivos WiFi

En el caso de WiFi es factible que los nuevos dispositivos inteligentes deshabiliten la búsqueda de redes o empleen mecanismos de gestión de energía cada vez más agresivos.



(a) Periferia de la ciudad



(b) Centro Urbano

Figura 6.76
Evolución a largo plazo de la tendencia del número de dispositivos WiFi diarios detectados.

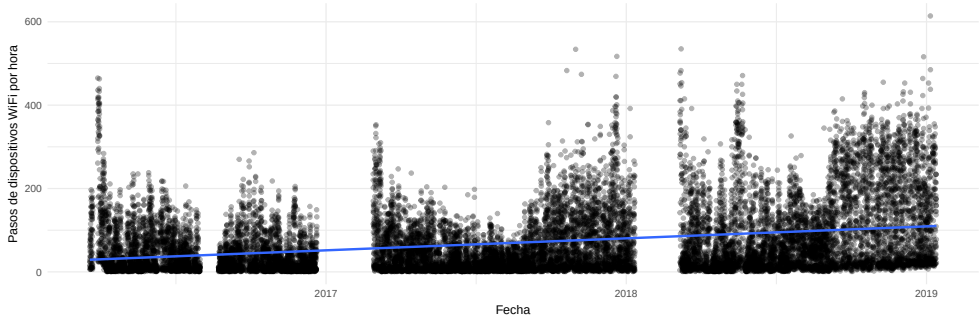
en 2016 y han permanecido en funcionamiento al término del proyecto sin ningún tipo de mantenimiento o gestión presencial.

40 ↑http://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/parque-vehiculos/

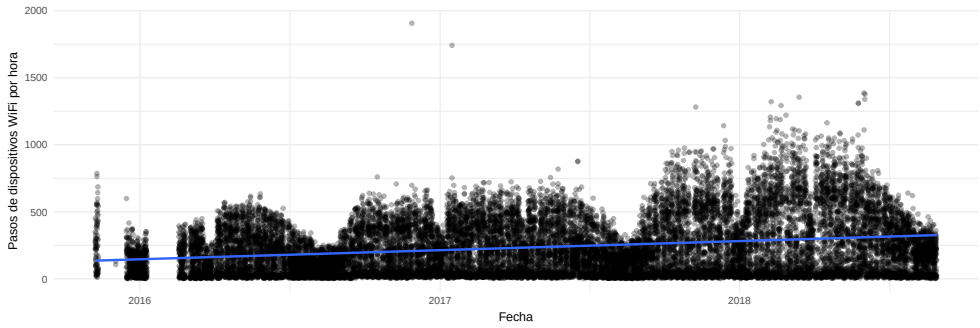
Se presenta la Figura 6.76 con las series temporales de varios años de monitorización con su línea de tendencia. Se observa un crecimiento de entorno un 35 % en el número de dispositivos WiFi detectados.

En contra a lo supuesto, el número de dispositivos detectables por WiFi está en aumento. Sin embargo esto también puede suponer un riesgo a la integridad del sistema. Aunque no se ha apreciado ningún impacto por el funcionamiento de los mecanismos de búsqueda con direcciones MAC aleatorias (Estudio 6.1.11), cabe la cuestión de si se están detectando más dispositivos de los existentes.

En la Figura 6.77 se presentan los mismos datos presentados en dispositivos por hora, en lugar de agrupados diariamente.



(a) Periferia de la ciudad



(b) Centro Urbano

Figura 6.77
Evolución a largo plazo de la tendencia del número de dispositivos WiFi por hora detectados.

Estudiando las series a nivel de hora, el crecimiento se diluye hasta un 15 % lo cual resulta tranquilizador.

En el trascurso de los 3 años los dispositivos inteligentes se ha popularizado y extendido sobre la población, como se ha presentado en la Sección 2.1. Es posible que el crecimiento de los dispositivos detectados sea debido al crecimiento del número de dispositivos en las calles. Así como la posibilidad de que varios dispositivos inteligentes, como varios smartphones, sean llevados al mismo tiempo por la misma persona[41]

41 ↑Este tema será tratado en la Sección 7 de Conclusiones de esta tesis.

También es posible que la mejora en eficiencia e interpretación de las tramas de red del software RAZIEL (Sección 5.6) sean justificación del impacto. Aunque se han obviado las diversas versiones del software[42], una mejora significativa fue desplegada en producción a finales Julio de 2017, lo cual podría justificar el incremento de los dispositivos detectados a partir de tal fecha. Sin embargo, dispositivos que solamente han hecho uso de una única versión del software como el presentado en la Figura 6.78 presentan también un crecimiento similar.

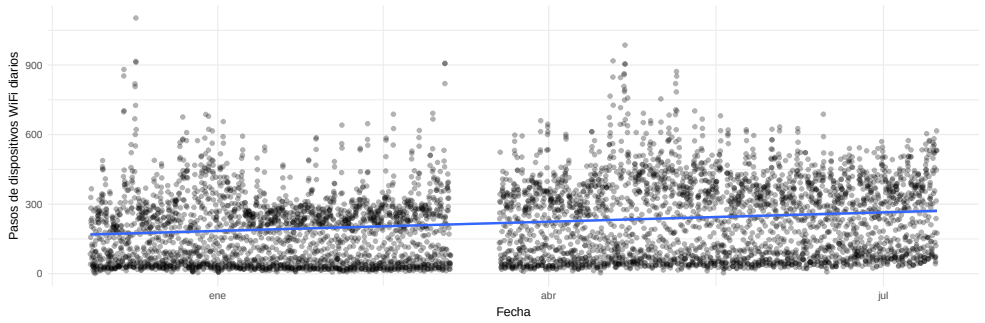


Figura 6.78
Evolución a largo plazo de la tendencia del número de dispositivos WiFi por hora detectados de nodos con una única versión del software.

No se encuentran evidencias de que los dispositivos inteligentes estén haciendo uso de mecanismos efectivos para evitar su detección por medio de la captación de comunicaciones inalámbricas.

Al contrario, se presenta un crecimiento constante y proporcional en todos los nodos sensores estudiados. Este crecimiento puede estar jusfiticado por la incorporación de cada vez más dispositivos inteligentes en la sociedad.

De igual manera que con Bluetooth, cabe lugar al debate sobre si el sistema de captación de comunicaciones se tornará exhaustivo con el paso de los años que el número de viandantes sin portar un dispositivo inteligente sea despreciable.

Conclusiones

El número de dispositivos Bluetooth y WiFi detectados por el sistema de monitorización propuesto a lo largo de los años de implantación muestra un crecimiento constante y proporcional en todos los sensores emplazados. No existen evidencias de que se hayan producido cambios en los dispositivos inteligentes para ocultarlos o dificultad su detección. Por el contrario, según las evidencias encontradas, el número de dispositivos en las calles y carreteras detectables es cada vez mayor.

En el caso de Bluetooth, parece justificarse con la renovación de la flota de vehículos y la incorporación de nuevos dispositivos inteligentes.

42 ↑Demasiado extenso ha quedado este documento, como para haber detallado las 8 versiones del software RAZIEL así como sus mejoras e innovaciones respecto a las anteriores.

En el caso de WiFi, por la popularización de los dispositivos inteligentes en el grueso de la población.

Una tercera opción, es que realmente el crecimiento del número de dispositivos detectados sea debido a aumento real y proporcional del número de personas o vehículos. Los estudios realizados para la equiparación de dispositivos y personas y vehículos deben ser recalibrados en el futuro para medir mejor este impacto.

Sin embargo, cabe lugar al seguimiento de estos estudios con el fin de verificar la consistencia del sistema.

6.3 HIPÓTESIS III: SOBRE LA APLICABILIDAD DE LA INFORMACIÓN GENERADA AL ÁMBITO DE A UNA SMARTCITY

La tercera hipótesis planteada en la tesis es que los análasis y resultados obtenidos por el sistema de monitorización propuesto pueden ser empleados para extraer conocimiento que permita gestionar de forma más eficiente una ciudad inteligente.

Se presentarán a modo de muestra algunos experimentos llevados a cabo para la predicción de series temporales, extracción de patrones, agrupamiento por características y detección de anomalías.

De igual manera que en la Hipótesis II, para minimizar la extensión de este documento cada tipo de experimento será presentado en un único escenario, aunque las técnicas presentadas hayan sido empleadas en la mayoría de los escenarios y situaciones distintos.

6.3.1 *Predicción del tráfico interurbano I*

Se presentan una muestra de los resultados de predicción realizados sobre la fuente de datos propuesta y como su utilización puede ser útil como estimador del número de vehículos y personas en el futuro.

El objetivo de estos estudios es presentar diversos escenarios distinto donde se ha probado con éxito el empleo de los procedimientos de predicción.

Predicción del tráfico interurbano I
Estudio 6.3.1: Predicción con métodos estadísticos

En la Sección 6.2.4 se han presentado series de datos basadas en el tráfico medido por el sistema propuesto, así como diversos factores de conversión del número de dispositivos detectados a vehículos.

Se presentan en esta sección algunos predictores basados en métodos estadísticos (Sección 5.11.1.3) para la estimación del número de vehículos en el futuro. Los métodos empleados están muy extendidos y se encuentran disponibles en el paquete `forecast`[127] de R [28] y son los siguientes:

[127] Automatic time series forecasting: The forecast package for R

[28] Using R for Numerical Analysis in Science and Engineering

- *Exponential smoothing state space model (ETS)*[126]: este método descompone la serie en componentes (Sección 5.11.1.1) y modela cada una de las curvas con una ecuación. Al predecir valores del horizonte de predicción, se obtiene un valor para cada componente y se vuelve a componer sumando o multiplicando las ecuaciones previas.
- *ARIMA*[33]: este método ha sido descrito en profunidad en la Sección 5.11.1.3 y se basa en el empleo de componentes autoregresivos previos al horizonte de predicción.
- *Theta*[18]: es un método de predicción univariado basado en la variación de las curvas locales de la serie temporal, descomponiéndolas en varias series modificadas que son extrapoladas por separado.
- *Mean*: El valor medio de la serie, empleado como métrica de control de los métodos presentados anteriormente.

Los datos empleados son los adquiridos por el nodo 1010, empleando un agrupación de 15 minutos. Se emplean valores del 12 al 31 de Octubre, lo que implica 1920 valores. Se decide emplear una ventana de predicción de 6 días (576 valores), el resto de valores serán empleados para el entrenamiento. Se emplea un horizonte de predicción de 1 valor. Esto implica, para cada valor a predecir se emplearan todos los valores reales obtenidos, de forma que se simule el comportamiento que se esperaría del sistema en producción.

En la Tabla 6.8 se presentan las medidas de error (Anexo A.8). En la mayoría de los casos el método `ETS` es el que más consigue minimizar el error en la predicción. En la Figura 6.79 se presentan los valores reales frente a los resultados de predicción del método `ETS`.

Tabla 6.8
Valores de error obtenidos por los métodos de predicción contra el valor obtenido por el sistema *Mobywit*.

Measure vs Method	Mean	ETS	ARIMA	Theta
ME	0.6631	0.0059	0.0043	**0.0034**
MAE	4.4611	**2.7106**	2.7409	2.7126
MAPE	4.7226	**1.0128**	1.0822	1.0599
MSE	28.5711	**13.0194**	13.2775	13.0421
RMSE	5.3452	**3.6082**	3.6438	3.6114
MASE	4.7226	**1.0128**	1.0822	1.0599
MdAE	1.4009	**0.8512**	0.8607	0.8519
MdAPE(%)	50.1142	35.0466	**33.5513**	35.0789
SMAPE(%)	68.3573	49.8147	49.7882	**49.5471**
SMdAPE(%)	54.3965	36.2454	**35.2512**	36.2994

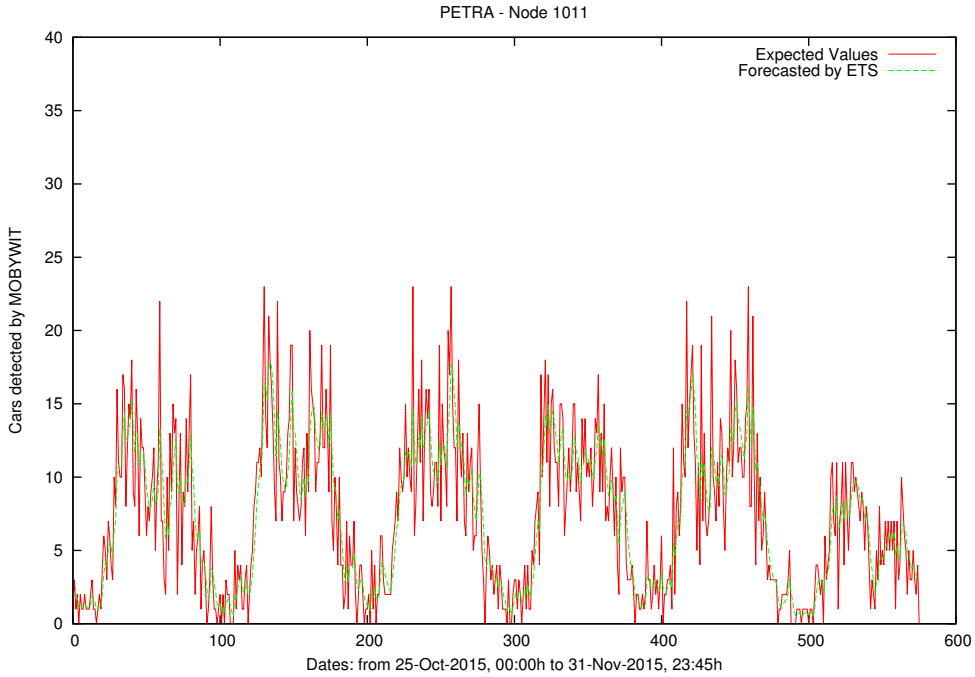


Figura 6.79
Valor obtenido por el sistema *Mobywit* (`Expected`) contra valor predicho (`forecasted`) por el método `ETS`.

Se emplea el factor de corrección[43] del ratio entre dispositivos Bluetooth y vehículos calculado en la Sección 6.2.4 para comparar con los oficiales de la DGT. La Tabla **??** presenta los errores de los métodos de predicción de comparar los valores reales obtenidos por la DGT con los valores predichos por la aplicación de los algoritmos de predicción con el ratio de conversión aplicado.

43 ↑Se emplea el mejor ratio de conversión obtenido, el basado en la mediana por cuarto de hora.

Tabla 6.9
Valores de error obtenidos por el método de predicción con la aplicación del factor de conversión frente a los datos reales ofrecidos por la DGT.

Measure vs Method	Mean	ETS	ARIMA	Theta
MAE	72.162	**42.789**	43.335	43.158
MAPE	27.961	**27.609**	27.961	27.847
MSE	7040.303	**3472.145**	3569.706	3517.506
RMSE	83.906	**58.925**	59.747	59.309

En la Figura 6.80 se comparan los valores obtenidos por la DGT mediante el empleo de aforadores con los resultados de emplear cada algoritmo de predicción con el factor de conversión.

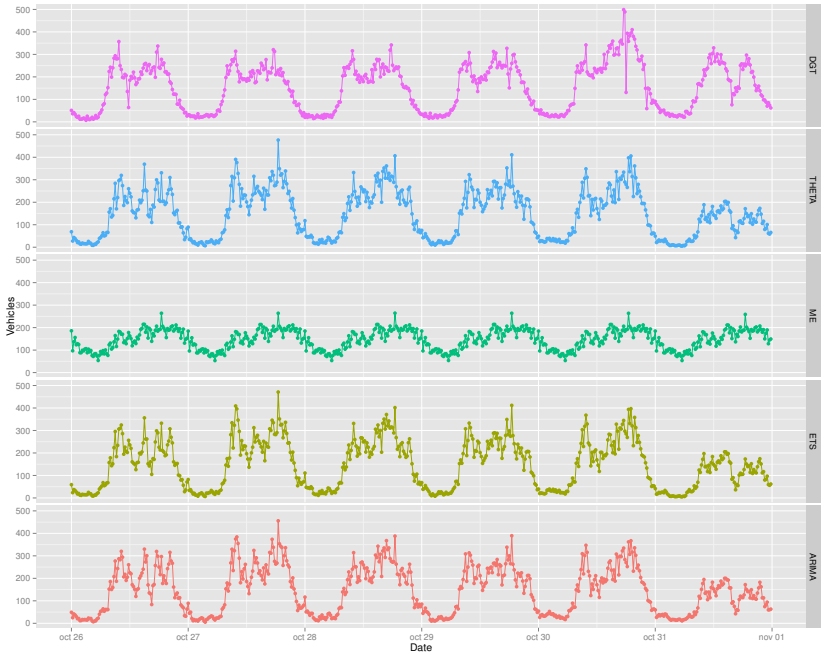


Figura 6.80
Número de vehículos detectados por los aforadores de la DGT comparado con los resultados de los distintos métodos de predicción con el ratio de conversión aplicado.

Estos métodos arrojan muy buenos resultados muy fidedignos a los reales, pero presentan varias limitaciones. Funcionan bien únicamente cuando el horizonte de predicción es un único valor, es decir, se va a predecir el siguiente valor de la serie. En el caso que nos ocupa, predecir los próximos 15 minutos. Al predecir los próximos valores del horizonte se tienen a centrar los valores de predicción en torno al valor de predicción.

Además, tienen problemas trabajando con varios factores de periodicidad (horas, días, semanas, meses) y requieren establecer un único periodo de ciclo.

Es por ello que se emplearán adicionalmente los algoritmos de predicción basados en `Machine Learning` vistos en las Sección 5.12.1.

6.3.2 *Predicción del tráfico urbano*

Los métodos estadísticos de predicción se complementan con métodos de `machine learning` presentados en la Sección 5.12.1. Se emplean emplean 3 nodos situados en zonas distintas de la ciudad, representados en el mapa de la Figura 6.82.

Figura 6.81
Mapa con la posición de los 3 nodos de monitorización.

Se va utilizar datos históricos para el entrenamiento de 451 horas y un horizonte de predicción de 24 horas. Las series temporales se presentan gráficamente en la Figura 6.82.

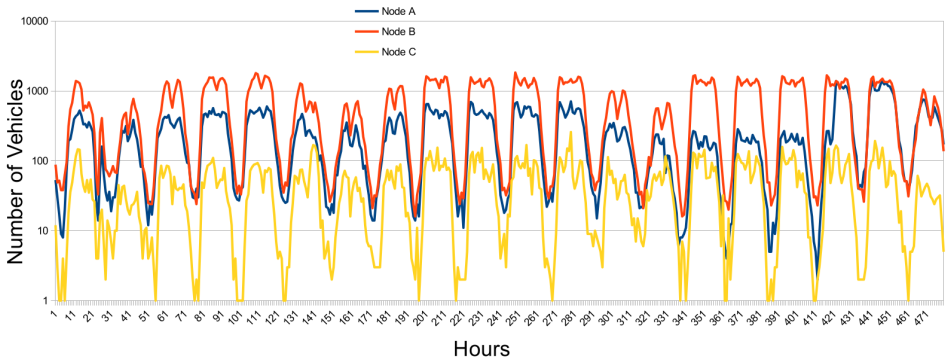


Figura 6.82
Series temporales empleadas para el entrenamiento de los modelos.

Para los métodos de predicción se emplean los parámetros por defecto de los `toolkits` empleados. Los métodos de `machine learning` son comparados con el método ARIMA. En el caso de los modelos ARIMA después de

realizar una búsqueda de los parámetros más óptimos para cada serie.[44]. Se emplea $ARIMA(1,0,9)$ para la serie A, $ARIMA(0,1,10)$ para la serie B y $ARIMA(2,0,18)$

Las métricas de error de los distintos métodos se presenta en la Tabla 6.10.

Tabla 6.10
Métricas de error de los distintos algoritmos probados para las 3 series temporales de estudio.

		MAE	MAPE(%)	MSE	RMSE	RAE(%)	RRSE(%)	DA(%)
A	ARIMA	174,67	124,11	45 751,58	213,90	218,81	214,34	54,17
	MLP	221,72	78,96	90 299,95	300,50	265,23	301,12	**79,17**
	L-CO-R	288,36	84,81	146 400,96	382,62	361,24	383,42	45,83
	SMOREG	**57,37**	**51,64**	**5118,30**	**71,54**	**72,87**	**71,69**	66,67
B	ARIMA	420,50	433,13	259 220,81	509,14	507,96	474,90	54,17
	MLP	132,78	230,87	30 287,99	174,03	160,39	162,33	**58,33**
	L-CO-R	204,81	320,63	76 271,78	382,62	361,24	383,42	45,83
	SMOREG	**52,10**	**42,92**	**4 571,18**	**67,61**	**62,94**	**63,06**	**58,33**
C	ARIMA	18,03	416,08	507,47	22,53	260,76	209,67	37,50
	MLP	16,42	283,82	508,19	22,54	237,59	209,82	41,67
	L-CO-R	17,48	183,07	571,37	23,90	252,84	222,48	**54,17**
	SMOREG	**9,44**	**119,06**	**150,85**	**12,28**	**136,50**	**114,32**	45,83

Se aprecia como `SMOReg` basado en `support vector machine for regression` es el método que obtiene los mejores resultados en las 3 series. Tan solo en métricas no absolutistas sino basas en la dirección, es superado por los métodos basados en redes neuronales. En todos los escenarios los métodos de `machine learning` consiguen minimizar mejor los errores que el método estadístico ARIMA, pese a haber sido este optimizado para encontrar sus valores óptimos.

En la Figura 6.83 se presentan los valores reales de la serie (no empleados para el entrenamiento) comparados con los valores de predicción del método.

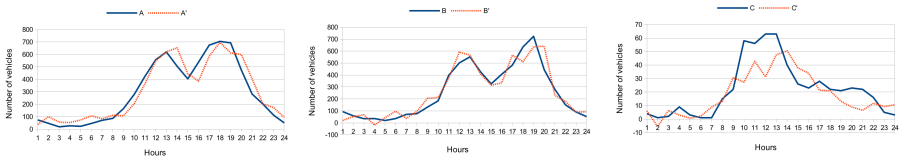


Figura 6.83
Valores reales de la serie (A, B, C) frente a los valores predichos por el método `SMOReg` (A', B', C').

Conclusiones

Los métodos de `machine learning`, y más concretamente el método `SMOReg`, resultan superiores a los métodos estadísticos cuando el horizonte de predicción implica más de un valor a predecir.

44 ↑Se emplea para ello la función `auto.arima()` del paquete `forecast` de *R*.

6.3.3 *Aprendizaje de patrones de estancias en edificios: Discoteca*

En la Sección 6.2.6 se han presentado algunos análisis realizados sobre la movilidad de personas en centros de ocio, obteniendo como resultado conjuntos de datos que son susceptibles de ser empleados para extraer conocimiento sobre los patrones de comportamiento habituales de los visitantes.

Debido a que desconocemos el patrón subyacente de comportamiento de los dispositivos individuales se trata de un problema de aprendizaje no supervisado de agrupamiento (Sección 5.12).

Se emplea un tipo especial de red neuronal (Sección 5.12.2) [118] denominada mapa autoorganizado (SOM) [152] para aprender los patrones comunes. Para ello se realizan proyecciones de una alta dimensionalidad a un espacio de baja dimensionalidad (2 dimensiones en este estudio) basado en celdas (hexagonales en este estudio) dispuestas en una rejilla. Cada neurona tiene asociado un peso vectorial del tamaño del conjunto de datos que es establecido mediante un proceso de entrenamiento competitivo con el conjunto de datos históricos.

[118] Neural Networks: A Comprehensive Approach

[152] The Self-Organizing Map

Una vez entrenada la red neuronal, se puede procesar la rejilla para obtener una matriz con las distancias unificadas (U-matrix) [272] cuyo valor de cada neurona en particular es la distancia promedio entre la neurona y sus vecinos más cercanos. Esa distancia se puede codificar por colores para facilitar su interpretación.

Se emplea la librería SOM Toolbox de MATLAB [151][45] para el entrenamiento de las redes neuronales y la visualización de los mapas autoorganizados.

Aprendizaje de patrones de estancias en edificios: Discoteca

Estudio 6.3.2: Aprendizaje sobre datos de una noche

Con los tiempos de estancia de los dispositivos de una noche en la discoteca se compone un conjunto de entrenamiento con las características que se presentan en la Tabla 6.11.

Tabla 6.11
Características del conjunto de entrenamiento de datos de una noche

Variable name	Description	Type
entrance_time	Instante de tiempo de primera detección (entrada)	Date
out_time	Instante de tiempo de última detección del dispositivo (salida)	Date
stay_time	Duración de la estancia en segundos del dispositivo	Integer
abs_time_node_X	Número total de segundos de estancia del dispositivo por cada nodo sensor	Integer
relat_time_node_X	Porcentaje de tiempo sobre el tiempo total de estancia por cada nodo sensor	Float
relat_night_time_node_X	Porcentaje de tiempo sobre el tiempo total de estancia por cada nodo sensor relativo a la duración del periodo de apertura.	Float

45 ↑Agradecimientos a Amorag quien fue quien realizó la ejecución de los primeros estudios basados en SOM que son aquí presentados. Posteriormente se hizo uso de la librería Kohonen de R para el entrenamiento de las redes SOM:
https://cran.r-project.org/web/packages/kohonen/index.html

Los resultados del entrenamiento del SOM se presentan en la Figura 6.84, donde se representan los valores del vector de pesos de cada una de la neuronas para una característica concreta.

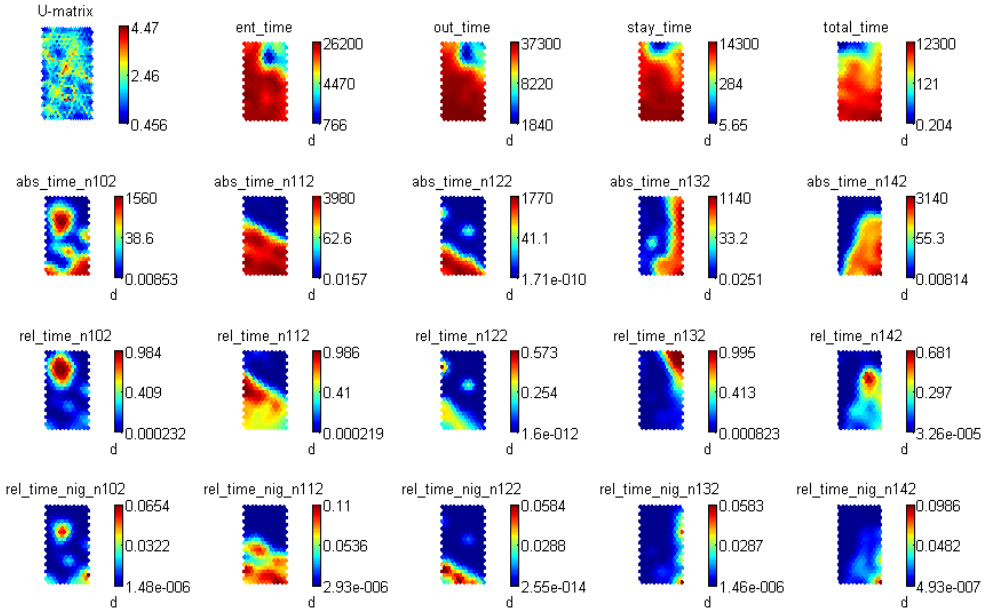


Figura 6.84
Proyección de los valores de la neurona de cada característica.

Los distintos clusters o zonas son determinados por la distribución de los distintos valores sobre la proyección. Un caso muy evidente, es como se relacionan los tiempos de entrada y salida con el tiempo de estancia. Esto se evidencia en las regiones coloreadas del mismo tono en la proyección en distintas características.

Así como por ejemplo se puede extraer una predilección muy fuerte por el nodo 102 en un grupo de personas cuyo de tiempo de estancia es medio[46].

La distancia euclídea de todas las características para cada neurona es representada visualmente en la Figura 6.85 que recoge la U-Matrix del Mapa. Las muestras que han presentado similitudes son mapeadas juntas y las que muestran diferencias presentan mayores distancias.

46 ↑El nodo 102 se encuentra emplazado en una terraza exterior, lo cual puede explicar dicho patrón.

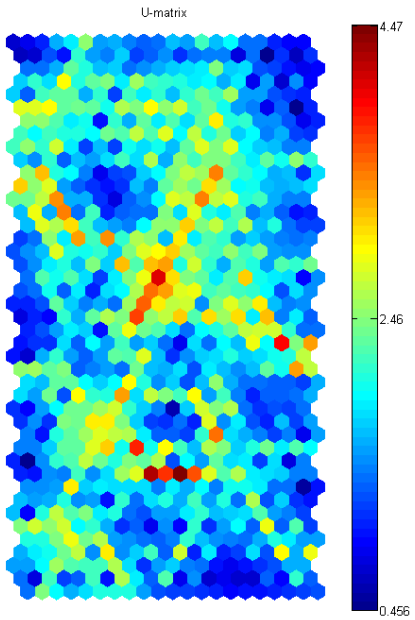


Figura 6.85
`U-Matrix` con la distancia entre las distintas neuronas del `SOM`.

Las distancias entre neuronas se presenta de uniformemente distribuida a lo largo de la red, únicamente algunas neuronas centrales presentan mayor distancia.

El cluster superior derecho corresponde a visitantes que entran y se van temprano, lo cual podría corresponder según indicaciones de los responsables a gente que sale del trabajo y echa un refresco o cerveza antes de volver a casa y de que comience la hora cuando se sube la música.

Existe también un patrón de gente que permanece la mayor parte el tiempo en la entrada (Nodo 132) pero abandona el local, mostrando tiempos absolutos de estancia muy cortos. Esto puede justificarse que entra solamente para echar un copa y no accede al resto de salas.

La sala principal (112) es donde se agolpa la mayor parte de las personas, con un varabilidad del tiempo de estancia considerable. Sin embargo los mayores tiempo de estancia se asocian a visitantes que pasan la mayor parte del tiempo en la sala del nodo sensor 142, que es más pequeña que la principal y tiene un estilo de música más alternativo.

Aunque los datos de una única noche nos pueden permitir extraer información sobre los patrones de los visitantes de ese momento, es deseable extraer patrones de comportamiento de varias noches distintas.

Aprendizaje de patrones de estancias en edificios: Discoteca

Estudio 6.3.3: Aprendizaje sobre visitantes recurrentes

Cualquier negocio busca fidelizar a sus clientes, por lo que extraer información sobre los visitantes habituales puede servir para emprender sus patrones de comportamiento que presentan sus gustos habituales.

Se analizan las noches de verano del centro de ocio, centrándose en los dispositivos que han sido detectados al menos dos noches distintas. Se dispone de conjunto de datos de 2100 dispositivos distintos. Se etiqueta cada dispositivo con el número de noches distintas que ha sido detectado.

Para cada dispositivo se estudia el tiempo que ha pasado por cada noche en cada una de las salas disponibles, en términos tanto relativos a su estancia como en términos absolutos.

La Figura 6.86 presenta la `U-Matrix` resultante. Cada neurona ha sido etiquetada con la clase predominante o mayoritaria. Se presentan 3 grandes `clusters` (las áreas azules) que maximizan el número de visitas realizadas.

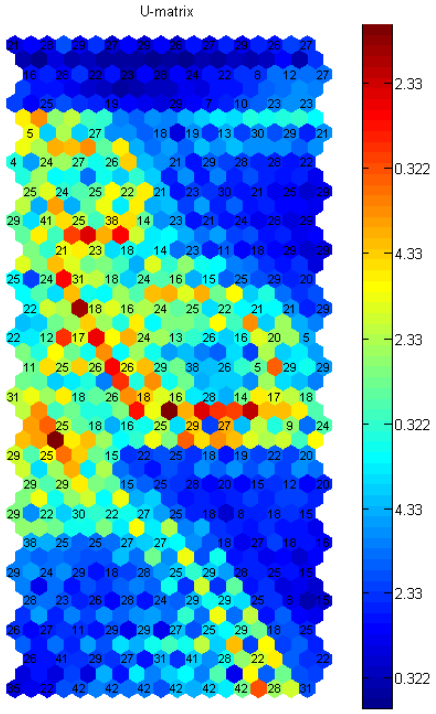


Figura 6.86
U-MATRIX del clustering de visitas recurrentes.

La Figura 6.87 presenta la proyección de las características, que nos permiten interpretar los distintos clusters encontrados.

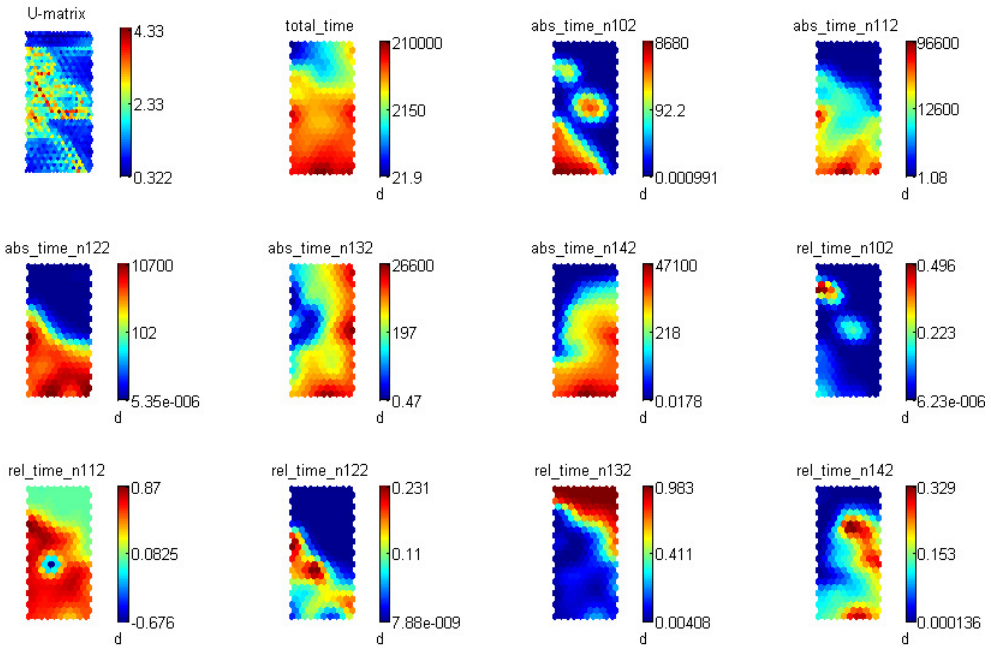


Figura 6.87
Proyección de los valores de la neurona para cada característica en visitas recurrentes

Existe correspondencia entre los visitantes más habituales con parte de aquellos que pasan mayor cantidad de tiempo en las inmediaciones del nodo 102 situado en la terraza y con aquellos que pasan la mayor cantidad de tiempo en la zona del nodo 132[47]. Estos zonas son las que concentran la mayor cantidad de personas habituales en el local.

EL tercer cluster se presenta con la gente que frecuencia las salas de los nodos 142 y 122, pero sin un patrón tan claro como en los casos anteriores.

La sala donde se encuentra situado el nodo 112 es la sala principal, más grande y con mayor cantidad de eventos y espectáculos programados. Sin embargo, no se encuentran entre sus visitantes ningún patrón que defina la permanencia en esta sala con las visitas recurrentes.

Este tipo de estudios que se presentan aquí de forma puntual, pueden ser realizados a lo largo del tiempo o para medir si nuevas políticas o eventos han tenido algún tipo de impacto en la fidelización del público. De igual manera, se pueden estudiar los factores periódicos habituales.

47 ↑Esta sala, al encontrarse situada en el punto de entrada es la primera en ser abierta y tiene una disposición más similar a un pub.

Aprendizaje de patrones de estancias en edificios: Discoteca

Estudio 6.3.4: Aprendizaje sobre el día de la semana

Dado que cada día el tipo de eventos y de público asistente resulta ser distinto, se realiza el mismo tipo de estudio pero incluyendo el día de la semana como variable. Se emplea un conjunto de 200 000 visitantes/noches. En la Figura 6.88 se presenta la convergencia del método de entrenamiento.

Figura 6.88
Discoteca: Evolución del proceso de entrenamiento del SOM

Se ha empleado un conjunto de datos que incluye el día de la semana, el tiempo de estancia, la hora de entrada y salida y el porcentaje de detección en cada sala. En la Figura 6.89 se presentan las proyecciones de la estancias y las hora de entrada y salida, así como la U-Matrix y la densidad. En la Figura 6.90 se presentan las proyecciones de las estancias en cada sala.

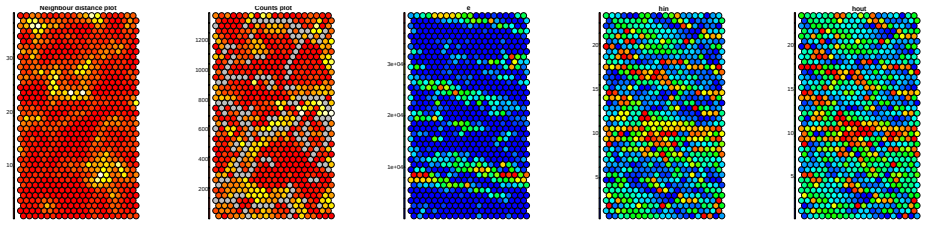


Figura 6.89
U-Matrix, reparto de instancias en las neuronas y proyección de la hora de entrada, salida y tiempo de estancia.

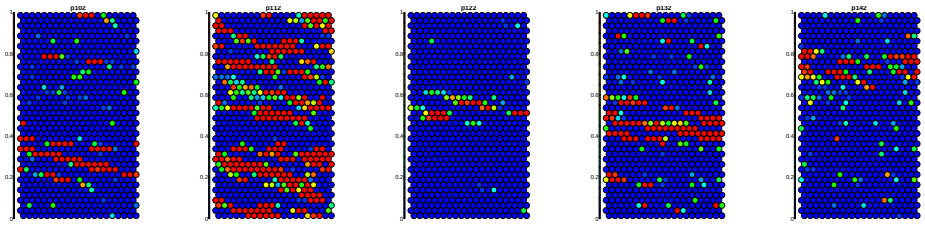


Figura 6.90
Proyección de la estancia en cada sala.

Como se han creado variables para cada día de la semana, se pueden proyectar los distintos días. La Figura 6.91 recoge las proyecciones de los días entre semana y la Figura 6.92 los fines de semana.

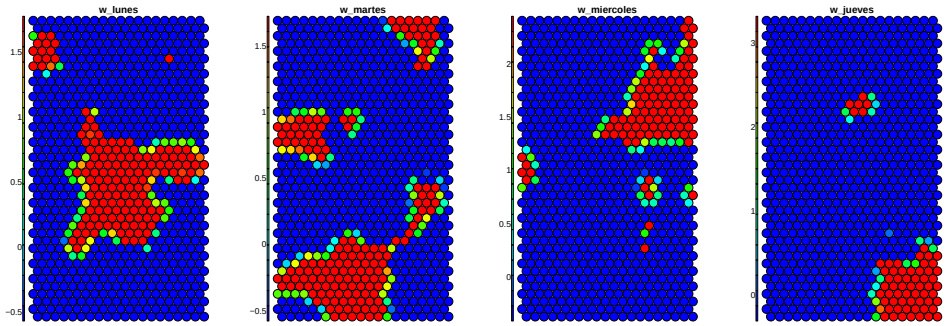


Figura 6.91
Proyección de los días entre semana.

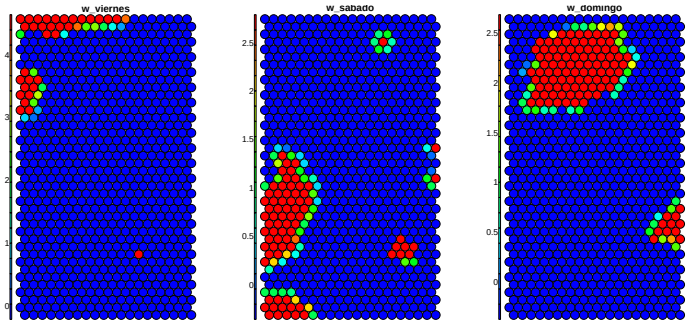


Figura 6.92
Proyección de los fines de semana.

Se aprecia como los visitantes de la sala del nodo 112 de los martes y sábados son los que alcanzan un mayor tiempo de estancia. Se encuentran 4 `clusters`, dos de ellos muy diferenciados.

Conclusiones

Aunque la información generada sea en apariencia escasa y limitada al tiempo de detección, la posibilidad de explotar los datos de varias visitas recurrentes o enmacarlas en un periodo concreto, permite localizar patrones. Estos patrones nos dan información sobre las preferencias de comportamiento de la mayoría de los visitantes y es un excelente mecanismo para la justificación de nuevas políticas o la toma de decisiones por el personal administrador.

6.3.4 *Análisis y detección de manifestaciones*

En la Sección 6.2.9 se ha presentado como el tránsito de personas se ve alterado por causas externas como un concierto o una manifestación.

Una protesta o manifestación implica un elevado número de ciudadanos en las calles manifestando su apoyo o protesta hacia alguna causa. Generalmente, el éxito de la manifestación se mide en función del número de personas que se unen a ella. Sin embargo los métodos empleados para contar a los manifestantes son bastante rudimentarios y en la mayoría de los casos, tiene escasa precisión. Es por ello, que ante una manifestación, los organizadores, el gobierno y los medios presentan siempre cifras de asistencia muy distintas entre ellas.

Actualmente el modo más extendido para medir manifestación consiste en analizar imágenes aéreas tomadas durante la manifestación y contar las cabezas de personas que aparecen en ella. Sin embargo, en manifestaciones multitudinarias, esta es una tarea titánica que en muchas ocasiones es realizada por personas. Es por ello que una práctica común consiste en dividir las fotos en cuadrados de la misma superficie y contar las cabezas en uno sólo de los cuadrados. El número total de manifestantes se estima en base al número de cuadrados en los que se ha divido la foto multiplicado por el número de cabezas contadas en dicha foto. Este método se denomina método de Jacobs [179].

Sin embargo este método presenta muchas limitaciones, debido a que la concentración de personas no es uniforme en todos los cuadrados, la obtención de cuadrados de igual área usando fotos es dificultosa debido a la perspectica y porque no trata bien con el mobiliario urbano fotografiado en la fotos. Además, su mayor limitación es que implica directamente a seres humanos, que son los encargados de realizar tanto la división como el conteo. Aunque actualmente se ha intentado automatizar el proceso [19], pero su aplicación no ha tenido mucho éxito.

En colaboración con el Área de Mobilidad del Ayuntamiento de Granada, como se ha comentado anteriormente, se dispone de nodos colocados en la ciudad. Algunos de esos nodos se encuentran situados cerca de la zona donde es habitual que comiencen las manifestaciones, como se muestra en la Figura 6.93.

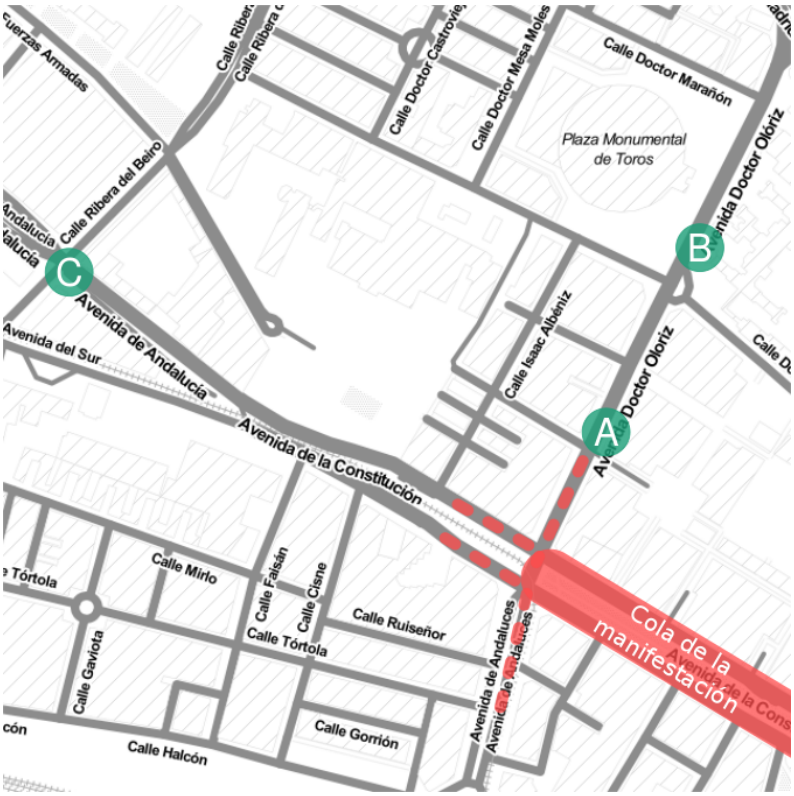


Figura 6.93
Mapa del escenario de las manifestaciones, con los tres puntos estudiados (A,B,C) cercanos al punto de inicio o cola de la manifestación

Se han analizado tres protestas y una concentración que han tenido lugar cerca de los nodos indicados. Todas las protestas y concentraciones tuvieron lugar en Domingo, convocadas a partir de las 12 am. Estos días han sido comparados con días habituales, también domingo y durante el mismo periodo de tiempo.

Tabla 6.12
Estimación del número de manifestantes por la policía local.

DÍA	NÚMERO DE MANIFESTANTES (ESTIMACIÓN DE LA POLICÍA LOCAL)
Día protesta 1	22 000
Día concentración 1	5 000
Día protesta 2	40 000
Día protesta 3	55 000

La existencia de un incremento en el número de dispositivos detectados, y por tanto de personas, en los momentos de la protesta mostraría que el sistema puede ser viable para analizar el número de manifestantes. Estos análisis se pueden centrar únicamente en el conteo de personas en la manifestación como, gracias a la capacidad de detección unívoca del sistema, a extraer hábitos en los manifestantes.

6.3.4.1 *Cuantificación de la manifestación*

Cuantificación de la manifestación

Estudio 6.3.5: Número de personas detectadas por hora

Cómo se ha indicado anteriormente los nodos *Mobywit* son útiles para contabilizar el número de dispositivos (personas) que han pasado por un sitio en particular, cercano a dicho nodo, a lo largo de distintos periodosd de tiempo. La Figura 6.94 representa el número de dispositivos registrados por hora para cada uno de los ocho días de estudio.

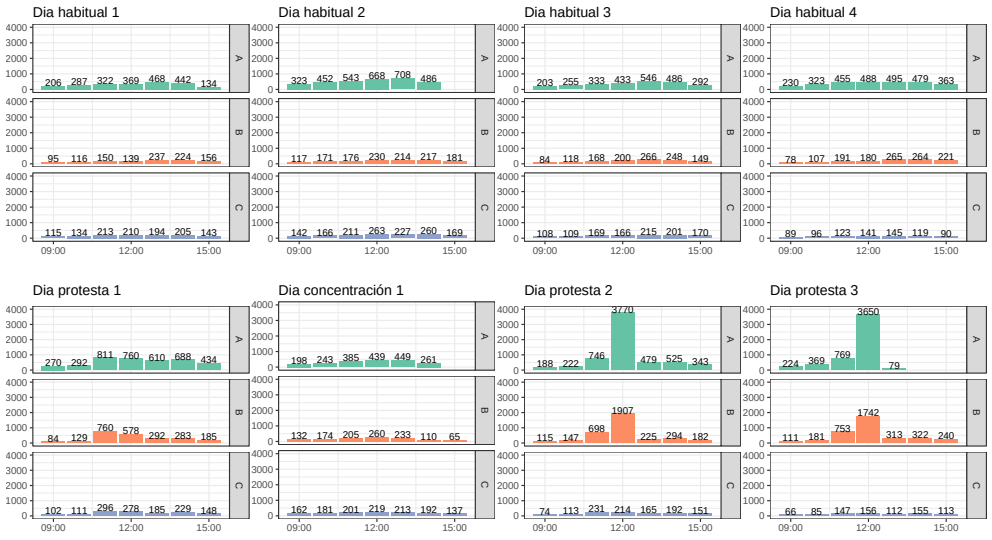


Figura 6.94
Número de dispositivos detectados por hora en los ocho días de estudio en 3 nodos diferentes.

Los días normales presentan una detección similar en los tres nodo, sin diferencias estadísticamente significativas según el test de ANOVA, como se muestra en la figura 6.95. Para la realización de dichos test, se ha demostrado con el test de Saphiro que la distribución de dispositivos en ese periodo de tiempo sigue se ajusta a la normalidad.

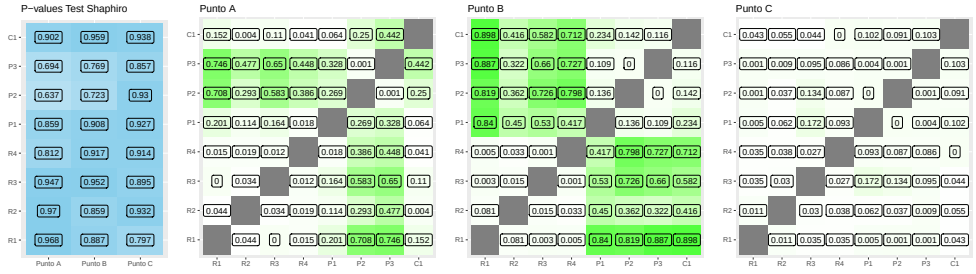


Figura 6.95
P-values del test de Shapiro para la normalidad y p-values de los tests ANOVA realizados entre los distintos conjuntos de datos.

En el punto A, el punto más cercano al punto de comienzo de la manifestaciones, el test ANOVA arroja que existen diferencias estadísticas significativas entre los días normales y los días de la segunda y tercera protesta. En el punto B, para todos los días de manifestaciones y protestas se presentan diferencias significativas respecto a los días normales. Para el punto C, el más alejado del principio de la manifestación, no se presentan diferencias respecto a los días normales.

De estos tests se puede extraer que una manifestación tiene un impacto claro en el número de dispositivos inteligentes en las calles, debido a que se congregan mayor cantidad de personas en las zonas cercanas a los nodos.

Cuantificación de la manifestación

Estudio 6.3.6: Incrementos de personas por hora

Debido a que el sistema no es exhaustivo, la manera precisa de presentar la información sobre el número de dispositivos que han sido detectados, es ofrecer respecto a los incrementos que se han presentado en comparación con los días normales. La Figura 6.96 muestra el incremento en el número de dispositivos detectados en las inmediaciones para los días de manifestaciones y concentraciones.

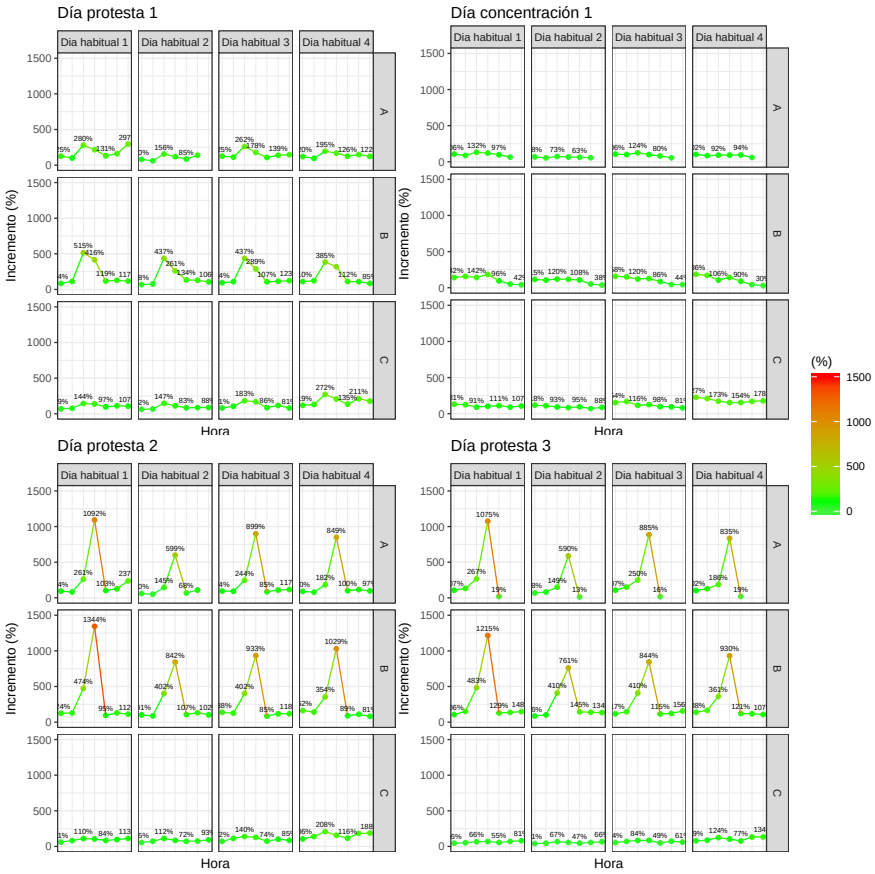


Figura 6.96
Incrementos del número de dispositivos durante las manifestaciones por hora basado en los días normales para cada día de protesta.

En el caso del segundo y tercer día de protesta, se capturan del orden de diez veces más dispositivos en las inmediaciones de lo que es habitual, lo que puede aproximarse a un aumento proporcional en el número de personas en las calles.

En el día de la primera protesta, el aumento es menor en torno a cinco veces más dispositivos. Por último durante el día de la concentración no se presenta ningún incremento significativo del número de dispositivos en las inmediaciones.

Esta información resulta de relevancia para medir el impacto de las manifestaciones en la ciudad, ya que se puede indicar la cantidad de personas adicionales que se han concentrado en las inmedicaciones fruto de la manifestación, y que de no haber sido por esta, no se hubiesen detectado en base a la información histórica registrada.

Cuantificación de la manifestación

Estudio 6.3.7: Resultados minuto a minuto

Debido a la naturaleza del sistema es posible reconstruir la manifestación en base al número de dispositivos en las inmediaciones para cualquier instante de tiempo. La Figura 6.97 muestra los dispositivos que estaban siendo detectados en instante de tiempo concreto durante la segunda manifestación en el punto 2. Se presentan dos instante de tiempo distintos: a las 10:00 (antes de la hora de comienzo de la manifestación) y a las 12:00 donde el número de dispositivos siendo detectados simultáneamente es mucho mayor.

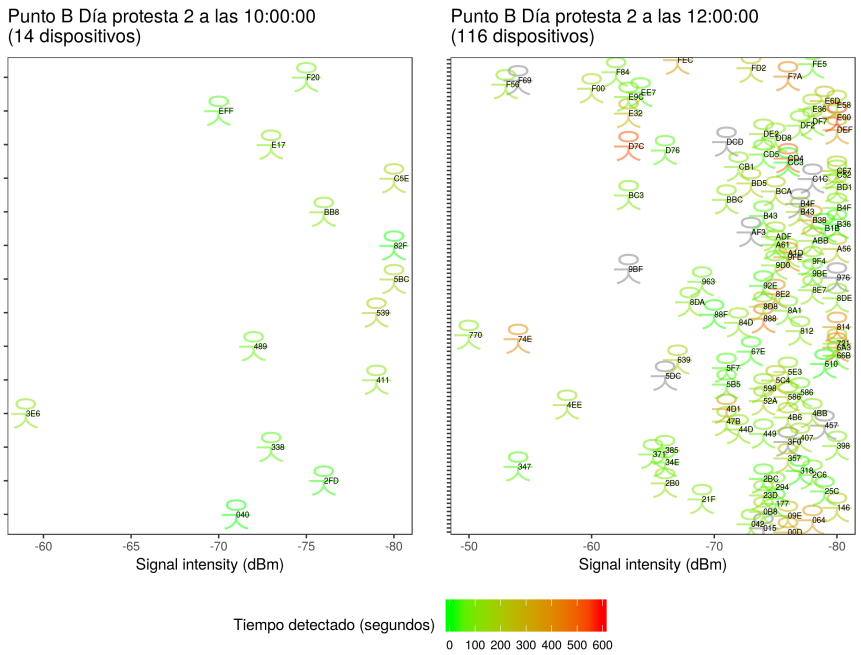


Figura 6.97
Detalle de los dispositivos detectados en dos instantes de tiempo diferente en el Punto B durante el segundo día de protesta. Cada dispositivo está representado por los tres dígitos hexadecimales más representativos de la `MAC ADDRESS`. El color de cada punto representa el tiempo que el dispositivo ha sido detectado. El eje x representa la intensidad de la señal-

Esta información resulta muy útil para ser observada en tiempo cercano al real durante el transcurso de la protesta para conocer el estado de la misma o para ser estudiada y analizada a posteriori. Además, estos datos son susceptibles de ser animados, para presentar una animación concreta

Cuantificación de la manifestación

Estudio 6.3.8: Análisis de recurrencia

Debido a que el sistema es capaz de reconocer a los dispositivos detectados en visitas sucesivas, es posible determinar si un dispositivo es habitual en un escenario o, en oposición, su detección es algo excepcional. La Figura 6.98 muestra el número de dispositivos que han sido detectados en los 4 días normales, por lo que su detección resulta algo habitual.

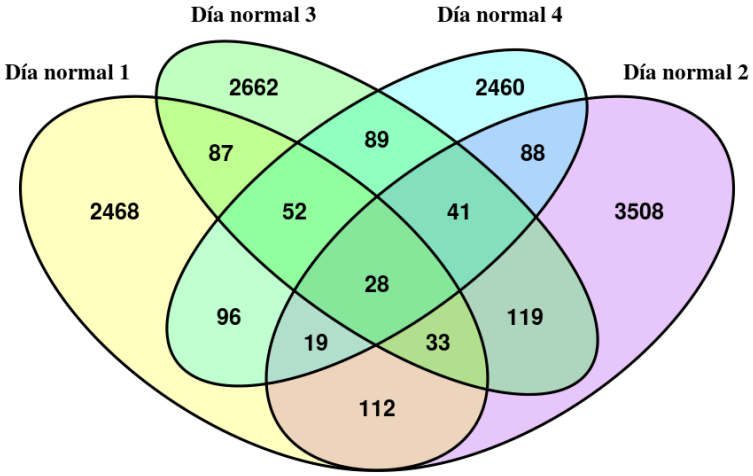


Figura 6.98
Dispositivos detectados en días recurrentes normales.

Debido a que esos dispositivos son habituales, pueden ser eliminados para determinar que dispositivos han sido detectados en varias manifestaciones, lo cual se recoge en la Figura **??**. El número de dispositivos detectados los días de protestas, pero no detectados los días habituales entre las Protestas 2 y 3, es significativo, siendo de un 15 %.

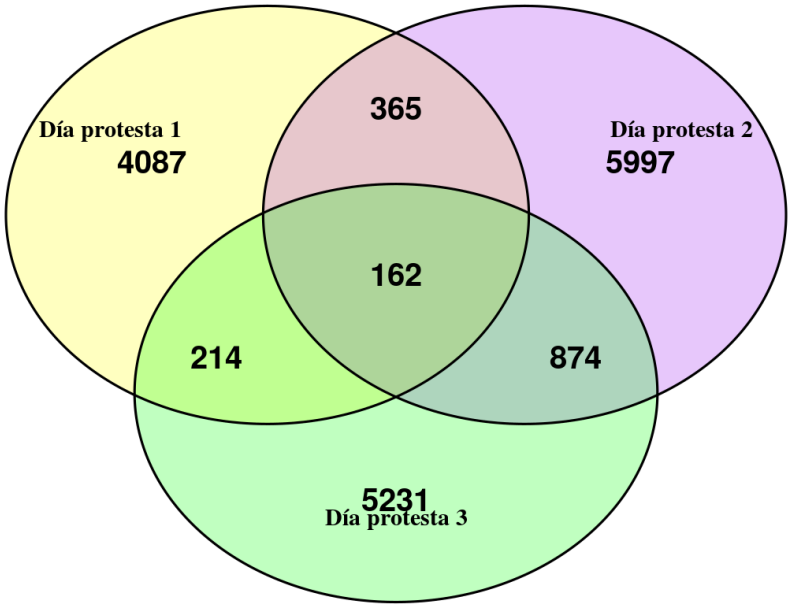


Figura 6.99
Dispositivos detectados en días recurrentes con manifestación.

6.3.4.2 *Detección de anomalías en manifestaciones*

Demostrado que el sistema propuesto puede ser empleado para cuantificar de forma aproximada las personas asistentes a una manifestación, se emplea el método `S-H-ESD` presentado en la Sección 5.12.5. Este método, como se ha descrito anteriormente, requiere que las series temporales a usar sean estacionarias y periódicas.

Detección de anomalías en manifestaciones
Estudio 6.3.9: Estudio de las series temporales

Las series temporales que van a ser empleadas son las generadas por el agrupamiento de pasos y simultaneos en los 3 puntos a distintos tamaños de ventana de muestreo. Esta series, para el el empleo del método S-H-ESD requieren ser estacionarias. Para demostrar que las series empleadas son estacionarias se emplean los test Augmented Dickey-Fuller Test (ADF) [93], Kwiatkowski-Phillips-Schmidt-Shin Test (KPSS) [120, 156] and Phillips-Perron Test (PP) [93, 208]. Para todas las series, el p-value resultante es inferior a 0.01 en todos los test, para los 3 tipos y para distintos tamaños de lags, por lo que se puede concluir que las series son estacionarias.

Adicionalmente, el método propuesto para la detección de anomalías está específicamente diseñado para su empleo en series con un fuerte componente periódico o `seasonal`. En las Figuras 6.100 y 6.101 se presentan las descomposiciones de las series temporales en sus componentes, donde se puede evidenciar que las series temporales empleadas son influenciadas por múltiples factores periódicos (días, semanas o meses).

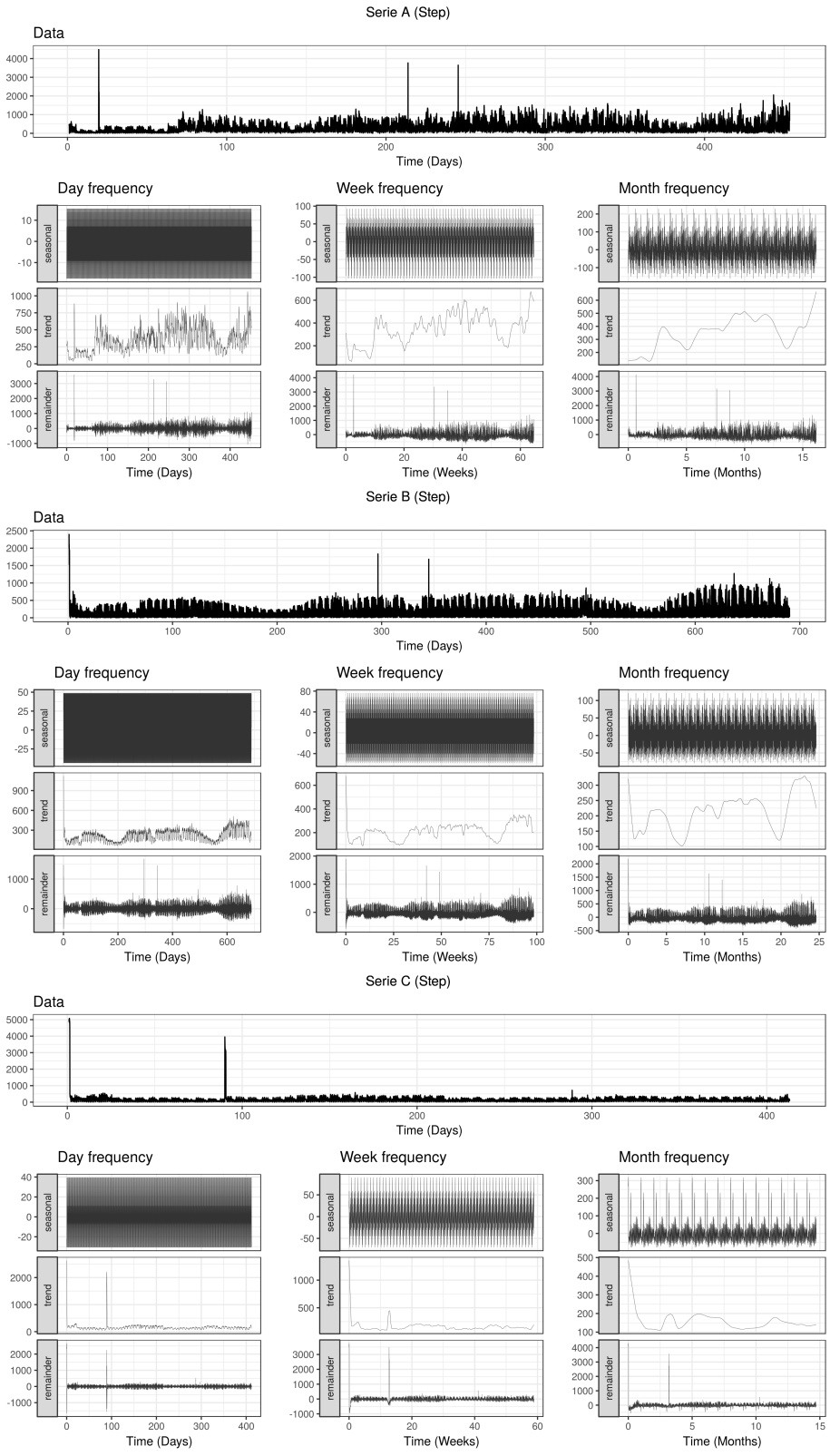


Figura 6.100
Descomposición en componentes de la series temporales de pasos (`steps`).

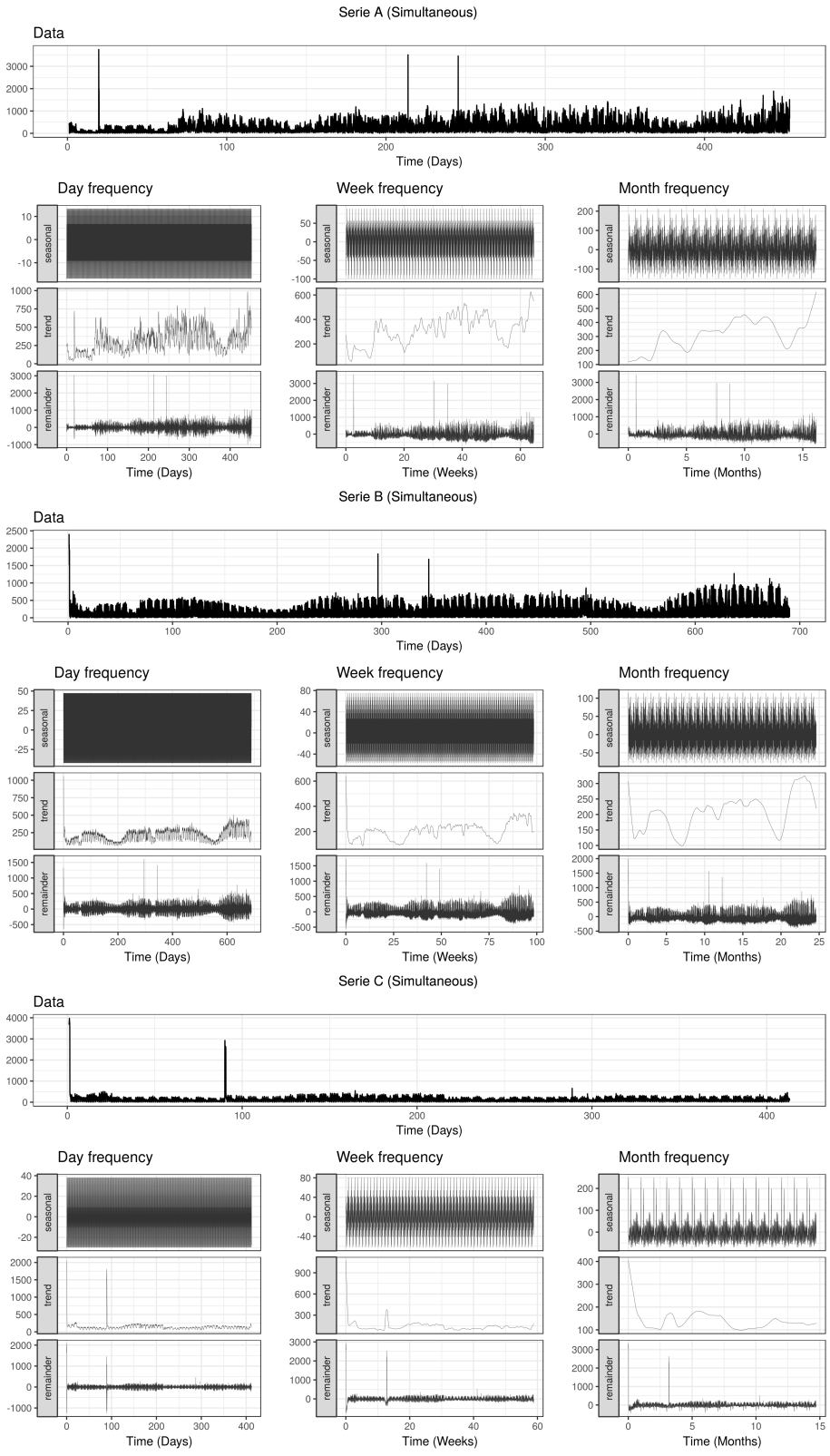


Figura 6.101
Descomposición en componentes de la series temporales de simultáneos (`simultaneous`).

Detección de anomalías en manifestaciones

Estudio 6.3.10: Detección de anomalías

Con las comprobaciones necesarias para la consideración de las series realizadas y satisfactorias, se procede a la ejecución del algoritmo `S-H-ESD` para que detecte anomalías en las series. Se limita el número máximo de anomalías detectables por el algoritmo `S-H-ESD` a un máximo del 5 % del tamaño de la serie.

El algoritmo es ejecutado para las series de los 3 puntos (A,B y C) con la información tanto del agrupamiento de pasos como de simultáneos. Además, para el agrupamiento se emplean varios tamaños de ventana distintos: 60,30,15,5 y 1 minuto.

La salida del algoritmo obtiene una lista de intervalos de tiempo que han sido considerados anómalos, así como el valor que el algoritmo había estimado para esos instantes de tiempo.

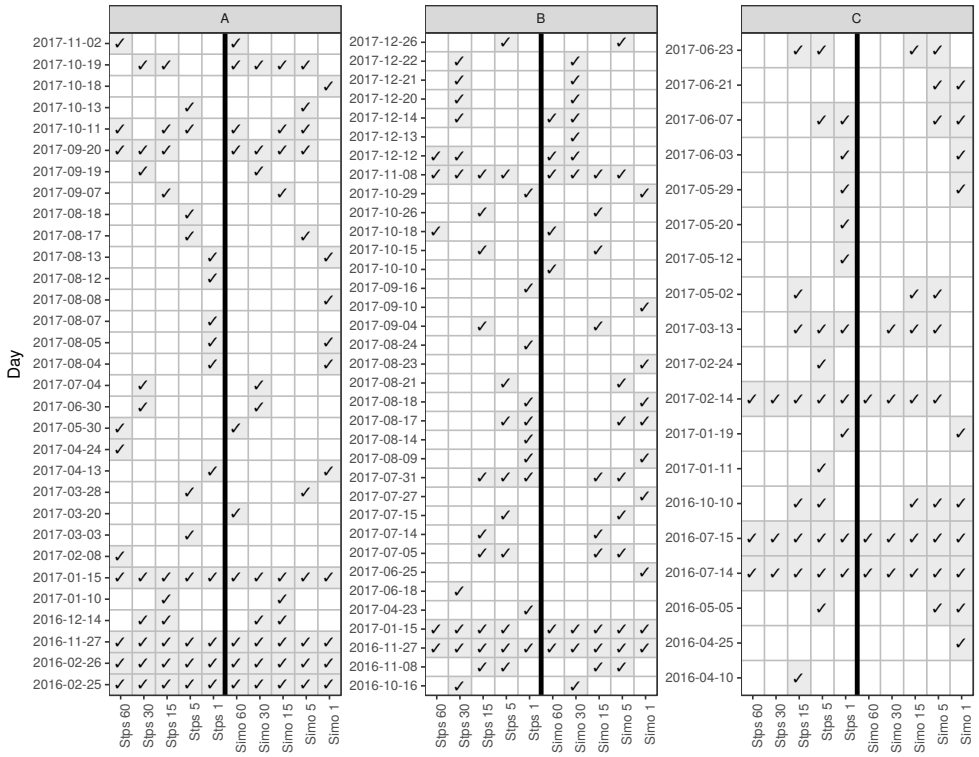


Figura 6.102
Anomalías detectadas en las series de las manifestaciones por el algoritmo `S-H-ESD`

Detección de anomalías en manifestaciones

Estudio 6.3.11: Influencia del tamaño de ventana en la detección de anomalías

Para los datos de cada punto, ya sea de pasos o simultaneos, se han estudiado distintos valores de la ventana de agrupamiento para la detección de anomalías. La tabla 6.13 muestra el porcentaje de anomalias (respecto al tamaño total de las serie) encontrado para cada serie.

Tabla 6.13
Porcentaje sobre el tamaño total de la serie que el algoritmo S-H-ESD ha determinado que es anómalo, para cada punto de monitorización, tipo de serie y tamaño de ventana.

Serie	Paso					Simultáneos				
	60	30	15	5	1	60	30	15	5	1
A	0.36 %	0.55 %	0.44 %	0.47 %	0.69 %	0.29 %	0.53 %	0.49 %	0.42 %	0.28 %
B	0.10 %	0.20 %	0.23 %	0.40 %	0.78 %	0.11 %	0.21 %	0.21 %	0.35 %	0.51 %
C	0.27 %	0.27 %	0.28 %	0.30 %	0.44 %	0.27 %	0.27 %	0.28 %	0.31 %	0.40 %

En todos los casos, al hacer más pequeña la ventana de agrupamiento, se detectan mayor cantidad de anomalías. Además, debido a que a hacer más pequeña la ventana de agrupamiento, hace que haya más cantidad de intervalos, la lista de intervalos anómalos se incremente de forma significativa al hacer más pequeña la ventana. Este comportamiento se recoge en la Figura 6.103.

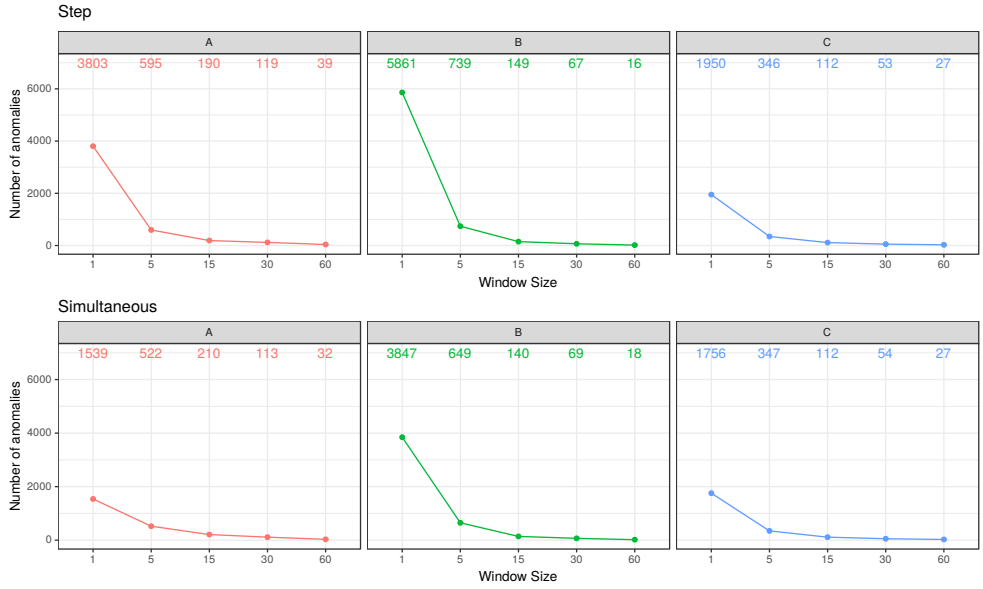


Figura 6.103
Número de anomalías detectadas para cada punto de monitorización, tipo de agrupamiento y tamaño de ventana. Se evidencia que tamaños muy pequeños de ventanas implican detectar un mayor número absoluto de anomalías en el mismo periodo de tiempo.

Esto es debido, principalmente, porque al considerar ventanas de agrupamiento muy grandes las anomalías pueden diluirse de forma que pasen inadvertidas. De igual manera, al hacer las ventanas de tiempo muy pequeñas, se pueden considerar anomalías fluctuaciones irregulares del tránsito, como por ejemplo, personas que caminen juntas en grupo, o momentos en los

que el tránsito se haya detenido momentáneamente, como un paso de cebras. Por ejemplo, la Figura 6.104 muestra como las anomalías se diluyen en los tamaños de ventana más grandes, y son más susceptibles en las ventanas más estrechas.

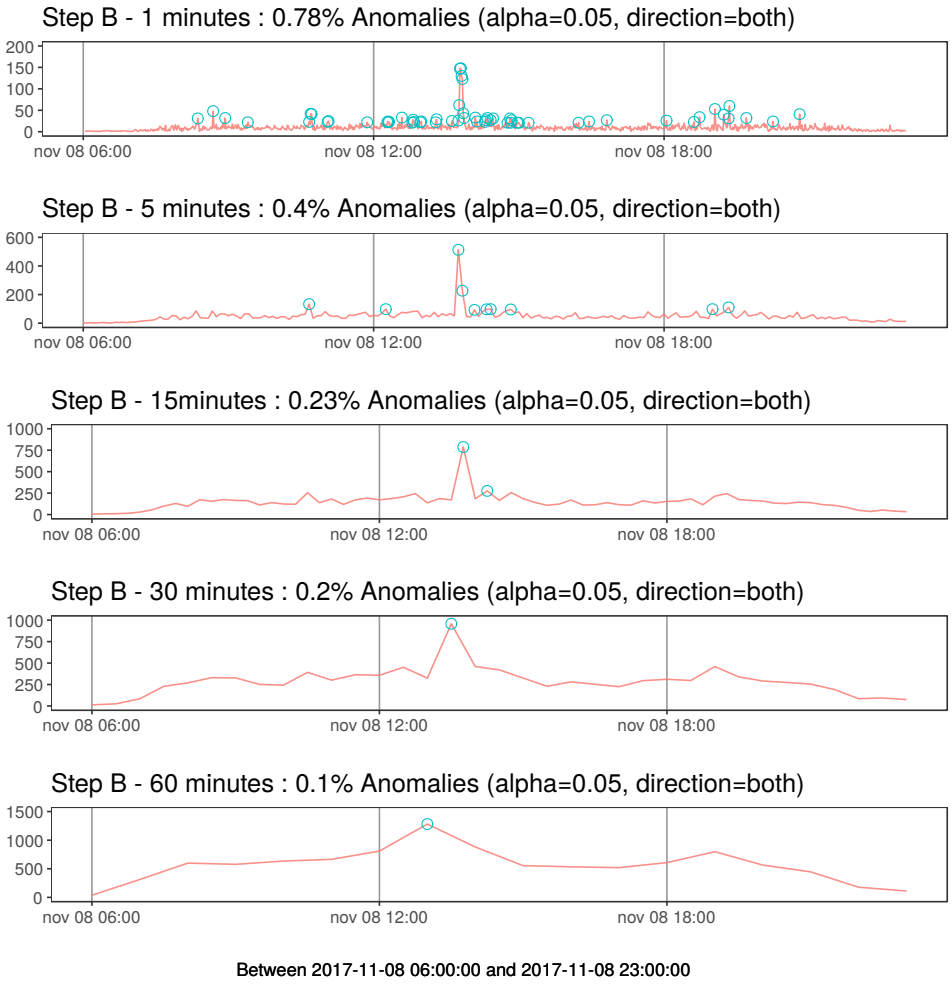


Figura 6.104
Ejemplo de la influencia en el tamaño de la ventana de agrupamiento en la detección de anomalías y la cantidad de anomalías detectadas.

Detección de anomalías en manifestaciones

Estudio 6.3.12: Selección de anomalías críticas

Disponer un criterio o una métrica por la que organizar las anomalías en base a su importancia se vuelve imprescindible si se desea emplear tamaños de ventana de agrupamiento muy estrechos. En este trabajo, se considera el nivel de la anomalía como el ratio entre el valor real y el valor esperado en el intervalo. De esta forma, anomalías en donde la realidad haya sido varios órdenes de magnitud por encima de lo esperado serán más críticas que otras donde, aunque anómalo, sea menos crítico. Se puede cuestionar si este criterio no haría que las anomalías con magnitudes más bajos fuesen prioritarias, por ejemplo en el tránsito a altas horas de la madrugada, sin

embargo, una de las principales ventajas del algoritmo S-H-ESD es que está pensado para no considerar ese tipo de fluctuaciones como anomalías.

La figura 6.105 recoge los días que han sido seleccionados como días con anomalías críticas. Para cada serie, un máximo de 10 días han sido seleccionados.

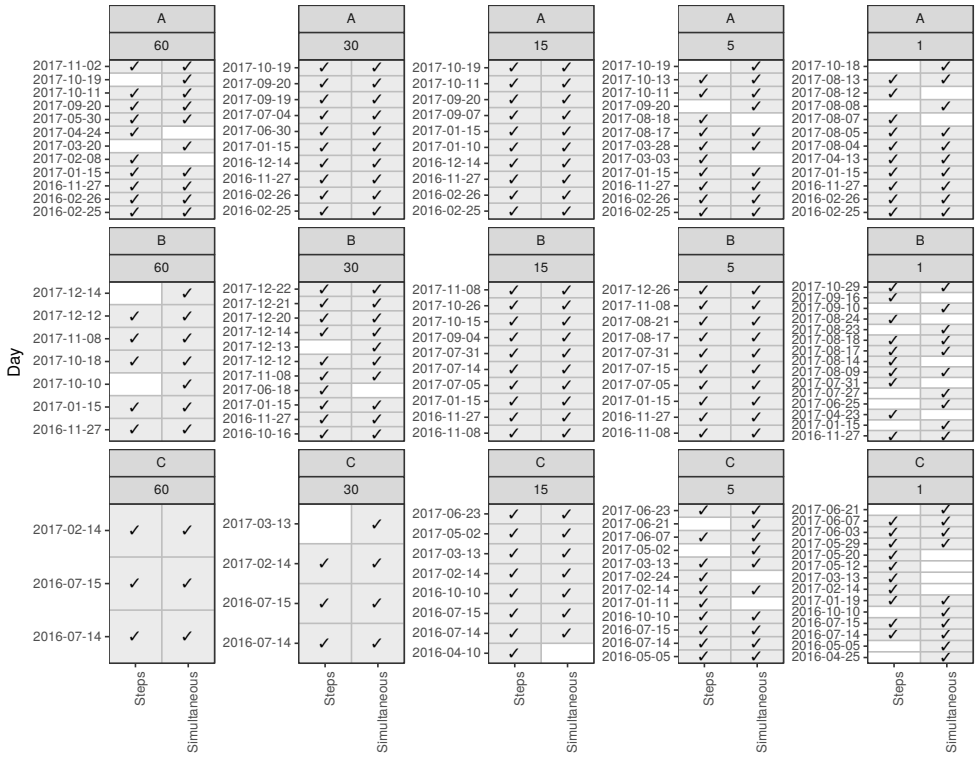


Figura 6.105
Días críticos con anomalías detectadas por el nivel de cada punto, tipo y tamaño de ventana.

Como se ha indicado, este criterio de selección puede experimentar problemas cuando las magnitudes se presenten en distintas escalas o en periodos de bajo tránsito[48]. Este efecto se incrementa si la ventana de muestreseo se hace muy pequeña, llegando a ser inferior que el tiempo medio de estancia, debido a que en el cómputo de pasos se impone que un dispositivo sólo sea considerado en en primer intervalo de su ocurrencia (Sección 5.1.3). La Figura 6.106 representa tanto los pasos como los dispositivos simultáneos con una ventana de muestreo de 1 minuto. A media mañana, la estancia de los dispositivos es superior al minuto, por lo que aunque el número de pasos de dispositivos por minuto no exceda los 75 dispositivos, simultáneamente se ha llegado a monitorizar más de 100 dispositivos simultáneos por minuto en las inmediaciones del nodo.

48 ↑Por ejemplo, una anomalía donde se espera 1 dispositivo pero fuesen detectados 10 dispositivo.

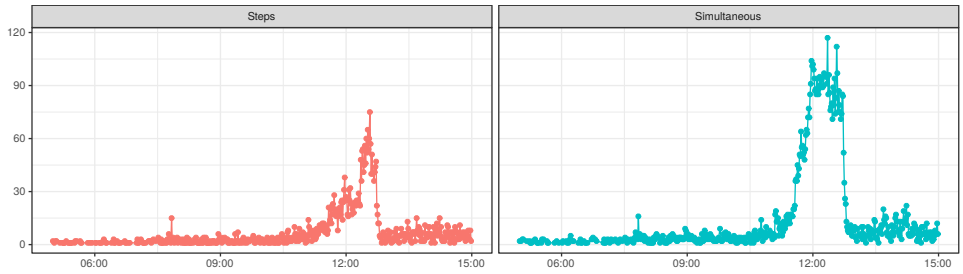


Figura 6.106
Dispositivos simultáneos frente a pasos con ventanas estrechas (1 minuto).

Resulta evidente, que cuanto más pequeña se haga la ventana, menos pasos serán otorgados a cada muestreo. Así por ejemplo, emplear una ventana de muestreo ridículamente pequeña (p.e de solo un milisegundo) tornaría la serie temporal en una secuencia de ceros, con ocurrencias de unos en momentos puntuales y por tanto, anómalos.

Es por ello que la ventana de muestreo debe situarse en valores cercanos a la duración de la estancia media del dispositivo. Sin embargo, a pesar de ello, el algoritmo `S-H-ESD` es capaz de lidiar con este tipo de falsos valores anómalos como se aprecia en la Figura 6.105.

Sin embargo, las series obtenidas por medio de la agrupación de los pasos resultan menos consistentes en la elección de los días anómalos como se puede ver en la Figura 6.102.

De igual manera que con el número de anomalías, emplear un tamaño de ventana muy estrecho o muy ancho influye en los intervalos detectados como anómalos. En este caso, haciéndo que días distintos sean seleccionados. Esto por ejemplo se evidencia en la Figura 6.107 donde se muestran los días con anomalías críticas para el punto A (Steps) en función del tamaño de ventana.

Al emplear un tamaño de ventana muy estrecho, un mismo día muestra una gran cantidad de anomalías, al contrario que al usar un tamaño muy ancho, en el que las anomalías suelen ser puntuales. De igual manera, emplear la serie de simultáneos reduce el número de anomalías en las series con ventana muy estreñas, lo cual se puede comprar comparando las gráficas 6.107 y 6.108 en el número de anomalías dentro de un mismo día para las ventanas estrechas.

Las series de pasos con ventanas estrechas presentan muchas anomalías puntuales y de corta duración, que pueden ser explicadas por el tránsito de grupos de personas caminando juntos. La mayoría de los días detectados con anomalías empleando ventanas de 1 minuto son días de verano, donde es habitual que grandes grupos de visitantes extranjeros caminen juntos siguiendo a un guía. Estudiar las características comunes de las anomalías, puede ser de vital importancia para prevenirlas.

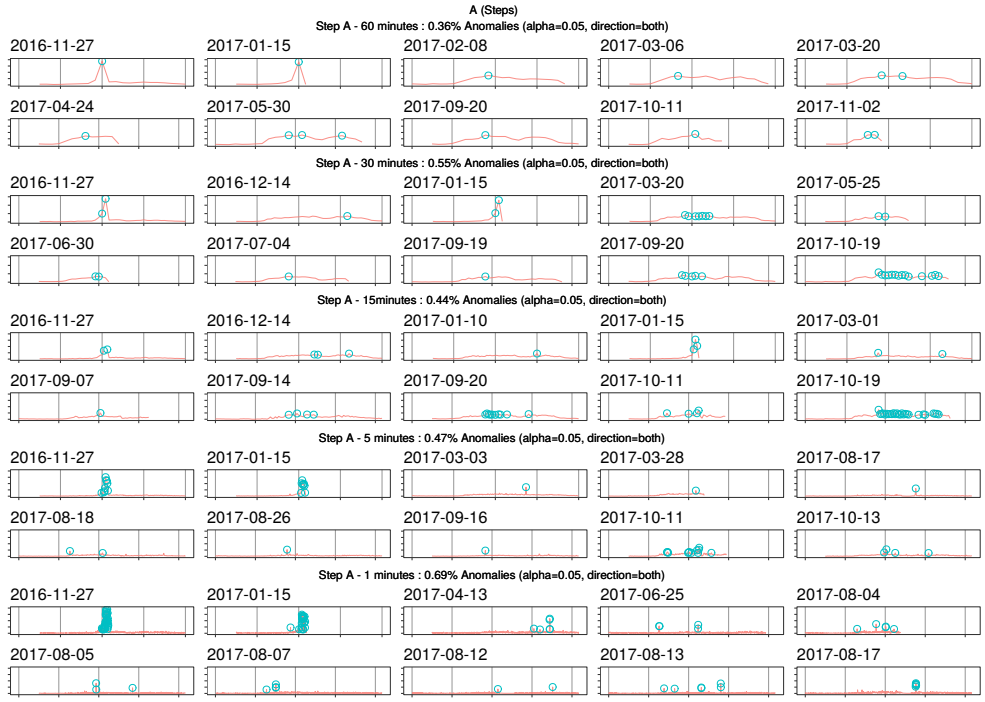


Figura 6.107
Series temporales de los días seleccionados para el punto A y serie por pasos.

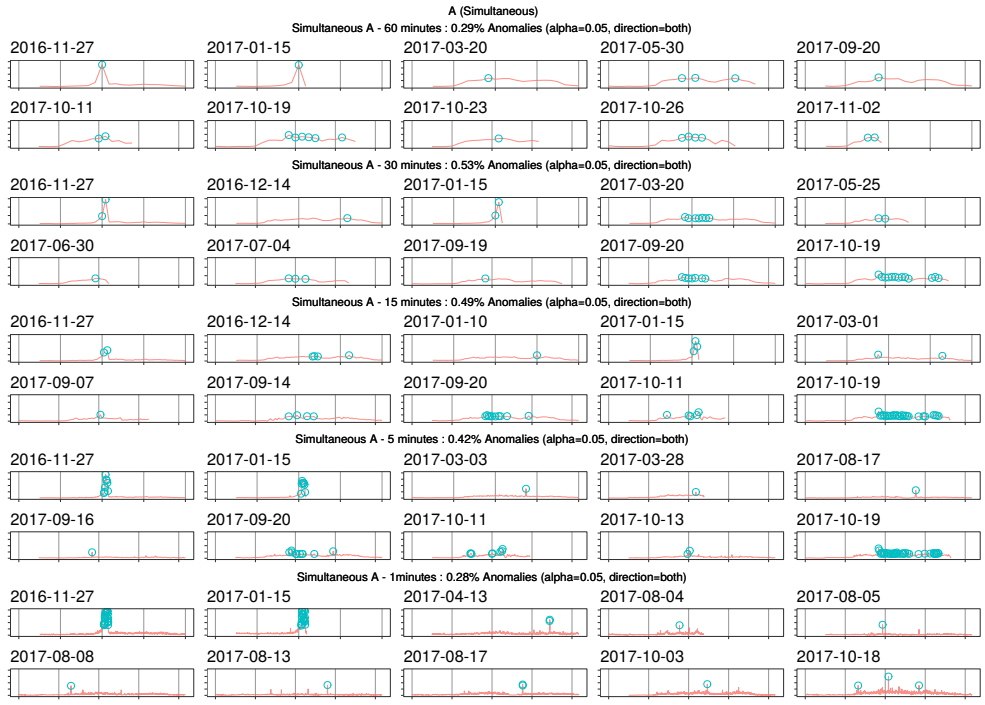


Figura 6.108
eries temporales de los días seleccionados para el punto A y serie por simultáneos

Detección de anomalías en manifestaciones

Estudio 6.3.13: Aprendizaje sobre las anomalías detectadas

Una vez se tienen determinandas las anomalías detectadas, se puede extraer conocimiento sobre las características comunes que tienen, como por ejemplo, en el estudio anterior que se encontró que las anomalías en muestreo de un minuto pertenecen a días de verano.

Como se presentó en la Sección 5.11.3, las marcas temporales pueden ser descompuestas en componentes con el fin de extraer los patrones habituales. La Figura 6.109 presenta el número de anomalías detectadas en cada mes para las tres series empleadas para los tres casos más extremos de tamaños de muestreo.

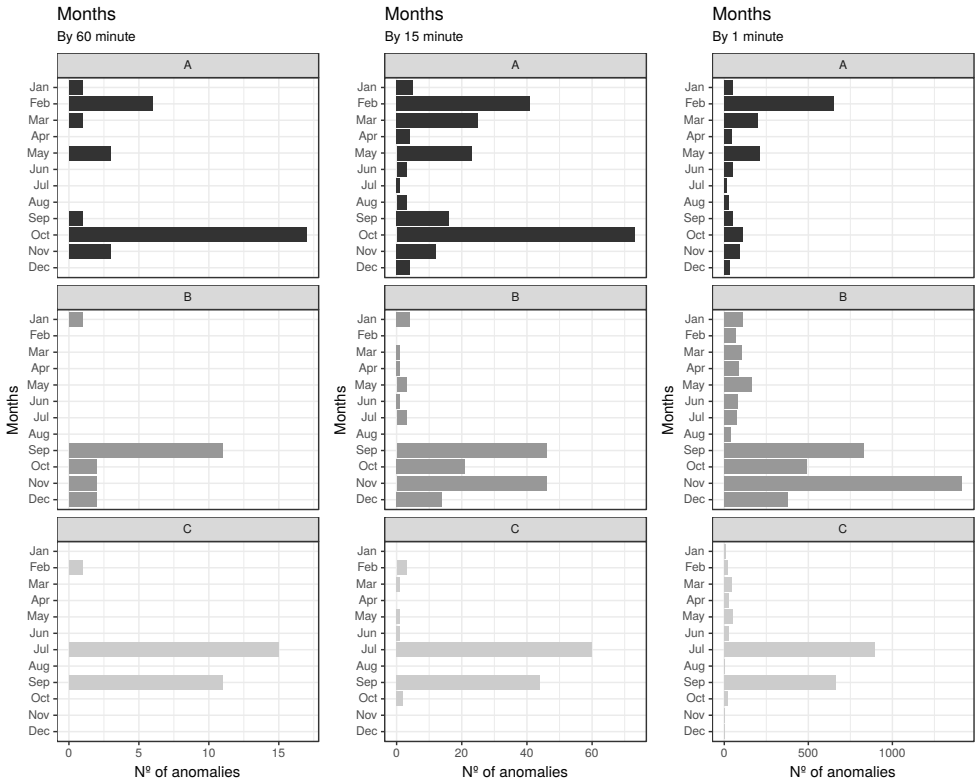


Figura 6.109
Número de anomalías detectadsa por mes por el sistema para los tres puntos y tres tamaños de ventana extremos. Se observa, como para la mayoría de las series las anomalías se focalizan en unos meses concretos.

La misma descomposición puede ser aplicada para estudiar el día de la semana, la hora del día o incluso el minuto dentro de las horas, como se presenta en la Figura 6.110.

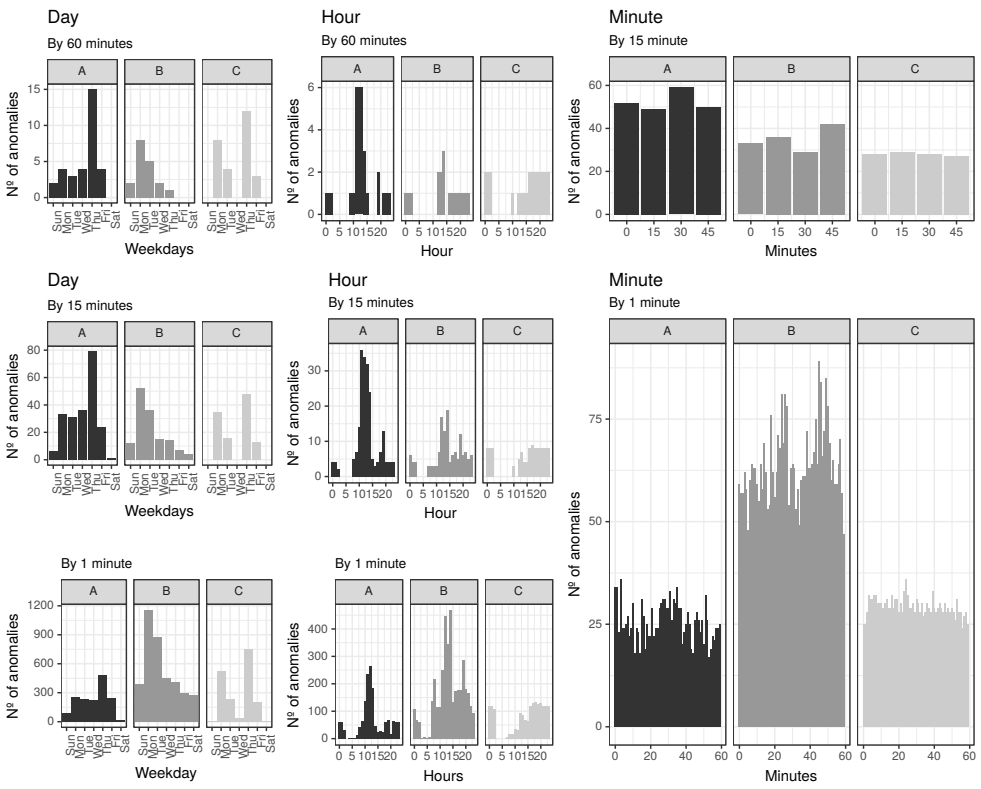


Figura 6.110
Número de anomalías detectadas por el sistema por día de la semana, hora del día o minuto dentro de la hora, para las tres series y para diferentes tamaños de ventanas extremos. La mayoría de las anomalías se focalizan en patrones concretos

Por ejemplo, resulta interesante observar como la mayoría de las anomalías detectadas en el punto A tienen lugar los viernes por la mañana. O que en el punto B, las anomalías se presentan los lunes por la noche. En el punto C, en cambio, no existe ningún patrón en cuanto a la hora de dlía, pero la mayoría de las anomalías tienen lugar los lunes y los jueves.

Una de las ventajas del sistema y la fuente de datos, es que la aplicación de cada técnica de agrupamiento, extracción de información o conocimiento, puede proveer de una nueva fuente de datos, que sea a su vez empleada para nuevas técnicas y procedimientos.

Detección de anomalías en manifestaciones

Estudio 6.3.14: Anomalías en la velocidad

El algoritmo de detección de anomalías `S-H-ESD` puede ser empleado en cualquiera de las magnitudes de tráfico que provee el sistema de monitorización propuesto (Secciones 3.2 y 5.1). En este estudio, se emplea el algoritmo `S-H-ESD` para encontrar anomalías en el tiempo de desplazamiento promedio entre los puntos A y B. La Figura 6.111 presenta las series temporales del tiempo de recorrido entre A y B en ambas direcciones, así como las anomalías detectadas.

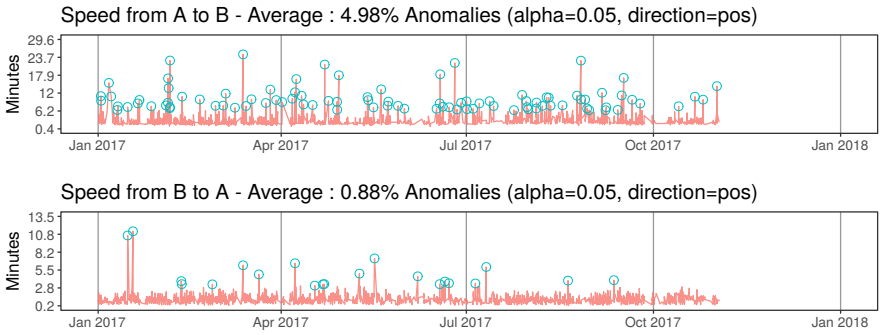


Figura 6.111
Anomalías detectadas el tiempo de desplazamiento promedio entre los puntos A y B.

Es interesante como la cantidad de anomalías detectadas de $A \rightarrow B$ es mucho mayor que en la dirección contraria, con más de cinco veces más anomalías detectadas. Además, las variaciones en la velocidad entre $A \rightarrow B$ respecto a $B \rightarrow A$ son significativas. Se nota la influencia del desnivel de la calle[49] que hace que al ir cuesta abajo ($B \rightarrow A$) se tarde menos que yendo cuesta arriba ($A \rightarrow B$). De igual manera se puede estudiar las anomalías para localizar patrones, como se presenta en la Figura 6.112

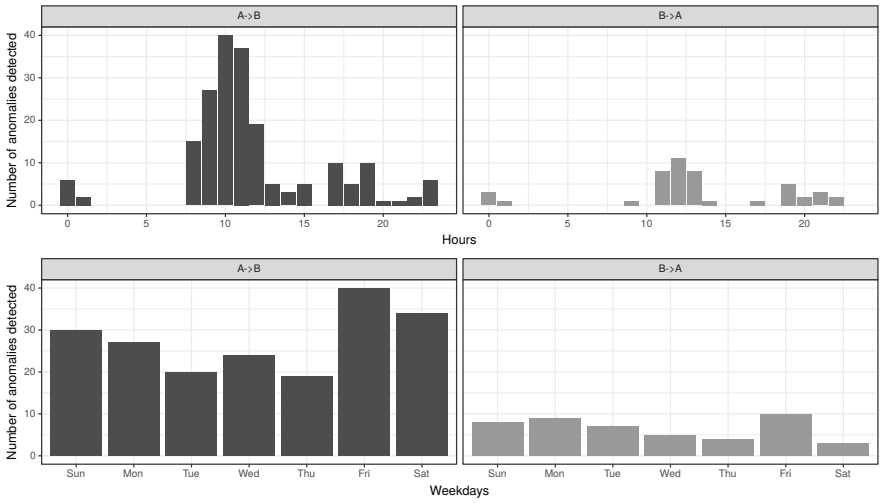


Figura 6.112
Patrones de las anomalías detectadas por el sistema en el tiempo de desplazamiento entre los nodos A and B, estudiando la hora y el día de la semana.

Las anomalías de los desplazamientos entre $A \rightarrow B$ se producen principalmente los viernes por las mañanas en ambas direcciones. A pesar de existir mayor cantidad de anomalías en la direción de $A \rightarrow B$, el sistema arroja que la mayoría de los viandantes detectados realizan el desplazamiento en la dirección $B \rightarrow A$.

49 ↑De unos 8 metros de desnivel en el total de la calle.

Conclusiones

Se ha observado un incremento estadísticamente significativo del número de dispositivos inteligentes durante los instantes previos a manifestaciones en las inmediaciones de nodos emplazados cerca de la zona de comienzo de la protesta.

Se ha cuantificado un incrementado de hasta 10 veces más dispositivos en los momentos de las manifestaciones. Supuesto que la proporción entre dispositivos inteligentes y personas no variase, supondría hasta 10 veces más personas en tránsito por las calles.

Debido a que el sistema es capaz es capaz de cuantificar el número de dispositivos detectados de forma simultánea en cualquier instante de tiempo, es posible reconstruir cualquier momento de la manifestación.

Como los dispositivos son identificados de forma unívoca, es posible determinar que dispositivos son habituales para un marco de tiempo y no contabilizarlos por la manifestación. De igual manera, es posible determinar que dispositivos han sido detectados en las diversas manifestaciones.

Se estudian las series temporales generadas, concluyendo que son viables para el empleo del algoritmo de detección de anomalías presentado en la Sección 5.12.5.

Se estudia la influencia de la ventana de muestreo en los resultados del algoritmo de detección de anomalías, así como la diferencia entre la contabilización de pasos y de dispositivos simultáneos.

Se observa que el empleo de ventanas de muestreo muy pequeñas perjudica al rendimiento del algoritmo, pues fluctuaciones normales del tráfico en anomalías. El empleo de ventanas de muestreo demasiado grandes diluye las anomalías puntuales, haciendo que pasen inadvertidas en la serie. Se concluye que la ventana de muestreo debe aproximarse al tiempo de media de estancia de los dispositivos en las inmediaciones del nodo.

Se propone una métrica para determinar el nivel de las anomalías, con el fin de disponer de un mecanismo para seleccionar aquellas más críticas.

Una vez seleccionadas las anomalías más criticas, se propone extraer patrones de ocurrencia de dichas anomalías, influidas por factores temporales periódicos, ampliamente estudiados en la tesis: la influencia del día de la semana y la hora del día. De esta forma, es posible determinar que momentos futuros serán más propicios de sufrir una anomalía.

6.3.5 *Aprendizaje de patrones de estancias en edificios: ETSIIT*

Para estudiar el potencial de la fuente de datos se emplea el sistema para monitorizar una de las entradas de la ETSIIT[50] de la Universidad de Granada. Dicha escuela tiene matriculados 2119 alumnos y en ella trabajan unas 250 personas[51]. Las clases tiene una hora de duración, y comienzan a las 8:30.

Un nodo *Mobywit* fue emplazado en la entrada principal del edificio (Figura 6.113) durante un periodo de un mes comprendido entre el 3 de Marzo y el 3 de Abril de 2018, incluyendo tanto periodos lectivos como no lectivos, el nodo *Mobywit* capturó 1 884 219.13*MB* de tramas WiFi. A partir del procesamiento en tiempo real, el nodo estableció los *pasos* de cada dispositivo por las inmediaciones del nodo.

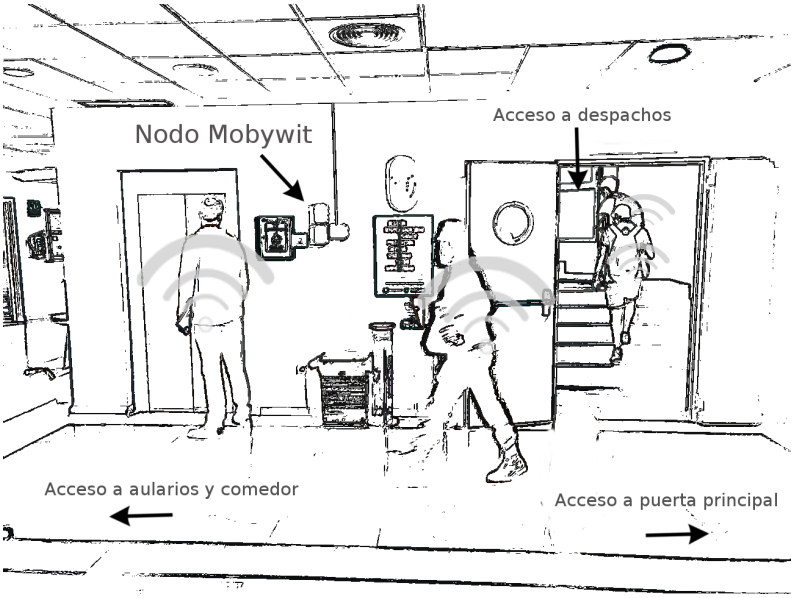


Figura 6.113
Emplazamiento del Nodo Mobywit en la Escuela. En la parte derecha de la fotografía, fuera de plano, se encuentra la puerta principal.La puerta en plano de la derecha

En base a la información en *pasos* obtenida por el procesamiento de dichas tramas, se propone realizar estudios y aplicar técnicas de aprendizaje máquina para obtener información sobre los hábitos y patrones de comportamiento de los visitantes.

Los resultados muestran que las estancias reincidentes a lo largo de varios días se pueden agrupar empleando algoritmos de *machine learning* o aprendizaje máquina para extraer patrones habituales de comportamiento, patrones que sirven para comprender mejor la ocupación y saturación del edificio en los distintos momentos del día.

50 ↑Escuela Técnica Superior de Ingenierías en Informática y Telecomunicaciones
51 ↑Según la memoria académica del curso 2017/2018 https://secretariageneral.ugr.es/pages/memorias/academica/20162017/docencia/centros/_doc/24/!/download

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.15: Cuantificación de visitantes diarios

Los pasos obtenidos se resumen y contabilizan en base a un intervalo de muestreo. La Figura 6.114 muestra el resultado de este resumen para tamaños de muestreo de un día y una hora. De esta forma, se puede observar como el periodo vacacional de Semana Santa la afluencia de gente se reduce drásticamente, de igual manera que durante los fines de semana.

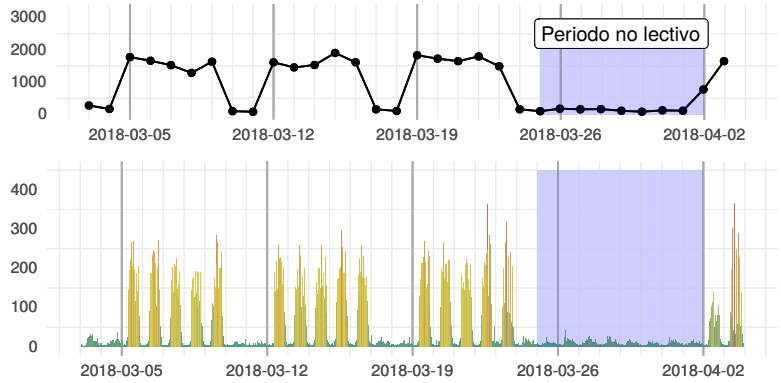


Figura 6.114
Conjunto de datos obtenidos durante un mes de monitorización. La gráfica superior presenta el número de dispositivos/personas detectadas cada día y la inferior por cada hora.

Esta misma información puede ser mostrada a nivel de día, como se presenta en la Figura 6.115.

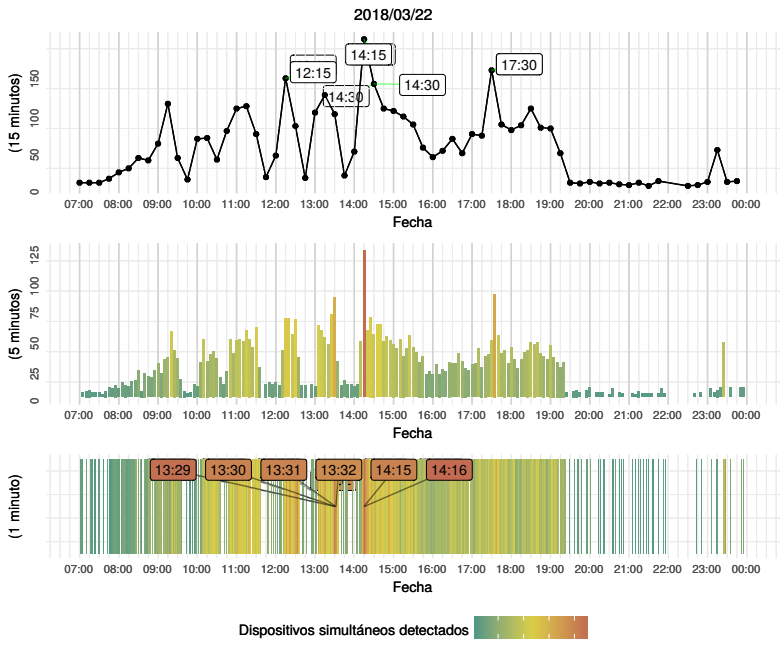


Figura 6.115
Vista detallada de un día monitorizado, empleando tres tamaños de muestreo.

Se espera que cada visitante al edificio registre al menos una entrada y una salida del mismo, siendo registradas ambas en el caso de realizarlas por la puerta monitorizada. Esto permite calcular para cada día el tiempo medio de visitas. La Figura 6.116 registra la duración de todas las visitas en las que se ha registrado tanto la entrada como la salida.

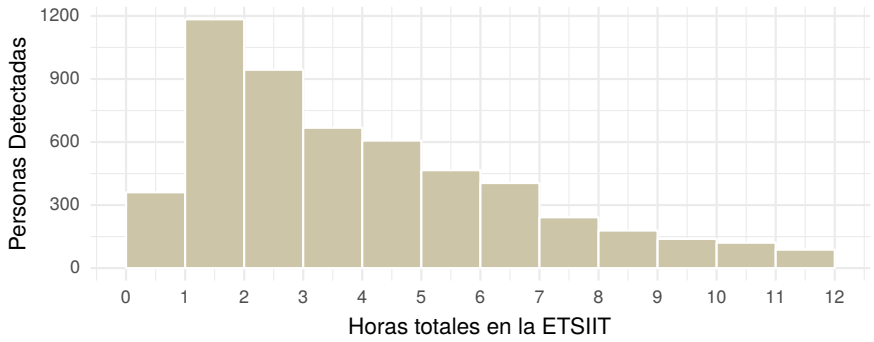


Figura 6.116
Duración de las visitas registradas. La mayoría de los visitantes están entre una y dos horas en la escuela.

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.16: Clasificación de los visitantes diarios en función de su hora de entrada y salida al edificio

Conociendo tanto la hora de entrada como la hora de salida de cada vistante, se ejecuta el algoritmos de agrupamiento *FarthestFirst* [121]. Para facilitar la interpretación de los resultados del algoritmos de agrupamiento lidiando con fechas y horas, las horas de entrada y salida se resumen en 5 categorías divisorias obtenidas como *boundaires* o fronteras del agrupador. Estas divisiones se recogen en la Tabla 6.14.

Tabla 6.14
Interpretación de los divisores de horas del agrupador.

Madrugada	$\in (0,1,2,3,4,5,6]$
Mañana	$\in (7,8,9,10,11]$
Mediodía	$\in (12,13,14,15]$
Tarde	$\in (16,17,18,19,20]$
Noche	$\in (21,22,23]$

El algoritmo de agrupamiento establece cinco clases de visitantes (notadas mediante □, ○, △, ◇ y ⊠), más una clase auxiliar para las visitas no clasificables o sin un patrón común extraído notado mediante +. Adicionalmente, en este artículo se notará con el símbolo × aquellas visitas registradas mediante un único paso cercano al sensor. En la Tabla 6.15 se recogen la asignación de cada divisor de hora a cada clase calculada mediante del algoritmo de agrupamiento.

Tabla 6.15
Clases resultantes del agrupamiento de categorías

CLASE		ENTRADA	SALIDA
□	g1	Mañana	Mediodía
o	g2	Mediodía	Mediodía
△	g3	Tarde	Tarde
◇	g4	Mediodía	Tarde
⊠	g5	Mañana	Tarde
+	-	*	*

Para la visita de cada día de cada persona distinta, se dispone de la hora de entrada, la hora de salida y la categoría que le ha asignada mediante el algoritmo de agrupamiento. En la Figura 6.120 se puede ver como se relacionan las tres variables entre ellas. Las regiones con patrones comunes de entrada y salida han sido asignadas a la misma clase por el algoritmo de agrupamiento. Debido al libre albedrío de las personas, hay visitas que no corresponden a un patrón compartido. Dichas visitas son notadas mediante + y descartadas por el agrupador, en lugar de intentar a ajustarlas de forma artificial a algún patrón o clase ya existente.

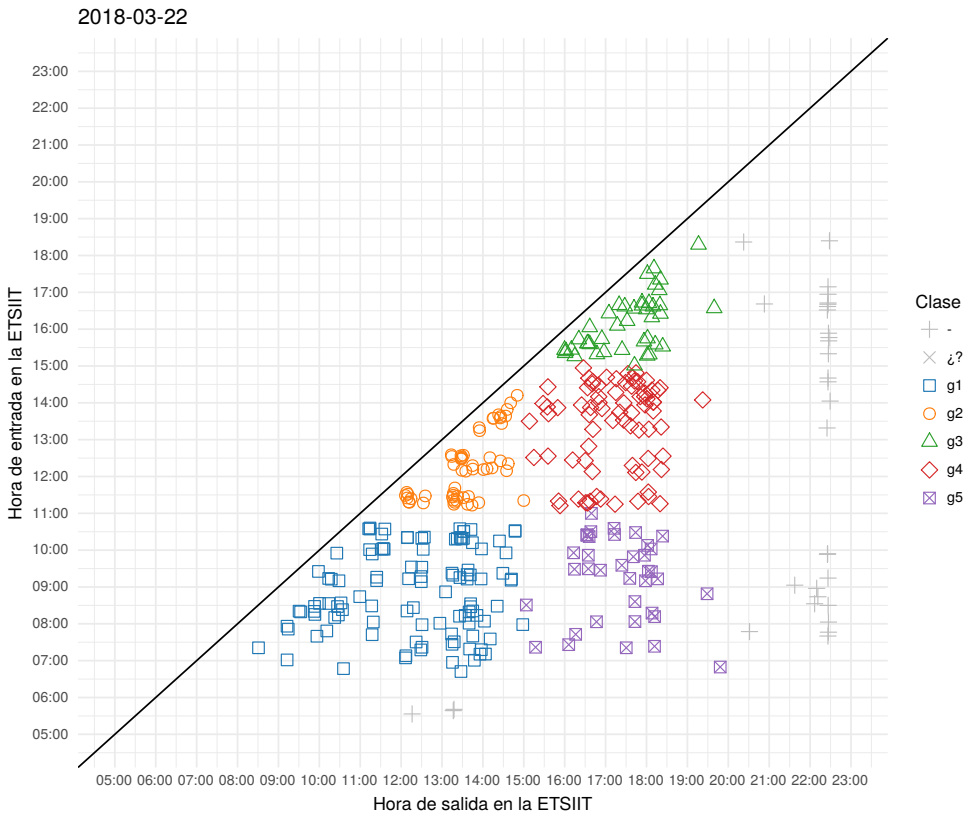


Figura 6.117
Relación entre la hora de entrada, la hora de salida y la clase asignada por el agrupador para todas las visitas de un día determinado.

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.17: Estudio de reincidencia de los visitantes

Es esperable que los visitantes acudan de forma regular a la escuela, ya sea porque estudien o trabajen en ella. En la Figura 6.118 se puede ver el número de visitantes que han sido detectados varios días distintos o de forma reincidente. Esta regularidad individual, supone a su vez un patrón en si mismo.

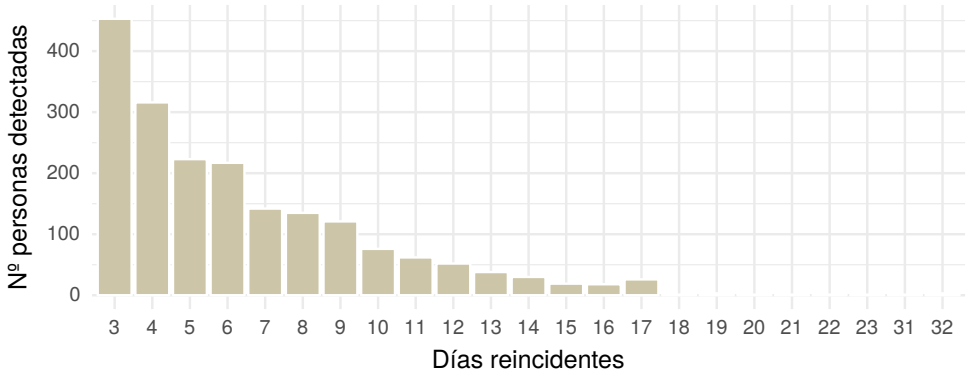


Figura 6.118
Representación del numero de visitantes reincidentes.

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.18: Patrón de comportamiento habitual de los visitantes

El algoritmo de agrupamiento se aplica a todas las visitas de todos los dispositivos detectados a lo largo de todo el periodo. La figura 6.119 recoge la relación entre la hora de entrada, la hora de salida y la categoría asignada a la visita para todos los días que han sido monitorizados por el nodo.

Dada la capacidad del sistema *Mobywit* para ser capaz de reconocer de forma reincidente al mismo dispositivo y por tanto a la misma persona, se pueden extraer patrones en los hábitos de una misma persona a lo largo de varios días. La Figura 6.120 muestra para un subconjunto de personas reincidentes más de 10 veces, la clase asignada a cada una de sus diferentes visitas en diferentes días. En dicha gráfica se puede como las personas se ajustan en su mayoría a un patrón común de comportamiento a lo largo del tiempo.

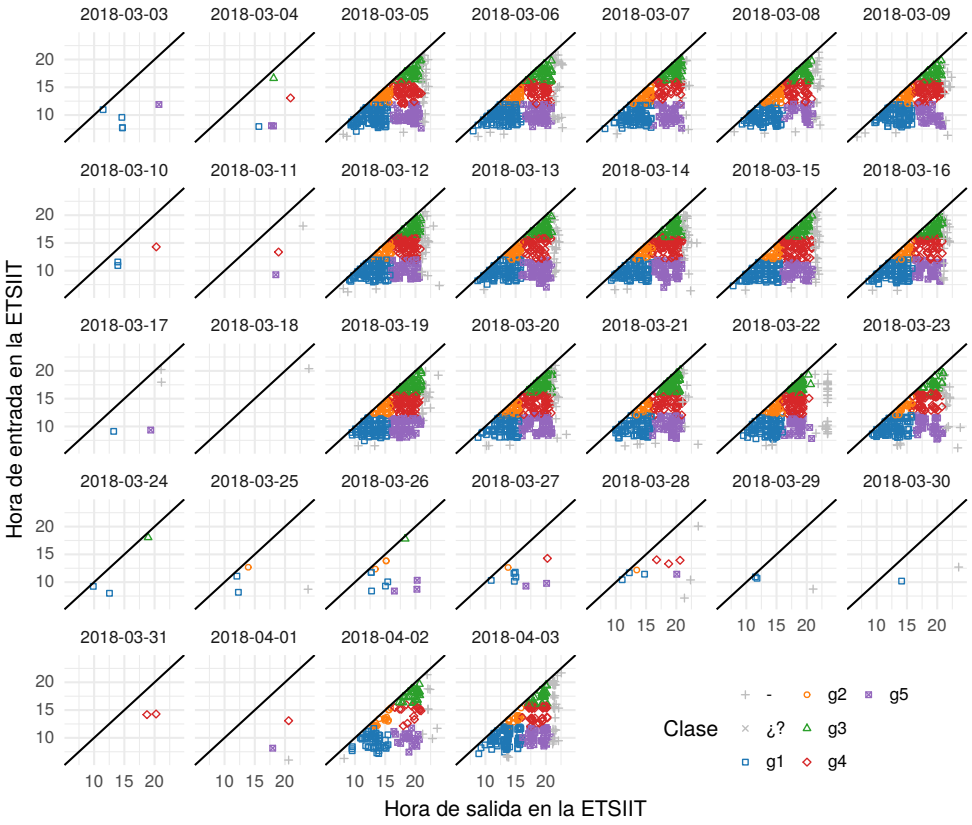


Figura 6.119
Relación entre la hora de entrada, la hora de salida y la clase asignada por el agrupador para todos las visitas del conjunto de datos.

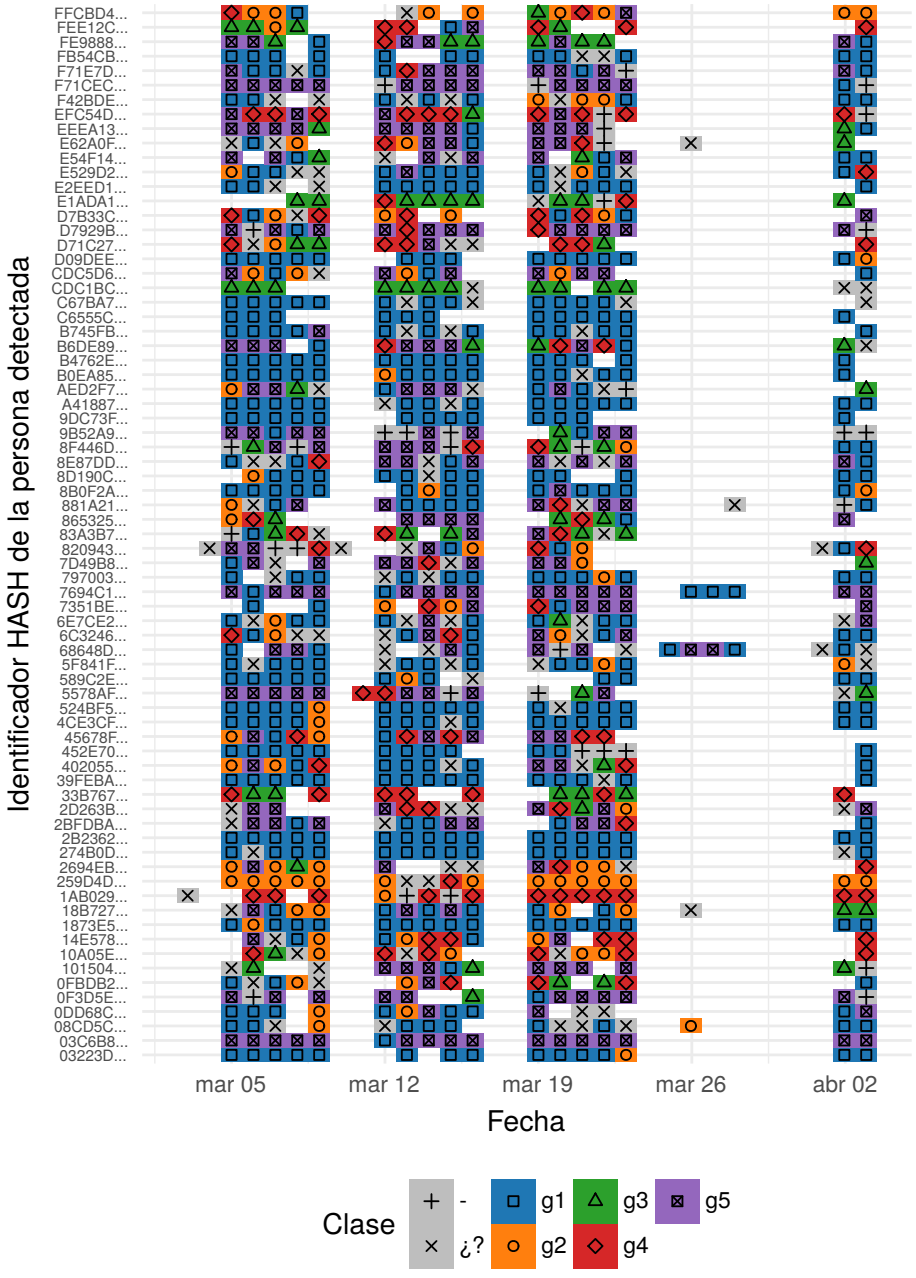


Figura 6.120
Representación de la clase asignada en diferentes visitas para un subconjunto de visitantes los reincidentes más de diez veces. Se observa como las visitas sucesivas de la mayoría de los visitantes son de la misma clase.

No todas las visitas de una misma persona tiene porque ser clasificadas con la misma categoría, sin embargo si puede ser definida una categoría predominante para cada persona de la cual se extraiga un patrón de comportamiento habitual, aunque debido a circunstancias externas ese patrón no sea perfecto. En la Tabla 6.16 se presenta la clase predominante de todos los visitantes ($n \geq 1$) como aquellos reincidentes al menos 10 veces ($n \geq 10$).

Tabla 6.16
Porcentaje de personas para cada clase predominante.

Clase		Predominancia	
	Nombre	$n \geq 1$	$n \geq 10$
□	g1	28.5 %	43.8 %
○	g2	17.9 %	4.1 %
△	g3	18.1 %	4.1 %
◇	g4	11.3 %	10.9 %
⊠	g5	8.1 %	31.5 %
+	-	15.8 %	5.4

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.19: Reconstrucción aproximada del horario habitual individual

Estos patrones de comportamiento pueden ser estudiados a lo largo del tiempo para ver la influencia de factores periódicos, como los días de la semana, los periodos de exámenes o los cuatrimestres en los que se divide el curso. Por ejemplo la figura 6.121 presenta los horarios habituales para cada persona detectada a lo largo de las distintas semana.

Se observa que la clase □ asignada a las personas que entran por la mañana y salen al medio día es la que ha sido asignada a un mayor porcentaje de gente, indicando que la mayoría de las personas que visitan la escuela lo hacen solo en horario de mañana. Además este patrón se muestra tanto en aquellas visitas puntuales, y se acentúa en las visitas reincidentes. Para los reincidentes, la segunda clase con mayor porcentaje (⊠) es aquella asignada a los visitantes que llegan por la mañana y se marchan por la tarde, constituyendo aproximadamente un tercio de los visitantes reincidentes. Un 5 % de las visitas, se produce únicamente al mediodía (○) posiblemente por gente que únicamente emplea los servicios de comedores del edificio. Los visitantes que llegan al mediodía y se van por la tarde (◇) suponen un 10 % de las visitas. Por último, sólo un 5 % de los visitantes realizan su visita únicamente durante horario de tarde (△).

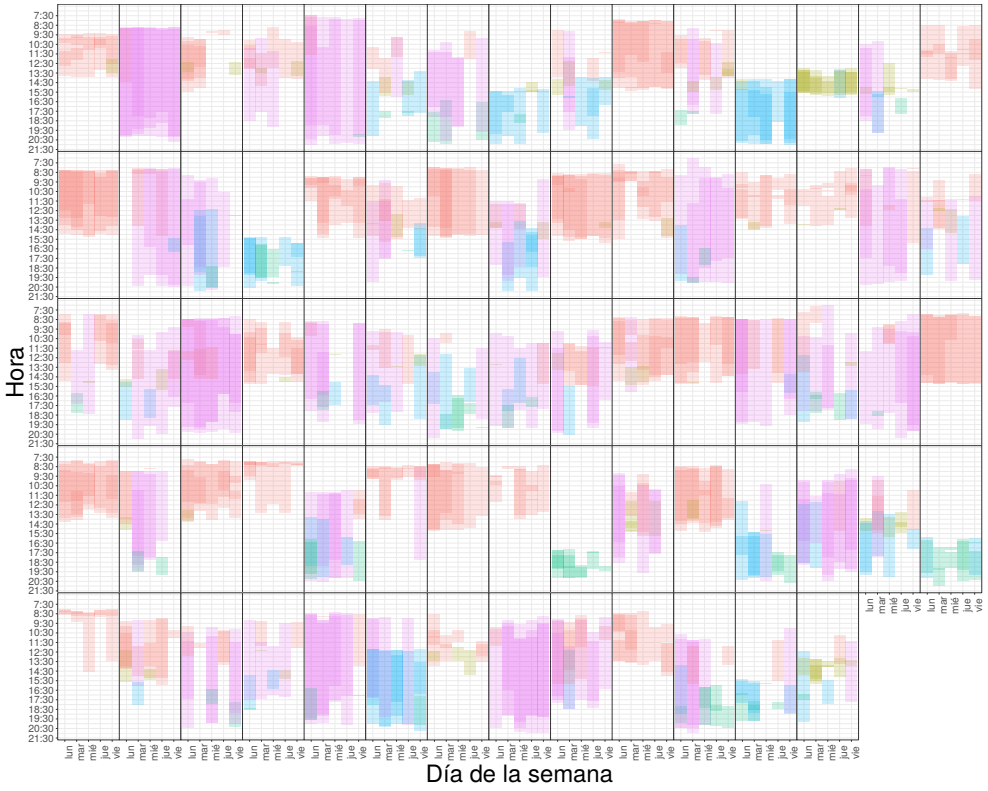


Figura 6.121
Horarios semanales de las 73 personas detectadas reincidentes más representativas. El color indica la clase en la que la que cada estancia ha sido clasificada. Cada visita se superpone con una capa de transparencia del 25 %. La codificación de colores es la empleada en la Tabla 6.16.

Aprendizaje de patrones de estancias en edificios: ETSIIT

Estudio 6.3.20: Determinación del mejor momento para eventos

Esta información puede ser útil para decidir a que hora emplazar eventos maximizando la disponibilidad de los alumnos, escogiendo aquellas horas en las que habitualmente estén habitualmente a punto de irse de la facultad. Empleando la información del agrupador, se puede elegir intervalos de tiempos libres que beneficien a un tipo de visitante concreto, como se muestra en la Figura 6.122.

En cuanto a la disponibilidad de horario, en términos generales la mayor disponibilidad se presenta los jueves a partir de las 18:30. Para los distintos turnos, la asistencia por la mañana presentará mayor disponibilidad los martes de hasta las 11:30, en los turnos de tarde los jueves a partir 18:30. Estos rangos de horas y días son por tanto el momento idóneo para emplazar cualquier evento que no interrumpa la docencia habitual, obtenida en base a los patrones de comportamiento habituales de los alumnos reincidentes. O en caso de querer atraer a toda la audiencia posible, elegir dos periodos de tiempo que permitan la afluencia de ambos tipos de visitantes disjuntos, repitiendo el evento en los dos intervalos de tiempo obtenidos.

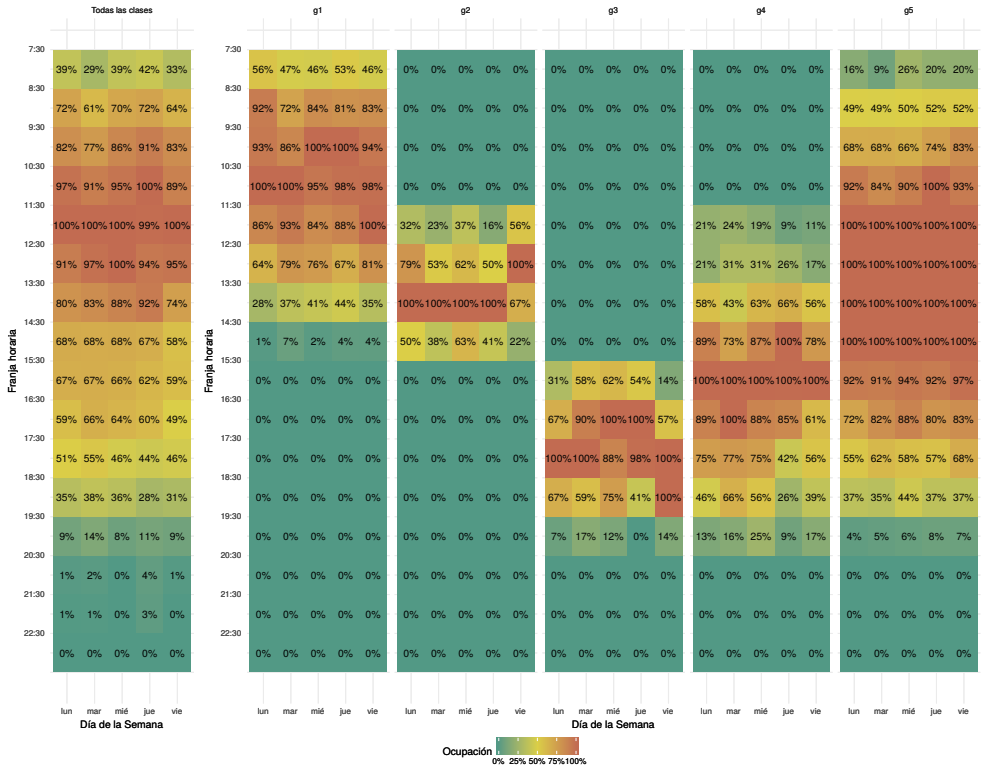


Figura 6.122
Disponibilidad de horarios para la realización de eventos en base a la asistencia a la escuela, tanto genérica como en base los distintos grupos de usuarios detectados. La ocupación se ha normalizado para cada día a una escala entre el 0 y el 100 % del máximo diario.

Conclusiones

Si bien la información obtenida no es más que una muestra de la viabilidad de la captación de comunicaciones inalámbricas WiFi para la monitorización y extración de patrones de comportamientos de las estancias de las personas en edificios, puede resultar una herramienta muy útil para la gestión eficiente de los recursos de dicho edificio. Así como para estudiar el impacto de nuevas medidas en los patrones de comportamiento de los visitantes.

A modo de ejemplo, en este artículo se han determinado los periodos de mayor disponibilidad de los alumnos para la posible asistencia a actividades no regladas, como charlas o conferencias, en base a los patrones de comportamiento extraídos de las visitas reincidentes.

La potencialidad de la aplicación de técnicas de inteligencia computacional y análisis de patrones a los datos obtenidos mediante la captación de comunicaciones inalámbricas permite dotar de fuentes de información baratas, fiables y anónimas a las ciudades inteligentes. Fuentes de datos que permitirán optimizar los recursos ajustándolos a los patrones de comportamiento aprendidos de las personas que los emplean.

La aplicabilidad del sistema *Mobywit* en multitud de escenarios de escenarios, supone también una futura línea de trabajo. El sistema puede ser empleado en cualquier punto geográfico donde se concentren masas de personas, y extraer información sobre el comportamiento de las mismas. Por ejemplo, ofreciendo tiempos de espera hasta ser atendidos en tareas administrativas, monitorizando los recintos habilitados para fumadores o el impacto de la calidad del menú del día en la afluencia de gente en los servicios de comedores.

7

CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO

Jamás fui tan consciente de lo lejos que estaba de mi meta,
que cuando me encontraba tan cerca de ella.

— Vincent - (GATTACA 1997)

En este capítulo se recogen las conclusiones finales de esta tesis, así como los retos y puntos de interés futuro a los que el sistema de monitorización empleando la fuente de datos propuesta se encuentra.

Índice del capítulo

7.1 CONCLUSIONES SOBRE LA VIABILIDAD DE LA CAPTACIÓN DE COMUNICACIONES INALÁMBRICAS

En el Capítulo 4 centrado en estudiar la viabilidad y legalidad de la captación de comunicaciones inalámbricas se concluye:

CONCLUSIÓN 1 Los dispositivos empleando Bluetooth BR/EDR pueden ser detectados empleado los mecanísmos de búsqueda nativos del protocolo.

CONCLUSIÓN 2 Los dispositivos empleando Bluetooth LE *NO* pueden ser detectados empleado los mecanísmos de búsqueda nativos del protocolo, al basarse en la suscripción de anuncios periódicos.

CONCLUSIÓN 3 Los dispositivos empleando WiFi pueden ser detectados en las inmediaciones mediante la captura en modo monitor de las tramas de red que dichos dispositivos emplean para detectar BSSs que provean de puntos de acceso de red.

CONCLUSIÓN 4 Los dispositivos empleando empleando protocolos de corto alcance como NFC y RFID no son considerados para la fuente de datos propuesta debido a su corto alcance.

CONCLUSIÓN 5 No existen impedimentos legales para la captación de las comunicaciones inalámbricas WiFi y Bluetooth empleadas en la fuente de datos propuesto.

CONCLUSIÓN 6 Al emplear bandas de frecuencia licendidas, no es posible emplear las comunicaciones telefónicas inalámbricas.

7.2 CONCLUSIONES SOBRE LA VIABILIDAD DEL DESARROLLO DE UN PROTOTIPO FUNCIONAL

A lo largo del extenso Capítulo 5 se ha presentado como se ha abordado el diseño e implementación de un prototipo de sistema de monitorización basado en la fuente de datos propuesto. Se concluye sobre dicho capítulo:

CONCLUSIÓN 7 Es posible el empleo de un hardware de bajo coste, tanto en el aspecto económico como energético, para la construcción de un nodo de monitorización que emplee captaciones de comunicaciones inalámbricas como fuente de datos.

CONCLUSIÓN 8 El software de monitorización desarrollado resulta eficiente, robusto y autonómo, permitiendo su funcionamiento sin requerir intervención directa por largos periodos de tiempo.

CONCLUSIÓN 9 El almacenamiento y procesamiento de los datos generados por los nodos de monitorización resultan escalables y eficientes permitiendo obtener resultados cercanos al tiempo real incluso en problemas computacionalmente complejos como la reconstrucción de rutas.

7.3 HIPÓTESIS I: SOBRE LA CAPTACIÓN, RECONOCIMIENTO Y MONITORIZACIÓN DE LAS COMUNICACIONES INALÁMBRICAS DE LOS DISPOSITIVOS INTELIGENTES

Sobre la captación de comunicaciones Bluetooth:

CONCLUSIÓN 10 Se concluye que los dispositivos detectados son dispositivos multimedia, de manos libres y smartphones. Solo un 3.8 % de los dispositivos detectados no pertenece a estas categorías.

CONCLUSIÓN 11 Los dispositivos que se encuentran en un modo que permite su descubrimiento, son detectados en tiempos inferiores a un $1ms$. Los dispositivos manos libres se encuentran siempre en dicho modo.

CONCLUSIÓN 12 El sistema de monitorización desarrollado tiene ventanas de detección de 10 segundos separadas por 0.5. Debido a la clase de tarjeta de red, su rango efectivo y la velocidad esperable de los dispositivos, se concluye que no habrá dispositivos detectables que no sean detectados por esa separación.

Sobre la captación de comunicaciones WiFi:

CONCLUSIÓN 13 El tiempo de procesamiento de las tramas capturadas es inferior al $1ms$, permitiendo al sistema de monitorización diseñado procesar hasta 3000 tramas por segundo y mantener monitorizados a más de 1 500 dispositivos de forma simultánea.

CONCLUSIÓN 14 Los dispositivos inteligentes estudiados estando en movimiento, realizan la búsqueda de redes WiFi en tiempos inferiores a cada 5 segundos. Los sistemas ahorro de energia se activan cuando el dispositivo se encuentra en reposo, espaciando la búsqueda de forma incremental al tiempo. Los dispositivos inteligentes realizan esta búsqueda incluso deshabilitando la interfaz o al entrar en modo avión.

CONCLUSIÓN 15 Factores como el rebote de las tramas o la aleatorización de la dirección de búsqueda no han supuesto complicaciones al sistema de monitorización propuesto.

CONCLUSIÓN 16 Los dispositivos detectados en producción pertenecen en su mayoría a fabricantes de dispositivos inteligentes, siendo solo un 0.1 % de los dispositivos detectados pertenecientes a fabricantes de dispositivos de infraestructura.

Comunes a ambas tecnologías:

CONCLUSIÓN 17 Los resultados de emplazar varios nodos de monitorización en una misma ubicación arroja resultados coherentes y consistentes entre ellos.

CONCLUSIÓN 18 El correcto emplazamiento de los nodos de monitorización resulta vital para una correcta monitorización de los dispositivos inteligentes en movimiento.

7.4 HIPÓTESIS II: SOBRE EL ESTUDIO DE LA MOVILIDAD DE LOS DISPOSITIVOS INTELIGENTES Y LA ADECUACIÓN AL MOVIMIENTO DE PERSONAS Y VEHÍCULOS

Sobre la monitorización de vehículos:

CONCLUSIÓN 19 Según la encuesta realizada a más de 600 personas, un 50 % de los conductores disponen de un dispositivo manos libres en su vehículo habitual. Solo un 10 % de los conductores habituales con dispositivo manos libres no lo usa diariamente.

CONCLUSIÓN 20 Tanto en el caso de la magnitud del tráfico como de los indicadores de congestión se encuentra un causalidad de Granger entre el número de dispositivos Bluetooth y el número de vehículos. Esto implica que aquellos factores que hacen variar al número de vehículos en las vías, afectan en igual medida al número de dispositivos Bluetooth. Esto se aplica a otros indicadores como la predilección giro o los desplazamientos realizados.

CONCLUSIÓN 21 Es posible inferir el número de vehículos mediante el número de dispositivos Bluetooth si se entrena con datos de aforadores reales. En caso contrario, el sistema de monitorización prové una fuente de datos no exhaustiva que permite determinar el estado del tráfico en base a las variaciones respecto al histórico.

CONCLUSIÓN 22 Debido a que el sistema es capaz de identificar detecciones sucesivas en distintos nodos sensores a lo largo del tiempo, es posible reconstruir las rutas seguidas por los dispositivos y extraer información sobre el tiempo total y velocidad promedia de la ruta.

CONCLUSIÓN 23 Existen fuertes indicios en las mediciones actuales de que la fuente de datos propuesta podrá ser empleada para la aproximación del número de ocupantes por vehículo en el futuro, cuando la renovación de la flota de vehículos haga que el ratio de vehículos y dispositivos Bluetooth detectados sea equiparable.

CONCLUSIÓN 24 La detección y monitorización de dispositivos Bluetooth prové de un mecanismo no exhaustivo para la monitorización y trazabilidad del movimiento de vehículos. Ninguno de los estudios y experimentos realizados arroja algún indicio de lo contrario.

Sobre la monitorización de personas:

CONCLUSIÓN 25 Según la encuesta realizada, solamente un 30 % de los encuestados se preocupa de apagar la interfaz WiFi de su `smartphone` cuando no hace uso de ella. Apagar la interfaz solo aumenta el tiempo entre búsquedas, por lo que aún así siguen siendo detectables.

CONCLUSIÓN 26 Al no existir métodos alternativos con los que realizar comparaciones exactas solo es posible acotar el número de dispositivos WiFi detectados con los aforos y público potencial de los eventos monitorizados, siendo en todos los escenarios estudiados resultados coherentes y acotados.

CONCLUSIÓN 27 Debido a que el sistema es capaz de identificar detecciones sucesivas en distintos nodos sensores, es posible reconstruir la estancia de un dispositivos en varias salas a lo largo del periodo de tiempo de su estancia total. Con esta información es posible realizar análisis sobre los comportamientos promedio de los visistantes.

CONCLUSIÓN 28 El empleazamiento del nodo de monitorización tanto en interiores como exteriores ha resultado inadvertido en todos aquellos escenarios donde ha sido implantado, sin producirse ningún robo o manipuación por actos vandálicos.

CONCLUSIÓN 29 La dección de dispositivos WiFi se ha visto influenciada por factores externos que producen un aumento del número de personas transitando por las calles, con variaciones dentro de lo esperable y acotadas por lo coherente.

CONCLUSIÓN 30 La detección y monitorización de dispositivos WiFi prové de un de un mecanismo no exhaustivo para la monitorización y trazabilidad del movimiento de personas. Ninguno de los estudios y experimentos realizados arroja algún indicio de lo contrario.

7.5 HIPÓTESIS III: SOBRE LA APLICABILIDAD DE LA INFORMACIÓN GENERADA AL ÁMBITO DE A UNA SMARTCITY

CONCLUSIÓN 31 Los resultados analíticos obtenidos por la fuente de datos propuesta puede servir para la aplicación de métodos de `machine learning` que permitan extraer patrones, predecir valores y detectar anomálias en las magnitudes del tráfico tanto de personas como de vehículos, así como cualquier otra métrica derivada del comportamiento individual, colectivo o reiterativo.

CONCLUSIÓN 32 El producto final de dichos métodos puede servir para mejorar la gestión de los recursos de las ciudades inteligentes, al ofrecer información y conocimiento sobre las causas y patrones que modelizan el comportamiento de los ciudadanos.

CONCLUSIÓN 33 El sistema puede ser empleado también para determinar el impacto real de las acciones emprendidas por las administraciones, para influir el tránsito de los ciudadanos, determinando si las acciones emprendidas han tenido algún impacto en la ciudadanía.

7.6 LÍNEA DE TRABAJO FUTURO

Es deseable la aplicación e implantación del sistema de monitorización basado en la fuente de datos propuesta en la mayor cantidad de escenarios, para ampliar la red de sensores desplegada. Monitorizar cuantos más puntos de las ciudades para el tráfico de vehículos o mayor cantidad de edificios para las estancias de personas, permite aumentar el tamaño de los conjuntos de datos disponibles para el entrenamiento de los métodos de aprendizaje empleados.

El despliegue de los nodos de monitorización en nuevos lugares siempre abre las puertas al descubrimiento de nuevos retos y limitaciones que afrontar en cuanto al diseño, eficiencia y recursos de los mismos.

Las mejoras en hardware, el abaratamiento de los componentes electrónicos y la gran cantidad de dispositivos sensores que se pueden implantar el nodo de monitorización permiten ampliar las magnitudes que oferta el nodo, siendo deseable incluir el futuro factores meteorológicos y ambientales que emplear como variables exógenas complementarias a la información ya adquirida.

Los cambios en la legislación y configuraciones poder defecto de los dispositivos inteligentes requiren de especial atención constante, para adaptar la fuente de datos y el sistema de monitorización al marco legal y al comportamiento de los nuevos dispositivos inteligentes emergentes en el mercado.

Es necesario entender que esta tesis doctoral, la fuente de datos propuesta y el sistema de monitorización empleado no suponen la culminación de una investigación puntual, si no que intentan sentar las bases de una línea de investigación con proyección y aplicación en el futuro, con deseo de servidumbre a las ciudades inteligentes y sus administraciones para ofrecer una parte ínfima de información que permita mejorar los recursos de la ciudad y suponer un impacto positivo en la vida de los ciudadanos.

Parte IV

Anexos y Bibliografía

A

ANEXOS

Puede que no tenga el talento,
¡pero tengo la determinación!

— Eren Jegger

Índice del anexo

A.1 BLUETOOTH - TABLAS DE MINOR DEVICE CLASSES

Tabla A.1
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Computer Major Class
Fuente: Bluetooth Baseband [252].

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	0	Uncategorized
0	0	0	0	0	1	Desktop Workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clamshell)
0	0	0	1	0	1	Palm-size PC/PDA
0	0	0	1	1	0	Wearable computer (watch size)
0	0	0	1	1	1	Tablet
X	X	X	X	X	X	–Reserved–

Tabla A.2
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Phone Major Class

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	0	Uncategorized
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smartphone
0	0	0	1	0	0	Wired moderm or voice gateway
0	0	0	1	0	1	Commom ISDN access
X	X	X	X	X	X	–Reserved–

Tabla A.3
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Network Major Class

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	0	Fully available
0	0	1	0	0	0	1% to 17% utilized
0	1	0	0	0	0	17% to 33% utilized
0	1	1	0	0	0	33% to 50% utilized
1	0	0	0	0	0	50% to 67% utilized
1	0	1	0	0	0	67% to 83% utilized
1	1	0	0	0	0	83% to 99% utilized
1	1	1	0	0	0	No service available
X	X	X	X	X	X	–Reserved–

Tabla A.4
Subtipo del los dispositivos Bluetooth según su `Minor Device Class` para `Audio/Video Major Class`

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	0	Uncategorized
0	0	0	0	0	1	Wearable Headset Device
0	0	0	0	1	0	Hands-free Device
0	0	0	0	1	1	–Reserved–
0	0	0	1	0	0	Microphone
0	0	0	1	0	1	Loudspeaker
0	0	0	1	1	0	Headphones
0	0	0	1	1	1	Portable Audio
0	0	1	0	0	0	Car audio
0	0	1	0	0	1	Set-top box
0	0	1	0	1	0	HiFi Audio Device
0	0	1	0	1	1	VCR
0	0	1	1	0	0	Video Camera
0	0	1	1	0	1	Camcorder
0	0	1	1	1	0	Video Monitor
0	0	1	1	1	1	Video Display and Loudspeaker
0	1	0	0	0	0	Video Conferencing
0	1	0	0	0	1	–Reserved–
0	1	0	0	1	0	Gaming/Toy
X	X	X	X	X	X	–Reserved–

Tabla A.5
Subtipo del los dispositivos Bluetooth según su `Minor Device Class` para `Peripheral Major Class (I)`

7	6	MINOR DEVICE CLASS
0	0	Not Keyboard / Not Pointing Device
0	1	Keyboard
1	0	Pointing device
1	1	Combo keyboard/pointing device

Tabla A.6
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Peripheral Major Class (II)

5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	Uncategorized device
0	0	0	1	Joystick
0	0	1	0	Gamepad
0	0	1	1	Remote control
0	1	0	0	Sensing device
0	1	0	1	Digitizer tablet
0	1	1	0	Card Reader (e.g. SIM Card Reader)
0	1	1	1	Digital Pen
1	0	0	0	Handheld scanner for bar-codes, RFID, etc.
1	0	0	1	Handheld gestural input device
X	X	X	X	–Reserved–

Tabla A.7
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Imaging Major Class (I)

7	6	5	4	MINOR DEVICE CLASS
X	X	X	1	Display
X	X	1	X	Camera
X	1	X	X	Scanner
1	X	X	X	Printer
X	X	X	X	–Reserved–

Tabla A.8
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Imagin Major Class (II)

7	6	MINOR DEVICE CLASS
0	0	Uncategorized
X	x	–Reserved–

Tabla A.9
Subtipo del los dispositivos Bluetooth según su Minor Device Class para Werable Major Class

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	1	Wristwatch
0	0	0	0	1	0	Pager
0	0	0	0	1	1	Jacket
0	0	0	1	0	0	Helmet
0	0	0	1	0	1	Glasses
X	X	X	X	X	X	–Reserved–

Tabla A.10
Subtipo del los dispositivos Bluetooth según su `Minor Device Class` para `Toy Major Class`

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	1	Robot
0	0	0	0	1	0	Vehicle
0	0	0	0	1	1	Doll / Action figure
0	0	0	1	0	0	Controller
0	0	0	1	0	1	Game
X	X	X	X	X	X	–Reserved–

Tabla A.11
Subtipo del los dispositivos Bluetooth según su `Minor Device Class` para `Health Major Class`

7	6	5	4	3	2	MINOR DEVICE CLASS
0	0	0	0	0	0	Undefined
0	0	0	0	0	1	Blood Pressure Monitor
0	0	0	0	1	0	Thermometer
0	0	0	0	1	1	Weighing Scale
0	0	0	1	0	0	Glucose Meter
0	0	0	1	0	1	Pulse Oximeter
0	0	0	1	1	0	Heart/Pulse Rate Monitor
0	0	0	1	1	1	Health Data Display
0	0	1	0	0	0	Step Counter
0	0	1	0	0	1	Body Composition Analyzer
0	0	1	0	1	0	Peak Flow Monitor
0	0	1	0	1	1	Medication Monitor
0	0	1	1	0	0	Knee Prosthesis
0	0	1	1	0	1	Ankle Prosthesis
0	0	1	1	1	0	Generic Health Manager
0	0	1	1	1	1	Personal Mobility Device
X	X	X	X	X	X	–Reserved–

A.2 WIFI - TABLA DE TIPOS DE TRAMAS WIFI

Tramas de administración

Tabla A.12
Tramas de administración
Fuente: 802.11-1999 [253]. Pag.380-384.

Tipo (b3b2)	Descripción Tipo	Subtipo (b7b6b5b4)	Descripción Subtipo
00	Administración	0000	Petición de asociación
00	Administración	0001	Respuesta de asociación
00	Administración	0010	Petición de reasociación
00	Administración	0011	Respuesta de reasociación
00	Administración	0100	Petición de prueba
00	Administración	0101	Respuesta de prueba
00	Administración	1000	Beacon
00	Administración	1001	Mensaje ATIM
00	Administración	1010	Disociación
00	Administración	1011	Autenticación
00	Administración	1100	Anulación de autenticación
00	Administración	1101	Acción (802.11h y QoS)

Tramas de control

Tabla A.13
Tramas de control
Fuente: 802.11-1999 [253]. Pag.380-384.

Tipo (b3b2)	Descripción Tipo	Subtipo (b7b6b5b4)	Descripción Subtipo
01	Control	1000	Petición de acuse de recibo de bloque (QoS)
01	Control	1001	Acuse de recibo de bloque (QoS)
01	Control	1010	Poll-Ahorro de potencia (PS)
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	R (ACK)
01	Control	1110	End-Free-Contention (CF)
01	Control	1111	End-CF+CF-Ack

Tramas de datos

Tabla A.14
Tramas de datos
Fuente: 802.11-1999 [253]. Pag.380-384.

Tipo (b3b2)	Descripción Tipo	Subtipo (b7b6b5b4)	Descripción Subtipo
10	Datos	0000	Datos
10	Datos	0001	Datos + CF-Ack
10	Datos	0010	Datos+Poll-CE
10	Datos	0011	Datos+CD-Ack+Poll-CE
10	Datos	0100	Datos Null (no se transmiten datos))
10	Datos	0101	CF-Ack (no se transmiten datos)
10	Datos	0110	CF-Poll (no se transmiten datos)
10	Datos	0111	CF-Ack-CF-Poll (no se transmiten datos)
10	Datos	1000	Datos QoS
10	Datos	1001	Datos QoS+CF-Ack
10	Datos	1010	Datos QoS+CF-Poll
10	Datos	1011	Datos QoS+CF-Ack+CF-Poll
10	Datos	1100	QoS Null (no se transmiten datos)
10	Datos	1101	QoS CF-Ack (no se transmiten datos)
10	Datos	1110	QoS CF-Foll (no se transmiten datos)
10	Datos	1111	QoS CF-Acl-CF-Poll (no se transmiten datos)

El tipo de trama 11 está reservado.

A.3 LIBRERÍAS EMPLEADAS EN EL SOFTWARE DE MONITORIZACIÓN DEL PROTOTIPO

A continuación se detallan brevemente las librerías empleadas para el desarrollo del **software de monitorización** del prototipo a modo de referencia de las mismas.

JNI Project

`(JNIProjectJava.jar)`
Permite añadir código C++ dentro de las clases `JAVA`. `(EXPERIMENTAL)`
`https://www.fer.unizg.hr/_download/repository/jni.pdf`

Addresing

`(addressing-1.0.jar)`
Usar SOAP
`https://docs.oracle.com/javaee/6/api/javax/xml/ws/soap/Addressing.html`

Bluecove GPL

`(bluecove-gpl-2.1.0-sources.jar)`
Gestiona de conexiones Bluetooth.
`http://bluecove.org/`

Apache Commons Compress

`(commons-compress-1.0.jar)`
Realizar compresiones y descompresiones de archivos
`(https://commons.apache.org/proper/commons-compress/)`

Apache Commons Daemons

`(commons-daemon-1.0.5.jar)`
Implementar un demonio en el sistema
`(https://commons.apache.org/proper/commons-daemon/)`

Apache Commons Discovery

`(commons-discovery.jar)`
Detectar implementaciones de interfaces
`https://commons.apache.org/dormant/commons-discovery/`

Apache Commons Email

`(commons-email.jar)`
Para el envío de correos electrónicos
`https://commons.apache.org/proper/commons-email/`

Apache Commons HttpClient

(commons-httpclient-3.0-rc2.jar)
Proporciona servicios para acceder a recursos vía HTTP
http://hc.apache.org/httpclient-3.x/

Apache Commons IO

(commons-io-1.3.2.jar)
Proporciona funcionalidades avanzadas para el empleo de funcionalidad de entrada/salida en ficheros y flujos
https://commons.apache.org/proper/commons-io/

Apache Commons Lang

(commons-lang-2.4.jar)
Proporciona métodos avanzados para la manipulación de cadenas de texto y su interpretación
(ttps://commons.apache.org/proper/commons-lang/

Apache Commons Logging

(commons-logging.jar)
Proporciona una capa de abstracción a la generación de logs de depuración
(https://commons.apache.org/proper/commons-logging/)

Apache Derby

(derby.jar)
Apache Derby, genera bases de datos dentro del ecosistema JAVA
http://db.apache.org/derby/

Dom4j

(dom4j-1.6.1.jar)
Para parsear, interpretar y manipular ficheros XML
https://dom4j.github.io/

Gson

(gson-1.7.1.jar)
Para parsear, interpretar y manipular ficheros JSON
https://mvnrepository.com/artifact/com.google.code.gson/gson/2.8.1

Jackson Data Processor

(jackson-all-1.9.4.jar)
Para parsear, interpretar y manipular ficheros JSON
https://github.com/FasterXML/jackson-core

JDOM

(jdom-1.1.jar)
Para parsear, interpretar y manipular ficheros XML
http://www.jdom.org/

Apache jersey

(jersey-apache-client-1.8.jar, jersey-client-1.8.jar, jersey-core-1.8.jar, jersey-guice-1.8.jar, jersey-guice-1.8.jar ,jersey-json-1.8.jar, jersey-multipart-1.8.jar)
Para el acceso a un servicio RESTful
https://jersey.github.io/

Apache Log4j

(log4j-1.2.15.jar)
Para la implementación de sistema de log con multiples niveles
https://logging.apache.org/log4j/2.x/)

Mysql Connector

(mysql-connector-java-5.0.8-bin.jar)
Para la conexión con bases de datos MySQL
https://dev.mysql.com/downloads/connector/j/

Jpcap

(net.sourceforge.jpcap-0.01.16.jar)
Para la captura de tramas WiFi
http://jpcap.sourceforge.net/

Opensaml

(opensaml-1.0.1.jar)
Para el intercambio de datos de autenticación y autorización
https://wiki.shibboleth.net/confluence/display/OpenSAML/Home

pi4j

(pi4j-core.jar, pi4j-device-javadoc.jar, pi4j-device.jar, pi4j-gpio-extension.jar, pi4j-service.jar)
Para el acceso pleno a las capacidades de entrada/salida de la raspberry pi
http://pi4j.com/

Saaj

(saaj.jar)
Para el envío de mensajes SAAJ mediante SOAP
https://docs.oracle.com/javaee/5/tutorial/doc/bnbhg.html

XStream

`(xstream-1.3.1.jar)`
Para serializar ficheros XML
`http://x-stream.github.io/`

A.4 ESPECIFICACIÓN DE LA API REST DE COMUNICACIONES

En este Anexo se detallan las funciones desarrolladas en la API REST de comunicaciones entre los nodos de monitorización y el servidor de cómputo presentada en la Sección 5.8.

Función /connection/test

Tabla A.15
Función /connection/test de la API REST.

URI	/connection/test		
Función encargada de comprobar el estado alcanzable del servidor.			
PARÁMETROS	-		
MÉTODO HTTP	GET	AUTENTICACIÓN	NO REQUERIDA
RESPUESTA	200 OK	Servidor se encuentra disponible	
	401 Unauthorized	Aplicación no autorizada	
FORMATO SALIDA	text/plain	Devuelve un código identificable por el servidor	

Función /nodes

Tabla A.16
Función /nodes de la API REST.

URI	/nodes		
Obtiene el listado de identificadores de nodos asociados a las credenciales proporcionadas.			
PARÁMETROS	uid	Identificador de usuario	
	token	Hash de la contraseña asociada	
MÉTODO HTTP	GET	AUTENTICACIÓN	REQUERIDA
RESPUESTA	200 OK	Credenciales correctas. Devuelve información nodos.	
	400 Bad Request	Credenciales erróneas proporcionadas.	
	401 Unauthorized	Aplicación no autorizada	
FORMATO SALIDA	application/json	Información sobre los nodos asociados al usuario	

Funciones /session/start y /session/end

Tabla A.17
Función `/session/start` de la API REST.

URI	`/session/start`		
Indica al servidor que el nodo determinado va a empezar a funcionar. Se habilita la recepción de pasos para ese nodo por parte del usuario indicado			
PARÁMETROS	`idNodo`	Identificador del nodo	
	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
	`version`	Número de versión del software en ejecución	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Se ha podido iniciar la sesión en el servidor	
	`400 Bad Request`	Credenciales erróneas proporcionadas.	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`text/plain`	Mensaje sobre la autorización del servidor	

Tabla A.18
Función `/session/end` de la API REST.

URI	`/session/end`		
Finaliza la sesión del nodo indicado. No se permitirá el envío de pasos de ese nodo hasta vuelva a iniciar sesión nuevamente.			
PARÁMETROS	`idNodo`	Identificador del nodo	
	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`application/json`	Información sobre los nodos asociados al usuario	

Función `/steps`

Enviado en el body de un multipart

Tabla A.19
Función `/steps` de la API REST.

URI	`/steps`		
Notifica al sistema los pasos considerados a ser insertados en al servidor.			
PARÁMETROS	`idNodo`	Identificador del nodo	
	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
	`file`	Fichero de pasos en formato CSV	
MÉTODO HTTP	`POST`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`text/plain`	Estado de la inserción de los pasos al sistema	

Funciones `/version/current` y `/version/app`

Tabla A.20
Función `/version/current` de la API REST.

URI	`/version/current`		
Obtiene el listado de identificadores de nodos asociados a las credenciales proporcionadas.			
PARÁMETROS	`idNodo`	Identificador del nodo	
	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`text/plain`	Información sobre los nodos asociados al usuario	

Tabla A.21
Función `/version/app` de la API REST.

URI	`/version/app`		
Obtiene el listado de identificadores de nodos asociados a las credenciales proporcionadas.			
PARÁMETROS	`idNodo`	Identificador del nodo	
	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`application/java`	Información sobre los nodos asociados al usuario	

Función `/version/script`

Tabla A.22
Función `/version/script` de la API REST.

URI	`/version/script`		
Obtiene el listado de identificadores de nodos asociados a las credenciales proporcionadas.			
PARÁMETROS	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`text/plain`	Información sobre los nodos asociados al usuario	

Función `/status`

Tabla A.23
Función `/status` de la API REST.

URI	`/status`		
Obtiene el listado de identificadores de nodos asociados a las credenciales proporcionadas.			
PARÁMETROS	`uid`	Identificador de usuario	
	`token`	Hash de la contraseña asociada	
MÉTODO HTTP	`GET`	AUTENTICACIÓN	`REQUERIDA`
RESPUESTA	`200 OK`	Servidor se encuentra disponible	
	`401 Unauthorized`	Aplicación no autorizada	
FORMATO SALIDA	`text/plain`	Información sobre los nodos asociados al usuario	

A.5 ESPECIFICACIÓN API REST DEL ALMACENAMIENTO EN LA NUBE

En este anexo se detalla la API REST proporcionada para el servicio `Google Fusion Tables` y los métodos desarrollados sobre ella para trabajar sobre ella de forma más cercana al SQL, desarrollándose una librería que amplia y extiende las funcionalidades básicas proporcionada por Google.

Entre otros métodos, se ha desarrollado un sistema que permite actualizar las tuplas ya insertadas, así como gestionar un sistema de colas que permite realizar de forma simultánea varias inserciones en el motor de base de datos con una única petición REST al servicio.

A.5.1 *Referencia rápida de la API REST*

A continuación se presentan los método disponibles en la API REST.

A.5.1.1 *Consultas sobre los datos: SELECT*

Para realizar consultas sobre los datos, se realiza una petición HTTP GET al servidor de Google empleando la sintaxis recogida en la Figura A.1.

```
1  "https://www.googleapis.com/fusiontables/v1/query?sql="
2  SELECT  <column_spec> {, <column_spec>}*
3
4  FROM <table_id>
5
6  { WHERE <filter_condition> | <spatial_condition> { AND <filter_condition>
   ↪  }* }
7
8  { GROUP BY <column_name> {, <column_name>}* }
9
10 { ORDER BY <column_spec> { ASC | DESC } | <spatial_relationship> }
11
12 { OFFSET <number> }
13
14 { LIMIT <number> }
```

Figura A.1
Google Fusion Tables API: Select

Retorno

Si la consulta SELECT es procesada correctamente, se devuelve un JSON formateando los datos en UTF-8. Si se desea, se puede modificar la petición para recibir los datos en formtao CSV.

A.5.1.2 *Borrado de datos: DELETE*

Para borrar una tupla, se usa la petición HTTP POST presentada en la Figura A.2.

```
1 DELETE FROM <table_id>{ WHERE ROWID = <row_id>}
```

Figura A.2
Google Fusion Tables API: DELETE

A.5.1.2.1 Retorno

Si la petición es procesada con éxito, no se devuelve ningún objeto, pero la petición devuelve un estado 200 OK.

A.5.1.3 *Inserccione sobre los datos: INSERT*

Para insertar una o más tuplas en la tabla, se usa la petición HTTP POST de la Figura A.3:

```
INSERT INTO <table_id> (<column_name> {, <column_name>}*)
  VALUES (<value> {, <value>}*)

{ {;INSERT INTO <table_id> (<column_name> {, <column_name>}*) VALUES
↪  (<value> {, <value>}*)}* ;}
```

Figura A.3
Google Fusion Tables API: INSERT

Se pueden insertar un máximo de 500 tuplas o un tamaño máximo de petición de 1MB.

A.5.1.3.1 Retorno

Si tiene éxito la petición devuelve el `ROWID` asociado a la tupla recién insertada.

A.5.1.4 *Actualizaciones sobre los datos: UPDATE*

Para actualizar una o más columnas de una ÚNICA tupla, se usa la petición HTTP POST de la Figura A.4:

```
UPDATE <table_id>
SET <column_name> = <value> {, <column_name> = <value> }*
WHERE ROWID = <row_id>
```

Figura A.4
Google Fusion Tables API: UPDATE

Retorno

Si la petición es procesada con éxito, no se devuelve ningún objeto, pero la petición devuelve un estado 200 OK.

A.5.2 *Métodos desarrollados para trabajar sobre la API REST*

Para facilitar el tratamiento de los datos, se implementan métodos que permitan de forma nativa trabajar con las operaciones más frecuentes en una base de datos así como otros métodos que permiten suplicar las carencias, como por ejemplo la directiva `ON DUPLICATE KEY UPDATE`.

Salvo las inserciones todas ellas hacen uso del método básico `SQL` que se presenta en la Figura A.5:

```
/** Procesa una petición SQL básica
 * @param query cadena de texto que indica la petición SQL a realizar
 * @return El archivo JSON procesado devuelto por la petición */
private Sqlresponse sql(String query) {
  if (query == "" || query == null) { return null; }
  Sqlresponse res = null;   boolean salir = false;
  while (!salir) {
    try {
      Logger.getGlobal().fine("Procesando:" + query);
      Sql sql = fusiontables.query().sql(query);
      res = sql.execute(); salir = true;
    } catch (Exception ex) {
      if (ex.getMessage().contains("403") || ex.getMessage().contains("Read
      ↪  timed out")) {
        try {
          sleep(_c.getInt("ft.tiempo_espera_error_ms"));
        } catch (InterruptedException ex1) {
        Logger.getGlobal().log(Level.SEVERE, "Error al dormir la hebra de
        ↪  Procesado de subidas a la nube", ex1); }
        salir = false;
        Logger.getGlobal().fine("Envío fallido " + ex.getMessage() + " .
        ↪  Demasiado rápido. Se volverá a intentar el envío");
      } else if (ex.getMessage().contains("503")) {
        //Hemos excedido la cuota
        salir = false;
        Logger.getGlobal().fine("Envío fallido " + ex.getMessage() + " .
        ↪  Cuota excedida. Se volverá a intentar el envío pasados " +
        ↪  _c.getInt("ft.tiempo.esperaSubida.dormir") / 1000 + "
        ↪  segundos.");
        try {
          sleep(_c.getInt("ft.tiempo.esperaSubida.dormir"));
        } catch (InterruptedException ex1) {
          Logger.getGlobal().log(Level.SEVERE, "Error al dormir la hebra de
          ↪  Procesado de subidas a la nube", ex1); }
      } else {
        salir = false;
        Logger.getGlobal().log(Level.SEVERE, "Envío fallido " +
        ↪  ex.getMessage() + "'" + query + "'", ex); }
    }
  }
  Logger.getGlobal().fine("Procesado correctamente.");
  return res;
}
```

Figura A.5
Código: Métodos para Fusion Table: SQL

A.5.2.1 *Métodos de selección: SELECT*

Se han implementado varias operaciones de selección obedeciendo a la necesidad de sobrecargar el método en función de los parámetros pasados. Se presenta en la Figura A.6 los dos métodos de selección:

```
1   /**
2    * Función que realiza una selección de una tabla
3    *
4    * @param tabla Identificador de la tabla que se quiere consultar
5    * @param campos Campos que queremos recuperar
6    * @param condiciones Condiciones que tiene que cumplir las tuplas para
     ↪  que
7    * sean devueltas
8    * @return
9    */
10  public Sqlresponse select(String tabla, String campos, String
     ↪  condiciones) {
11    return sql("SELECT " + campos + " FROM " + tabla + " WHERE " +
       ↪  condiciones);
12  }
13
14  /**
15   * Función que realiza una selección de una tabla con condiciones extras
16   *
17   * @param tabla Identificador de la tabla que se quiere consultar
18   * @param campos Campos que queremos recuperar
19   * @param condiciones Condiciones que tiene que cumplir las tuplas para
     ↪  que
20   * sean devueltas
21   * @return
22   */
23  public Sqlresponse select(String tabla, String campos, String condiciones,
     ↪  String extras) {
24    if (!"".equals(condiciones)) {
25      return sql("SELECT " + campos + " FROM " + tabla + " WHERE " +
         ↪  condiciones + " " + extras);
26    } else {
27      return sql("SELECT " + campos + " FROM " + tabla + " " + extras);
28    }
29  }
```

Figura A.6
Código: Métodos para Fusion Table: SELECT

A.5.2.2 *Métodos de inserción: INSERT*

De igual manera que con la selección, sobrecargamos el método para permitir invocarlo con distintos parámetros. En la Figura A.7 se encuentran los métodos sobrecargados.

```java
/**
 * Sobrecarga del método insert para no indicar que se compruebe que existe
 * el elemento. Por defecto, el elemento a insertar se comprueba si existe
 * en la base de datos de FUSION TABLES antes de insertarlo.
 * @param tabla identificador de la tabla
 * @param campos listado de campos (separads por comas) de la tabla
 * @param valores listado de valores (separados por comas) a insertar
 * @return La respuesta devuelta por el servidor de Fusion Tables.
 */
public Sqlresponse insert(String tabla, List<String> campos, List<String> valores) {
  return insert(tabla, campos, valores, true);
}

/**
 * Sobrecarga del método insert para no indicar que se compruebe que existe
 * el elemento y para un solo elemento. Por defecto, el elemento a insertar
 * se comprueba si existe en la base de datos de FUSION TABLES antes de
 * insertarlo.
 * @param tabla identificador de la tabla
 * @param campos listado de campos (separads por comas) de la tabla
 * @param valores listado de valores (separados por comas) a insertar
 * @return La respuesta devuelta por el servidor de Fusion Tables.
 */
public Sqlresponse insert(String tabla, String campos, String valores) {
  List<String> c = new ArrayList<>();
  List<String> v = new ArrayList<>();
  c.add(campos);
  v.add(valores);
  return insert(tabla, c, v, true);
}

/**
 * Sobrecarga del método insert para un solo elemento.
 * @param tabla identificador de la tabla
 * @param campos listado de campos (separads por comas) de la tabla
 * @param valores listado de valores (separados por comas) a insertar
 * @return La respuesta devuelta por el servidor de Fusion Tables.
 */
public Sqlresponse insert(String tabla, String campos, String valores, boolean check) {
  List<String> c = new ArrayList<>();
  List<String> v = new ArrayList<>();
  c.add(campos);
  v.add(valores);
  return insert(tabla, c, v, check);
}
```

Figura A.7
Código: Métodos para Fusion Table: INSERT Sobrecarga

La inserción implica la operación más compleja realizable en nuestras bases de datos Fusion Tables. Esto es debido a que al carecer de clave primaria establecida por nosotros, podemos correr el riesgo de subir varias veces la misma tupla. Al carecer de un mecanismo como el `ON DUPLICATE KEYS UPDATE` que proporciona SQL, antes de realizar una inserción debemos comprobar que la tupla que deseamos insertar no se encuentra ya insertada. Es por ello que se debe realizar una función de selección con toda la tupla (o parte de ella), y comprobar si existe ya almacenada en nuestro sistema.

Debido a la ineficiencia de manera de proceder, (deben de hacerse el doble comunicaciones para insertar cada tupla), se permite no realizar dicha comprobación. Este comportamiento se parametriza con el valor `check` que se muestra en la Figura A.8.

Sin embargo, este método comprueba todas las columnas de la tupla para ver si la tupla se encuentra ya en la base de datos o no. Es por ello que se realiza el método ampliado (Figura A.9) para permitir decidir el número de columnas que son empleadas en la comprobación de que la tupla ya se encuentra en el sistema.

```java
/**
 * Función que inserta una nueva tupla en la tabla alojada en FusionTables
 *
 * @param tabla identificador de la tabla
 * @param campos listado de campos (separads por comas) de la tabla
 * @param valores listado de valores (separados por comas) a insertar
 * @param check si es necesario realizar comprobación de insercción para en
 * caso de ocurrencia usar UPDATE
 * @return La respuesta devuelta por el servidor de Fusion Tables.
 */
public Sqlresponse insert(String tabla, List<String> campos, List<String>
↪  valores, boolean check) {
  String peticion;
  if (check) {
    peticion = "SELECT ROWID FROM " + tabla + " WHERE ";
    for (int i = 0; i < campos.size(); i++) {
      if (i != 0) { peticion = peticion + " AND "; }
      peticion = peticion + campos.get(i) + "='" + valores.get(i) + "'";
    }
    Sqlresponse s = this.sql(peticion);

    if (s.size() == 2) {
      peticion = "INSERT INTO " + tabla + "(";
      for (int i = 0; i < campos.size(); i++) {
        if (i != 0) {
          peticion = peticion + ",";
        }
        peticion = peticion + campos.get(i);
      }
      peticion = peticion + ") VALUES (";
      for (int i = 0; i < valores.size(); i++) {
        if (i != 0) { peticion = peticion + ","; }
        peticion = peticion + "'" + valores.get(i) + "'";
      }
      peticion = peticion + ");";
      insertCache = insertCache + peticion;
      insertCacheContador++;
    } else {
      return this.update(tabla, campos, valores, s.getRows());
    }
  } else {
    peticion = "INSERT INTO " + tabla + "(";
    for (int i = 0; i < campos.size(); i++) {
      if (i != 0) { peticion = peticion + ",";}
      peticion = peticion + campos.get(i);
    }
    peticion = peticion + ") VALUES (";
    for (int i = 0; i < valores.size(); i++) {
      if (i != 0) { peticion = peticion + ","; }
      peticion = peticion + "'" + valores.get(i) + "'";
    }
    peticion = peticion + ");";
    insertCache = insertCache + peticion;
    insertCacheContador++;
  }
  sync();
  return null;
}
```

Figura A.8
Código: Métodos para Fusion Table: INSERT

```
/** @param para número de campos de la lista valores a utilizar en la
 * comparación en caso de que check sea verdadero */
public Sqlresponse insert(String tabla, List<String> campos, List<String>
↪  valores, boolean check, int para) {
  String peticion;
  if (check) {
    peticion = "SELECT ROWID ";
    for (int i = para; i < campos.size(); i++) {
      peticion = peticion + "," + campos.get(i);
    }
    peticion = peticion + " FROM " + tabla + " WHERE ";
    for (int i = 0; i < para; i++) {
      if (i != 0) { peticion = peticion + " AND "; }
      peticion = peticion + campos.get(i) + "=\'" + valores.get(i) + "\'";
    }
    Sqlresponse s = this.sql(peticion);
    if (s.size() == 2) {
      peticion = "INSERT INTO " + tabla + "(";
      for (int i = 0; i < campos.size(); i++) {
        if (i != 0) { peticion = peticion + ","; }
        peticion = peticion + campos.get(i);
      }
      peticion = peticion + ") VALUES (";
      for (int i = 0; i < valores.size(); i++) {
        if (i != 0) { peticion = peticion + ","; }
        peticion = peticion + "\'" + valores.get(i) + "\'";
      }
      peticion = peticion + ");";
      insertCache = insertCache + peticion; insertCacheContador++;
    } else {
      boolean cambio = false; int j = 1; String depura = "";
      for (int i = para; (i < campos.size() && cambio==false); i++) {
        try{
        if (!(valores.get(i).equals((String) s.getRows().get(0).get(j)))) {
          cambio = true;
          Logger.getGlobal().fine("Son distintos " + depura);
        }}catch(ClassCastException ex){
          Object v = s.getRows().get(0).get(j);
          if (!(valores.get(i).equals(v.toString()))) {
          cambio = true;
        }
        }catch(Exception ex){ cambio = true; }
        j++; //En 0 está ROWID
      }
      if (cambio == true) {
        return this.update(tabla, campos, valores, s.getRows());
      } else { return null; }
    }
  } else {
    peticion = "INSERT INTO " + tabla + "(";
    for (int i = 0; i < campos.size(); i++) {
      if (i != 0) {peticion = peticion + ",";}
      peticion = peticion + campos.get(i);
    }
    peticion = peticion + ") VALUES (";
    for (int i = 0; i < valores.size(); i++) {
      if (i != 0) { peticion = peticion + ","; }
      peticion = peticion + "\'" + valores.get(i) + "\'";
    }
    peticion = peticion + ");";
    insertCache = insertCache + peticion;
    insertCacheContador++;
  }
  sync();
  return null;
}
```

A.5.3 *Métodos de actualización: UPDATE*

Esta función (que ya se ha mostrado su uso en el método insertar), permite actualizar la información de una tupla concreta. Para ello, se requiere proporcionar el `ROWID` o identificador de tupla, que puede ser adquirido realizando una selección.

```
1  /**
2   * Función que actualiza una tupla en la tabla alojada en FusionTables
3   *
4   * @param tabla identificador de la tabla
5   * @param campos campo que será actualizado
6   * @param valores valor que será actualizado
7   * @param ROWID identificador de la tupla
8   * @return
9   */
10 public Sqlresponse update(String tabla, List<String> campos, List<String>
   ↪  valores, List<List<Object>> ROWIDs) {
11   for (Object ite : ROWIDs) {
12
13     String peticion = "UPDATE " + tabla + " SET ";
14
15     for (int i = 0; i < campos.size(); i++) {
16       if (i > 0) {
17         peticion = peticion + ",";
18       }
19       peticion = peticion + campos.get(i) + " = \'" + valores.get(i) +
         ↪  "\'";
20     }
21     peticion = peticion + "  WHERE ROWID = " + "\'" + ((String)
       ↪  ((List<String>) ite).get(0)) + "\'; ";
22
23     Logger.getGlobal().fine(peticion);
24     this.sql(peticion);
25   }
26   return null;
27 }
```

Figura A.10
Código: Métodos para Fusion Table: UPDATE

A.5.3.1 *Métodos de borrado: DELETE*

Esta función permite borrar una tupla concreta. Para ello, se requiere proporcionar el `ROWID` o identificador de tupla, que puede ser adquirido realizando una selección.

```
1  /**
2   * Función que elimina una tupla en la tabla alojada en FusionTables
3   *
4   * @param tabla identificador de la tabla
5   * @param ROWID identificador de la tupla
6   * @return
7   */
8  public Sqlresponse delete(String tabla, List<List<Object>> ROWIDs) {
9    String peticion;
10
11   for (Object ite : ROWIDs) {
12     peticion = "DELETE FROM " + tabla + "\" WHERE ROWID = " + "\'" +
       ↪  ((String) ((List<String>) ite).get(0)) + "\';";
13     this.sql(peticion);
14   }
15
16   return null;
17 }
```

Figura A.11
Código: Métodos para Fusion Table: DELETE

A.6 EJEMPLOS DE USO DE HERRAMIENTAS DE DIFUSIÓN

En este anexo, se presentan de forma breve las estructuras y fundamentos de los ejemplos de difusión presentados en el Capítulo 5.14.

A.6.1 *Integración con Google Maps*

Google nos ofrece dos maneras de incorporar la información alojada en Google Fusion Tables en un mapa de Google Maps. La primera de ella es un mecanismo nativo, que aunque es bastante potente, ofrece poco mecanismos avanzados y de personalización. El segundo método presentado, hace uso de la API REST para realizar una petición y crear de forma manual la capa.

A.6.1.1 *Forma nativa*

Google Maps funciona mediante *capas* (layer), que no son más que gráficos geolocalizados que son superpuestos sobre el mapa. De forma nativa, Google permite crear una capa sobre un mapa. En el siguiente código, se muestra como añadir una capa con la información relativa a una capa de Fusion Tables mediante Javascript:

Código A.1
Ejemplo de uso de Google Fusion Tables en Google Maps.

```
layer = new google.maps.FusionTablesLayer({      map: map,
     heatmap: { enabled: false },
     query: {
       select: "poligono",
       from: "1WENRMKLPrLdl-8WCKEOP6PTCwRkqCn88tRg-WuHV",
       where: "Intervalo = '2014-03-12 10:00:00'"
     },
     options: {
       styleId: 3,
       templateId: 5,
       suppressInfoWindows: true
     }
   });
```

A.6.1.2 *Mediante API REST*

El método anterior no permite mucha personalización (aunque si lo suficiente para un uso básico) ni permite realizar modificaciones en los datos u asociarles funcionalidades extras. Es por ello, que se presenta el siguiente código donde se hace uso de una capa de Google Maps usando la API REST directamente. En este caso, para realizar una capa con un "mapa de calor"

Código A.2
Ejemplo de uso de la API REST de Google Fusion Tables en Google Maps.

```
var query = 'select latitud,longitud,Total from
↪ 1WENRMKLPrLdl-8WCKE0P6PTCwRkqCn88tRg-WuHV where col1\x3e\x3e0 \x3d
↪ \x27'+c+'\x27 limit 1000';var request =
↪ gapi.client.fusiontables.query.sqlGet({ sql: query });

request.execute(function(response) {
  onDataFetched(response);
  });

function onDataFetched(response) {
  if (response.error) {
    alert('Unable to fetch data. ' + response.error.message +
        ' (' + response.error.code + ')');
  } else {
    drawHeatmap(extractLocations(response.rows));
  }
}

function extractLocations(rows) {
  var locations = [];
  for (var i = 0; i < rows.length; ++i) {
    var row = rows[i];
    if (row[0]) {
      var lat = row[0];
      var lng = row[1];
      if (lat && lng && !isNaN(lat) && !isNaN(lng)) {
        var latLng = new google.maps.LatLng(lat, lng);
        var weight = row[2];
         locations.push({ location: latLng, weight: parseFloat(weight) });
      }
    }
  }
  console.log(locations);
  return locations;
  }

function drawHeatmap(locations) {
var heatmap = new google.maps.visualization.HeatmapLayer({
   dissipating: true,
   gradient: ['rgba(102,255,0,0)', 'rgba(147,255,0,1)',
   ↪ 'rgba(193,255,0,1)','rgba(238,255,0,1)', 'rgba(244,227,0,1)',
   ↪ 'rgba(244,227,0,1)', 'rgba(249,198,0,1)', 'rgba(255,170,0,1)',
   ↪ 'rgba(255,113,0,1)', 'rgba(255,57,0,1)', 'rgba(255,0,0,1)'],
   opacity: 0.8,
   radius: 30,
   maxIntensity: 1000,
   data: locations
});
heatmap.setMap(map);
}
```

A.6.2 *Ejemplos de Rshiny*

Una aplicación `Rshiny` está compuesta de dos componentes principales, que se corresponden con el `backend` y `frontend` respectivamente.

En la parte del `backend` o servidor, se implementn las depencias del código, así como los elementos que se producirán desde el servidor. Un ejemplo se puede ver en el Código A.3.

Código A.3
Estructura del servidor en una aplicación `Rshiny`

```
library(shiny)
library(mobywit)

check <- function(){ return(now())}

shinyServer(function(input, output,session) {
  autoInvalidate_second <- reactiveTimer(1000, session = session)
  autoInvalidate <- reactiveTimer(60000, session = session)
  autoInvalidate_5m <- reactiveTimer(300000, session = session)
  #CARGAR CONTENIDO ESTÁTICO
  source("sources/mapa.R",TRUE)
  source("sources/log.R",TRUE)
  #CARGAR CONTENIDO DINÁMICO
  source("sources/variables.R",TRUE)
  source("sources/_cargarDatos.R",TRUE)
  #GENERAR LOS ANÁLISIS
  source("sources/info.R",TRUE)
  source("sources/pasos.R",TRUE)
  source("sources/simultaneos.R",TRUE)
  source("sources/multitud.R",TRUE)
  source("sources/fabricantes.R",TRUE)
  source("sources/densidades.R",TRUE)
  source("sources/stats.R",TRUE)
})
```

Un elemento importante del servidor son los elementos reactivos. Estos obedecen a eventos, como la pulsación de un botón o un pulso de un temporizador, para generar nueva respuesta desde el servidor. Se emplean principalmente para poder actualizar la información mostrada en el `frontend`. En el Código A.4 se presenta el uso de la función `autoinvalidate` para determinar que cierta información de salida tiene que ser invalidada al pasar 1 segundo.

Código A.4
Ejemplo de elemento autoinvalidado `Rshiny`

```
output$stats_duracion <- renderUI({
  autoInvalidate()
  m <- mean(rv$tiempos_entrada$duracion.x/60)
  m <- round(m,digits = 0)
  tags$h2(paste0("Duración media: ", m, " minutos "))
})
```

El `frontend` se define mediante la disposición de distintos elementos de salida definidos. En el Código A.5 se presenta una interfaz `responsive` de un `dashboard`.

Código A.5
Diseño de la interfaz en una aplicación `Rshiny`

```
library(shinythemes)
library(shiny)
library(plotly)
rowheight = "250px"
shinyUI(
  fluidPage(
    titlePanel("MOBYWIT - Monitorización por captación de comunicaciones inalámbricas de dispositivos Inteligentes", windowTitle = "MOBYWIT"),
    shinythemes::themeSelector(),
    tabsetPanel(
      tabPanel("Graficas",value = "graficas",
        fluidRow(
          column(3, htmlOutput("mapa")),
          column(2, includeHTML("web/esquema2.html")),
          column(4, includeHTML("web/esquema.html")),
          column(3, includeHTML("web/youtube.html"))
        ),
          fluidRow(
            column(3,
                   tags$h1("Estadísticas:"),
                   htmlOutput("today"),
                   htmlOutput("stats_visitantes"),
                   htmlOutput("stats_pasos"),
                   htmlOutput("stats_duracion")
                   ),
            column(3, plotlyOutput("pasos",height="300px" )),
            column(3, plotlyOutput("pasos_hora",height="300px" )),
            column(3, plotlyOutput("simultaneos",height="300px" ))
        ),
          fluidRow(
            column(4, plotlyOutput("multitud",height="400px" )),
            column(4, plotlyOutput("fabricantes_wifi",height="400px" )),
            column(2, plotlyOutput("duraction_visitas",height="400px" )),
            column(2, plotlyOutput("duraction_visitas_entrada",height="400px" ))
        ),
          fluidRow(
          )
      ),
      tabPanel("Estado",value = "estado",
          fluidRow(
            column(12, dataTableOutput("status"))
          )
        ),
      tabPanel("Twitter",value = "twitter",
               fluidRow(
                 column(12, includeHTML("web/twitter.html"))
               )
      ),
      tabPanel("Logs",value = "logs",
        fluidRow(
          column(6, htmlOutput("log"))
        )
      )
    )
  )
)
```

A.7 MEDIDAS DE PRECISIÍN Y EVALUACIÓN DE LA PREDICCIÓN

Sea P_i la predicción y R_i el valor real.

e_i

Absoulte error o Error absoluto:

$$e_i = P_i - R_i \tag{A.1}$$

$$e = \sum_{i=1}^{n} e_i \tag{A.2}$$

p_i

Absoulte percentage error o Error porcentual absoluto:

$$p_i = \frac{P_i - R_i}{R_i} \tag{A.3}$$

$$p = \sum_{i=1}^{n} e_i \tag{A.4}$$

MSE

Median squared error o Error cuadrático medio:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (e_i)^2 \tag{A.5}$$

RMSE

Root median squared error o Raíz del Error cuadrático medio:

$$RMSE = \sqrt{MSE} \tag{A.6}$$

MAE

Mean absolute error o Error medio absoluto:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |e_i| \tag{A.7}$$

MdAE

Median absolute error o Error mediano absoluto:

$$MdAE = mediana|e_i| \tag{A.8}$$

MAPE

Mean absolute percentage error o Media de error porcentual absoluto:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|e_i|}{R_i} \tag{A.9}$$

MdAPE

Median absolute percentage error o Mediana de error porcentual absoluto:

$$MdAPE = mediana \left(\frac{|e_i|}{R_i} \right) \times 100 \tag{A.10}$$

sMAPE

Symmetric Mean absolute percentage error o Media simétrica del error porcentual absoluto:

$$sMAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times |e_i|}{e_i} \tag{A.11}$$

sMdAPE

Symmetric Median absolute percentage error o Mediana simétrica del error porcentual absoluto:

$$sMdAPE = mediana \left(\frac{2 \times |e_i|}{e_i} \right) \tag{A.12}$$

r_i

Relative error o Error relativo:

$$r_i = \frac{e_i}{R_i} \times 100 \tag{A.13}$$

$$r = \sum_{i=1}^{n} r_i \tag{A.14}$$

MARE

Mean absolute relative error o Media del error relativo absoluto:

$$MARE = \frac{1}{n} \sum_{i=1}^{n} r_i \tag{A.15}$$

MdARE

Median absolute relative error o Mediana del error relativo absoluto:

$$MdARE = mediana\,(r_i) \tag{A.16}$$

GMARE

Geometric Mean absolute relative error o Medio geométrica del error relativo absoluto:

$$GMARE = \sqrt{\frac{1}{n}\prod_{i=1}^{n} r_i} \tag{A.17}$$

MDA

Mean Direction Accuracy o Precisión media de la dirección:

$$MDA = \frac{1}{n}\sum_{t}^{n} 1 \times signo\left(R_t - R_{t-1}\right) == signo\left(P_t - P_{t-1}\right) \tag{A.18}$$

A.8 CÓDIGO DE MUESTRA DEL SISTEMA EZEQUIEL

Código A.6
Ezequiel: Gestión de colas para Fusion Tables

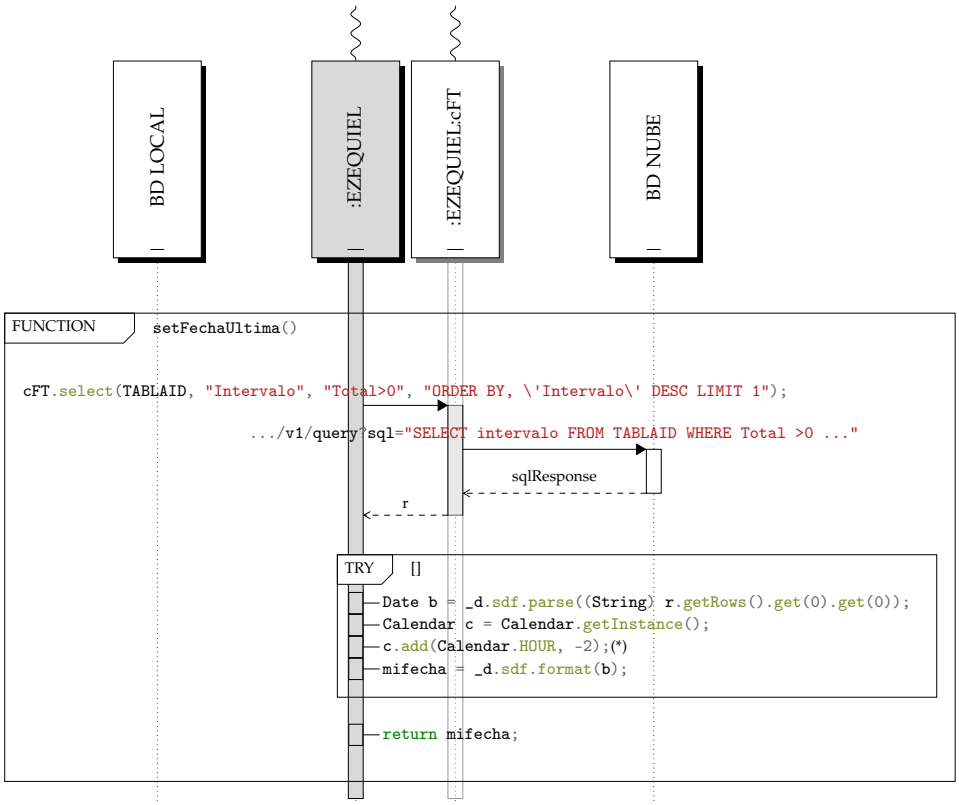
```
/**
 * Manejador de colas de peticiones de inserción a Fusion Table
 */
private static class colasManejadorInsert extends Thread {
  static String pendienteProcesar = null;
  Boolean vacio = false;

  public colasManejadorInsert() {
    this.setName("Manejador colas Inserts FT");
  }
  @Override
  public void run() {
    do {
      synchronized (insertCacheLista) {
        Logger.getGlobal().fine("Soy la hebra manejadora de FT " +
        ↪  insertCacheLista.size() + " elementos pendientes");
        if (insertCacheLista.isEmpty()) {
          try {
            insertCacheLista.wait();
            pendienteProcesar = insertCacheLista.poll();
          } catch (InterruptedException ex) {
            Logger.getGlobal().log(Level.SEVERE, null, ex);
          }
        } else if (pendienteProcesar == null ||
        ↪  "".equals(pendienteProcesar)) {
          pendienteProcesar = insertCacheLista.poll();
        }
      }
      Logger.getGlobal().fine("Soy la hebra manejadora y voy a procesar " +
      ↪  pendienteProcesar);
      if (sqlStatic(pendienteProcesar) == null) {
        try {
          //Ha fallado la transación, por lo que la tenemos que volvemos a
          ↪  procesar pasados unos segundos
          sleep(_c.getInt("ft.tiempo_espera_error_ms"));
        } catch (InterruptedException ex) {
          Logger.getGlobal().log(Level.SEVERE, null, ex);
        }
      } else {
        //Se ha procesado correctamente
        pendienteProcesar = null;
      }
    } while (true);
  }
}
```

Código A.7
Ezequiel: Subsistema de actualización de nodos

```
public boolean calcular() {
 Conectar conectar = new Conectar();
 try {
  Statement st = conectar.crearSt();
  rs = st.executeQuery("select * from nodo");
  List<String> valores = new ArrayList<>();
  while (rs.next()) {
   valores.add(rs.getString(1)); //idNodo
   valores.add(rs.getString(2)); //latitud
   valores.add(rs.getString(3)); //longitud
   valores.add(rs.getString(4)); //nombre
   double lat = Math.toRadians(new Double(rs.getString(2)));
   double lon = Math.toRadians(new Double(rs.getString(3)));
   double radio = 50.0 / 6378137.0;
   String cadena_poligono =
    ↪  "<Polygon><outerBoundaryIs><LinearRing><coordinates>";
    for (int j = 0; j <= 360; j = j + 15) {
     double r = Math.toRadians(j);
     double lat_rad = Math.asin(Math.sin(lat) * Math.cos(radio) +
      ↪  Math.cos(lat) * Math.sin(radio) * Math.cos(r));
     double lon_rad = Math.atan2(Math.sin(r) * Math.sin(radio) *
      ↪  Math.cos(lat), Math.cos(radio) - Math.sin(lat) *
      ↪  Math.sin(lat_rad));
     double lon_rad_f = ((lon + lon_rad + Math.PI) % (2 * Math.PI)) -
      ↪  Math.PI;
     cadena_poligono = cadena_poligono + Math.toDegrees(lon_rad_f) + "," +
      ↪  Math.toDegrees(lat_rad) + ",0.0 ";
    }
   cadena_poligono = cadena_poligono +
    ↪  "</coordinates></LinearRing></outerBoundaryIs></Polygon>";
   valores.add(cadena_poligono); //poligono
   Logger.getGlobal().fine("Insertando valores del
    ↪  nodo"+valores.toString());
   cFT.insert(TABLAID, campos, valores, check, 1);
   cFT.forzarSync(); (*)
   Logger.getGlobal().fine("Nodo insertado");
   valores.clear();
   }
   cFT.forzarSync();
   cFT.esperarSubida();
 } catch (SQLException ex) {Logger.getGlobal().log(Level.SEVERE, null,
  ↪  ex);}
 return false;
}
```

(*) Los nodos son especiales, dado que incorporan mucha más información que un paso o una trama, por lo que es recomendables no usar el mecanismo de lotes ya que se corre el riesgo de desbordar el tamaño de la petición a la API. Es por ello, que se fuerza la sincronización despúes de cada insercción.

Código A.8
EZEQUIEL: Comprobación última fecha de actualización en la nube.
(*) El valor 2 se encuentra parametrizado en el fichero de configuración, pero ha sido mostrado con su valor habitual para facilitar la legibilidad del ejemplo.



Código A.9
EZEQUIEL: Subsistema de subida de Pasos por hora.

```
public boolean calcular() {
 Conectar conectar = new Conectar();
 try {
  Statement st = conectar.crearSt();
  Logger.getGlobal().log(Level.INFO, "Calculando pasos en DB LOCAL");
  rs = st.executeQuery("CALL agrupaPasosPorIntervalosNodosSeparados('" +
   ↪  fecha + "','" + _d.sdf.format(Calendar.getInstance().getTime()) +
   ↪  "','" + 60 + "')");
  List<String> valores = new ArrayList<>();
  Logger.getGlobal().log(Level.INFO, "Subiendo información a la Nube");
  while (rs.next()) {
   valores.add(rs.getString(1)); //Intervalo
   valores.add(rs.getString(2)); //idNodo
   valores.add(rs.getString(3)); //Total
   try {
    Logger.getGlobal().fine("Valores:" + rs.getString(2) + " " +
     ↪  rs.getInt(3) + " " + rs.getString(6));
     infoNodo_pasoPorHora.update(rs.getString(2), rs.getInt(3),
      ↪  rs.getString(6));
   } catch (Exception ex) {Logger.getGlobal().log(Level.SEVERE,
    ↪  ex.getMessage());}
    cFT.insert(TABLAID, campos, valores, check, 2);
    valores.clear();
   }
   Logger.getGlobal().log(Level.INFO, "Todos los valores procesados.");
   cFT.forzarSync();
   Logger.getGlobal().log(Level.INFO, "Esperando al envío y confirmación de
    ↪  los valores en la nube.");
   cFT.esperarSubida();
   Logger.getGlobal().log(Level.INFO, "Todos los valores subidos a la
    ↪  nube.");
 } catch (SQLException ex) {
  Logger.getGlobal().log(Level.SEVERE, "Fallo en cálculo de los pasos. " +
   ↪  ex.getMessage(), ex);
 }
 return true;
}
```

Código A.10
Código: Fusion Tables: Subsitema de subida de Trazas por hora

```
1  public boolean calcular() {
2   Conectar conectar = new Conectar();
3   try {
4     Statement st = conectar.crearSt();
5     rs = st.executeQuery("CALL localizaTrazasNodos('" + fecha + "','" +
      ↪  _d.sdf.format(Calendar.getInstance().getTime()) + "','" + 60 +
      ↪  "')");
6     List<String> valores = new ArrayList<>();
7   while (rs.next()) {
8     valores.add(rs.getString(1)); //Fecha
9     valores.add(rs.getString(2)); //Origen
10    valores.add(rs.getString(3)); //Destino
11    valores.add(rs.getString(4)); //total
12    valores.add(rs.getString(5)); //Diferencia
13    String poligono = "<LineString>  <coordinates>   "+rs.getString(7)+",
      ↪  "+rs.getString(6)+", 0.  "+rs.getString(9)+", "+rs.getString(8)+", 0.
      ↪  </coordinates> </LineString>";
14    valores.add(poligono); //poligono
15    double earthRadius = 3958.75;
16    double dLat = Math.toRadians(rs.getDouble(8)-rs.getDouble(6));
17    double dLng = Math.toRadians(rs.getDouble(9)-rs.getDouble(7));
18    double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
19         Math.cos(Math.toRadians(rs.getDouble(6))) *
          ↪  Math.cos(Math.toRadians(rs.getDouble(8))) *
20         Math.sin(dLng/2) * Math.sin(dLng/2);
21    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
22    double dist = earthRadius * c;
23    int meterConversion = 1609;
24    float t = (float) dist * meterConversion;
25    valores.add(Float.toString(t));
26    cFT.insert(TABLAID, campos, valores, check);
27    valores.clear();
28    }
29     cFT.forzarSync();
30     cFT.esperarSubida();
31  } catch (SQLException ex) {Logger.getGlobal().log(Level.SEVERE, null, ex);
    ↪  }
32  return false;
33  }
```

Código A.11
EZEQUIEL: Cliente Actualizador de FT.

```
1  public void start() {
2   if (_c.getBool("ft.primera_vez")) {
3    Logger.getGlobal().log(Level.INFO, "Esperando 10 minutos para el comienzo
      ↪ del sincronizado forzado en la NUBE");
4    try {
5     Thread.sleep(1000);
6    } catch (InterruptedException ex) {
7     Logger.getLogger(ActualizadorFT.class.getName()).log(Level.SEVERE, null,
       ↪ ex);
8    }
9    Logger.getGlobal().log(Level.INFO, "Comenzando sincronizado forzado en la
      ↪ nube");
10
11   //Si es nuestra primera vez, habrá que hacerlo desde el origen de los
      ↪ tiempos!
12   Nodos n = new Nodos();
13   n.calcular();
14   Logger.getGlobal().log(Level.INFO, "Nodos sincronizados");
15   PasosPorHoras h = new PasosPorHoras("2018-07-01 00:00:00 "); (*)
16   h.check = true;
17   h.calcular();
18   Logger.getGlobal().log(Level.INFO, "Pasos sincronizados");
19   TrazasPorHoras t = new TrazasPorHoras("2018-07-01 00:00:00 "); (*)
20   t.check = true;
21   t.calcular();
22   Logger.getGlobal().log(Level.INFO, "Trazas sincronizadas");
23   _c.set("ft.primera_vez", "false");
24   Logger.getGlobal().log(Level.INFO, "Sincronizado forzado en la nube
      ↪ programado COMPLETO.");
25   }
26   Timer timer = new Timer("ActualizadorFusionTable", true);
27
28   timer.scheduleAtFixedRate(temporizador, _c.getLong("ft.tiempo_espera"),
      ↪ _c.getLong("ft.periodo_actualizacion")); (**)
29  }
```

(*) Los valores son almacenados en el fichero de configuración pero son mostrados como fecha en el ejemplo para una mayor legibilidad.
(**)Estos valores no han sido ocultados para poder explicar su funcionamiento. `ft.tiempo_espera` impone un tiempo de espera a la tarea programada antes de su primer inicio. La variable `ft.periodo_actualizacion` indica cada cuanto tiempo se producirá la sincronización en la nube. La programación de la tarea programa consta de una pequeña parte programada, eliminando el tiempo de cómputo del tiempo de sincronización. Así por ejemplo, si la tarea se tiene que iniciar cada 60 segundos, pero la última tarea tardó 15 segundos, la próxima sincronización se realizará a los 45 segundos de haber terminado la sincronización anterior.

Código A.12
EZEQUIEL: Cliente Actualizador de FT - Tarea programada

```
this.temporizador = new TimerTask() {
 @Override
 public void run() {
  Logger.getGlobal().log(Level.INFO, "Comenzando sincronizado programado en
  ↪  la nube.");
  Nodos n = new Nodos();
  n.calcular();
  Logger.getGlobal().log(Level.INFO, "Nodos sincronizados");

  PasosPorHoras h = new PasosPorHoras();
  Logger.getGlobal().log(Level.INFO, "Sincronizando Pasos desde " +
  ↪  h.getFecha());
  h.check = true;
  h.calcular();
  Logger.getGlobal().log(Level.INFO, "Pasos sincronizados.");

  TrazasPorHoras t = new TrazasPorHoras();
  Logger.getGlobal().log(Level.INFO, "Pasos sincronizados desde " +
  ↪  t.getFecha());
  t.check = true;
  t.calcular();

  Logger.getGlobal().log(Level.INFO, "Trazas sincronizadas.");
  Logger.getGlobal().log(Level.INFO, "Sincronizado programado en la nube
  ↪  programado COMPLETO.");
  }
};
```

BIBLIOGRAFÍA

[1] Baher Abdulhai, Himanshu Porwal y Will Recker. "Short-term traffic flow prediction using neuro-genetic algorithms". En: *ITS Journal-Intelligent Transportation Systems Journal* 7.1 (2002), págs. 3-41.

[2] Fadel Adib y Dina Katabi. *See through walls with WiFi!* Vol. 43. 4. ACM, 2013.

[3] Ali-Reza Adl-Tabatabai, Michał Cierniak, Guei-Yuan Lueh, Vishesh M Parikh y James M Stichnoth. "Fast, effective code generation in a just-in-time Java compiler". En: *ACM SIGPlAN Notices*. Vol. 33. 5. ACM. 1998, págs. 280-290.

[4] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[5] Charu C Aggarwal. "Outlier analysis". En: *Data mining*. Springer. 2015, págs. 237-263.

[6] Subutai Ahmad, Alexander Lavin, Scott Purdy y Zuha Agha. "Unsupervised real-time anomaly detection for streaming data". En: *Neurocomputing* 262 (2017), págs. 134-147.

[7] N. Ahmed, A. Ghose, A.K. Agrawal, C. Bhaumik, V. Chandel y A. Kumar. "SmartEvacTrak: A people counting and coarse-level localization solution for efficient evacuation of large buildings". En: *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. Mar. de 2015, págs. 372-377.

[8] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar e Hisham El-Shishiny. "An empirical comparison of machine learning models for time series forecasting". En: *Econometric Reviews* 29.5-6 (2010), págs. 594-621.

[9] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos y Eleftherios Kayafas. "License plate recognition from still images and video sequences: A survey". En: *IEEE Transactions on intelligent transportation systems* 9.3 (2008), págs. 377-391.

[10] Jacob Andersson y Kerstin Ersson. *Pattern matching with neural networks for the PANDA at FAIR experiment*. Inf. téc. Uppsala Universitet, 2018.

[11] *ARM Cortex A7 Specifications*. 2017. URL: https://developer.arm.com/products/processors/cortex-a/cortex-a7.

[12] Daniel Arp, Erwin Quiring, Christian Wressnegger y Konrad Rieck. "Privacy Threats through Ultrasonic Side Channels on Mobile Devices". En: ().

[13] *Arudino Company*. 2017. URL: www.arduino.cc/.

[14] G Asada, M Dong, TS Lin, F Newberg, G Pottie, WJ Kaiser y HO Marcy. "Wireless integrated network sensors: Low power systems on a chip". En: *Solid-State Circuits Conference, 1998. ESSCIRC'98. Proceedings of the 24th European*. IEEE. 1998, págs. 9-16.

[15] Kevin Ashton. Auto-ID Center del MIT. 1999.

[16] Atilla Aslanargun, Mammadagha Mammadov, Berna Yazici y Senay Yolacan. "Comparison of ARIMA, neural networks and hybrid models in time series: tourist arrival forecasting". En: *Journal of Statistical Computation and Simulation* 77.1 (2007), págs. 29-53.

[17] UN General Assembly. "Universal declaration of human rights". En: *UN General Assembly* (1948).

[18] V. Assimakopoulos y K. Nikolopoulos. "The Theta Model: A Decomposition Approach to Forecasting". En: *International Journal of Forecasting* 16.4 (2000), págs. 521-530.

[19] Grupo Atolon. *Method Lynce for crowd measurement*. 2011. URL: https://web.archive.org/web/20110915235904/http://lynce.es/es/metodo.php.

[20] Luigi Atzori, Antonio Iera y Giacomo Morabito. "The Internet of Things: A survey". En: *Computer Networks* 54.15 (2010), págs. 2787-2805. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2010.05.010.

[21] E Michael Azoff. *Neural network time series forecasting of financial markets*. John Wiley & Sons, Inc., 1994.

[22] C Narendra Babu y B Eswara Reddy. "Predictive data mining on average global temperature using variants of ARIMA models". En: *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012)*. IEEE. 2012, págs. 256-260.

[23] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis e Y. Portugali. "Smart cities of the future". En: *The European Physical Journal Special Topics* 214.1 (dic. de 2012), págs. 481-518.

[24] Alfredo J Berard, James L Mentzer y David C Nixon. *Cellular/GPS system for vehicle tracking*. US Patent 5,515,043. Mayo de 1996.

[25] Roberto Bez, Emilio Camerlenghi, Alberto Modelli y Angelo Visconti. "Introduction to flash memory". En: *Proceedings of the IEEE* 91.4 (2003), págs. 489-502.

[26] Christopher M Bishop. "Pattern recognition". En: *Machine Learning* 128 (2006), págs. 1-58.

[27] Peter Bloomfield. "Trends in global temperature". En: *Climatic change* 21.1 (1992), págs. 1-16.

[28] Victor A. Bloomfield. *Using R for Numerical Analysis in Science and Engineering*. Chapman & Hall/CRC, 2014. ISBN: 978-1439884485. URL: http://www.crcpress.com/product/isbn/9781439884485.

[29] SIG Bluetooth. "Specification of the Bluetooth System, version 5.0". En: *Dic* (2016).

[30] Piero P Bonissone. "Soft computing: the convergence of emerging reasoning technologies". En: *Soft computing* 1.1 (1997), págs. 6-18.

[31] Atreyi Bose y Chuan Heng Foh. "A practical path loss model for indoor WiFi positioning enhancement". En: *2007 6th International Conference on Information, Communications & Signal Processing*. IEEE. 2007, págs. 1-5.

[32] B Bowerman, J Braverman, J Taylor, H Todosow y U Von Wimmersperg. "The vision of a smart city". En: *2nd International Life Extension Technology Workshop, Paris*. Vol. 28. 2000.

[33] G.E. Box y G.M. Jenkins. *Time series analysis: forecasting and control*. San Francisco: Holden Day, 1976.

[34] George EP Box y Gwilym M Jenkins. *Time series analysis: forecasting and control*. Holden-Day, 1976.

[35] Paul C Box y Joseph C Oppenlander. "Manual of traffic engineering studies". En: (1976).

[36] Dirk Brockmann, Lars Hufnagel y Theo Geisel. "The scaling laws of human travel". En: *Nature* 439.7075 (2006), págs. 462-465.

[37] Peter J Brockwell, Richard A Davis y Matthew V Calder. *Introduction to time series and forecasting*. Vol. 2. Springer, 2002.

[38] Alberto Broggi y Simona Berte. "Vision-based road detection in automotive systems: A real-time expectation-driven approach". En: *Journal of Artificial Intelligence Research* 3 (1995), págs. 325-348.

[39] David S Broomhead y David Lowe. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Inf. téc. Royal Signals y Radar Establishment Malvern (United Kingdom), 1988.

[40] GWTA Bruinderink y E Hazebroek. "Ungulate traffic collisions in Europe". En: *Conservation Biology* 10.4 (1996), págs. 1059-1067.

[41] Colin Buchanan. *Mixed blessing: the motor in Britain*. L. Hill, 1958.

[42] J Mark Bull, Lorna A Smith, Lindsay Pottage y Robin Freeman. "Benchmarking Java against C and Fortran for scientific applications". En: *Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande*. ACM. 2001, págs. 97-105.

[43] Francesco Calabrese, Francisco C Pereira, Giusy Di Lorenzo, Liang Liu y Carlo Ratti. "The geography of taste: analyzing cell-phone mobility and social events". En: *International Conference on Pervasive Computing*. Springer. 2010, págs. 22-37.

[44] Andrea Caragliu, Chiara Del Bo y Peter Nijkamp. "Smart Cities in Europe". En: *Journal of Urban Technology* 18.2 (abr. de 2011), págs. 65-82. ISSN: 1063-0732. DOI: 10.1080/10630732.2011.601117. URL: http://www.tandfonline.com/doi/abs/10.1080/10630732.2011.601117.

[45] Peter Carnes. *TraffaxInc*. http://www.TraffaxInc.com/.

[46] Stephen Cass. *The 2017 Top Programming Languages*. 2017. URL: https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages.

[47] Luca Catarinucci, Riccardo Colella, Luca Mainetti, Luigi Patrono, Stefano Pieretti, Ilaria Sergi y Luciano Tarricone. "Smart RFID antenna system for indoor tracking and behavior analysis of small animals in colony cages". En: *IEEE Sensors Journal* 14.4 (2014), págs. 1198-1206.

[48] H. Celik, A. Hanjalic y E.A. Hendriks. "Towards a Robust Solution to People Counting". En: *Image Processing, 2006 IEEE International Conference on*. Oct. de 2006, págs. 2401-2404.

[49] Varun Chandola, Arindam Banerjee y Vipin Kumar. "Anomaly detection: A survey". En: *ACM computing surveys (CSUR)* 41.3 (2009), pág. 15.

[50] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung y Sei-Wan Chen. "Automatic license plate recognition". En: *IEEE transactions on intelligent transportation systems* 5.1 (2004), págs. 42-53.

[51] Chris Chatfield. "The holt-winters forecasting procedure". En: *Applied Statistics* (1978), págs. 264-279.

[52] Peiyuan Chen, Troels Pedersen, Birgitte Bak-Jensen y Zhe Chen. "ARIMA-based time series model of stochastic wind power generation". En: *IEEE transactions on power systems* 25.2 (2010), págs. 667-676.

[53] Sing Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan y Pravin Varaiya. "Traffic measurement and vehicle classification with single magnetic sensor". En: *Transportation research record: journal of the transportation research board* 1917 (2005), págs. 173-181.

[54] CA Coello Coello y Margarita Reyes Sierra. "A coevolutionary multi-objective evolutionary algorithm". En: *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* Vol. 1. IEEE. 2003, págs. 482-489.

[55] Benjamin Coifman. "Using dual loop speed traps to identify detector errors". En: *Transportation Research Record: Journal of the Transportation Research Board* 1683 (1999), págs. 47-58.

[56] Benjamin Coifman, David Beymer, Philip McLauchlan y Jitendra Malik. "A real-time computer vision system for vehicle tracking and traffic surveillance". En: *Transportation Research Part C: Emerging Technologies* 6.4 (1998), págs. 271-288.

[57] Jeremy Cole. *B+Tree index structures in InnoDB*. https://blog.jcole.us/2013/01/10/btree-index-structures-in-innodb/. 2014.

[58] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt y col. "A system for video surveillance and monitoring". En: (2000).

[59] Brundtland Commission y col. "Our common future: Report of the World Commission on Environment and Development". En: *UN Documents Gatheringa Body of Global Agreements* (1987).

[60] ISO/IEC Joint Technical Committee. *ISO/IEC 18092: Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)*. 2013. URL: https://www.iso.org/standard/56692.html.

[61] ISO/IEC Joint Technical Committee. *ISO/IEC 14443: Identification cards — Contactless integrated circuit cards — Proximity cards*. 2016. URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:14443:-1:ed-3:v1:en.

[62] Steven J Cooke, Jonathan D Midwood, Jason D Thiem, Peter Klimley, Martyn C Lucas, Eva B Thorstad, John Eiler, Chris Holbrook y Brendan C Ebner. "Tracking animals in freshwater with electronic tags: past, present and future". En: *Animal Biotelemetry* 1.5 (2013).

[63] Corinna Cortes y Vladimir Vapnik. "Support-vector networks". En: *Machine learning* 20.3 (1995), págs. 273-297.

[64] Alan Cudmore. "Pi-Sat: A Low Cost Small Satellite and Distributed Spacecraft Mission System Test Platform". En: (2015). URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150023353.pdf.

[65] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani y Edgar Weippl. "IMSI-catch me if you can: IMSI-catcher-catchers". En: *Proceedings of the 30th annual computer security applications Conference*. ACM. 2014, págs. 246-255.

[66] Michael R D'angelo, Geoffrey M Eggert, Robert G Bresler y Joseph E Qualitz. *Anti-theft device with alarm screening*. US Patent 5,963,131. Oct. de 1999.

[67] James W Davis y Mark A Keck. "A two-stage template approach to person detection in thermal imagery". En: *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*. Vol. 1. IEEE. 2005, págs. 364-369.

[68] James W Davis y Vinay Sharma. "Robust Background-Subtraction for Person Detection in Thermal Imagery." En: *CVPR Workshops*. 2004, pág. 128.

[69] Federal Highway Administration Washington DC. *Manual of Uniform Traffic Control Devices for Streets and Highways*. Inf. téc. 2003.

[70] L Peter Deutsch. "GZIP file format specification version 4.3". En: (1996).

[71] *Documentación mecanismos autorización Google Fusion Tables*. URL: https://developers.google.com/fusiontables/docs/v1/using%5C#auth.

[72] *Documentación OAuth2*. URL: https://developers.google.com/accounts/docs/OAuth2.

[73] Stuart E Dreyfus. "Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure". En: *Journal of guidance, control, and dynamics* 13.5 (1990), págs. 926-928.

[74] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola y Vladimir Vapnik. "Support vector regression machines". En: *Advances in neural information processing systems*. 1997, págs. 155-161.

[75] Nathan Eagle, Alex Sandy Pentland y David Lazer. "Inferring friendship network structure by using mobile phone data". En: *Proceedings of the national academy of sciences* 106.36 (2009), págs. 15274-15278.

[76] Erik Eckermann. "World history of the automobile". En: *Training* 2011 (2001), págs. 04-20.

[77] AA El Desouky y MM El Kateb. "Hybrid adaptive techniques for electric-load forecast using ANN and ARIMA". En: *IEE Proceedings-Generation, Transmission and Distribution* 147.4 (2000), págs. 213-217.

[78] CTN 200 - NORMAS BÁSICAS ELÉCTRICAS. "UNE 20324:1993 - Grados de protección proporcionados por las envolventes (Código IP). (CEI 529:1989)." En: (2017). URL: http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0001122#.WfC09iGLRhE.

[79] Instituto Nacional de Estadística. *Nota de Prensa sobre Equipamiento y Uso de Tecnologías de Información y Comunición en los Hogares*. 2015.

[80] Hsuan-Ming Feng. "Self-generation RBFNs using evolutional PSO learning". En: *Neurocomputing* 70.1-3 (2006), págs. 241-251.

[81] Sheikh Ferdoush y Xinrong Li. "Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications". En: *Procedia Computer Science* 34 (2014), págs. 103-110.

[82] Roy T Fielding y Richard N Taylor. *Architectural styles and the design of network-based software architectures*. Vol. 7. University of California, Irvine Doctoral dissertation, 2000.

[83] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach y Tim Berners-Lee. *RFC 2616 - Hypertext transfer protocol–HTTP/1.1*. Inf. téc. 1999. URL: https://tools.ietf.org/html/rfc2616.

[84] Gabe Fierro, Omar Rehmane, Andrew Krioukov y David Culler. "Zone-level Occupancy Counting with Existing Infrastructure". En: *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. BuildSys '12. Toronto, Ontario, Canada: ACM, 2012, págs. 205-206. ISBN: 978-1-4503-1170-0.

[85] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.

[86] Jeffrey Fischer. "NFC in cell phones: The new paradigm for an interactive world [Near-Field Communications]". En: *IEEE communications Magazine* 47.6 (2009), págs. 22-28.

[87] Mendez F.M. *System and method for monitoring people and/or vehicles in urban environments*. EP Patent App. EP20,080,805,357. Mayo de 2011. URL: http://www.google.com/patents/EP2325823A1?cl=en.

[88] Lawrence J Fogel, Alvin J Owens y Michael J Walsh. "Artificial intelligence through simulated evolution". En: (1966).

[89] Anthony J Fox. "Outliers in time series". En: *Journal of the Royal Statistical Society. Series B (Methodological)* (1972), págs. 350-363.

[90] Julien Freudiger. "How talkative is your mobile device?: an experimental study of Wi-Fi probe requests". En: *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM. 2015, pág. 8.

[91] Yoav Freund y Robert E Schapire. "Large margin classification using the perceptron algorithm". En: *Machine learning* 37.3 (1999), págs. 277-296.

[92] Emory Fry y Leslie A Lenert. "MASCAL: RFID Tracking of Patients, Staff and Equipment to Enhance HospitalResponse to Mass Casualty Events." En: *AMIA*. 2005.

[93] Wayne A Fuller. *Introduction to statistical time series*. Vol. 428. John Wiley & Sons, 2009.

[94] Janusz Gajda, Ryszard Sroka, Marek Stencel, Andrzej Wajda y Tadeusz Zeglen. "A vehicle classification based on inductive loop detectors". En: *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*. Vol. 1. IEEE. 2001, págs. 460-464.

[95] Amir Gandomi y Murtaza Haider. "Beyond the hype: Big data concepts, methods, and analytics". En: *International Journal of Information Management* 35.2 (2015), págs. 137-144.

[96] Julian W Gardner y Vijay K Varadan. *Microsensors, MEMS and smart devices*. John Wiley & Sons, Inc., 2001.

[97] Fosca Giannotti, Mirco Nanni, Fabio Pinelli y Dino Pedreschi. "Trajectory pattern mining". En: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2007, págs. 330-339.

[98] Rudolf Giffinger y Haindlmaier Gudrun. "Smart cities ranking: an effective instrument for the positioning of the cities?" En: *ACE: Architecture, City and Environment* 4.12 (2010), págs. 7-26.

[99] Daniel Giusto, Antonio Iera, Giacomo Morabito y Luigi Atzori. *The internet of things: 20th Tyrrhenian workshop on digital communications*. Springer Science & Business Media, 2010.

[100] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[101] Hector Gonzalez, Alon Y Halevy, Christian S Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen y Jonathan Goldberg-Kidon. "Google fusion tables: web-centered data management and collaboration". En: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM. 2010, págs. 1061-1066.

[102] Hector Gonzalez, Alon Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley y Warren Shen. "Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud". En: *SoCC10 Proceedings of the 1st ACM symposium on Cloud computing*. ACM, 2010, págs. 175-180.

[103] Marta C Gonzalez, Cesar A Hidalgo y Albert-Laszlo Barabasi. "Understanding individual human mobility patterns". En: *Nature* 453.7196 (2008), págs. 779-782.

[104] Matthew Gordon. "Big Data: It's Not the Size That Matters". En: *J. Nat'l Sec. L. & Pol'y* 7 (2014), pág. 311.

[105] Robert L Gordon, Robert A Reiss, Herman Haenel, E Ryerson Case, Robert L French, Abbas Mohaddes y Ronald Wolcott. *Traffic Control Systems Handbook*. Inf. téc. 1996.

[106] C.W.J. Granger. "Testing for causality: A personal viewpoint". En: *Journal of Economic Dynamics and Control* 2 (1980), págs. 329-352.

[107] Marianne L Gras. "The legal regulation of CCTV in Europe". En: *Surveillance & Society* 2.2/3 (2002).

[108] The Hammershith group. *The Internet of things: Networked objects and smart devices*. The hammersmith group, 2010.

[109] Erico Guizzo. "How google's self-driving car works". En: *IEEE Spectrum Online, October* 18 (2011).

[110] *Gumstix Company*. 2017. URL: https://www.gumstix.com/.

[111] Keijo MJ Haataja y Konstantin Hypponen. "Man-in-the-middle attacks on bluetooth: a comparative analysis, a novel attack, and countermeasures". En: *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*. IEEE. 2008, págs. 1096-1102.

[112] Theo Haerder y Andreas Reuter. "Principles of transaction-oriented database recovery". En: *ACM Computing Surveys (CSUR)* 15.4 (1983), págs. 287-317.

[113] David J Hand. "Mining the past to determine the future: Problems and possibilities". En: *International journal of Forecasting* 25.3 (2009), págs. 441-451.

[114] Colin Harrison, Barbara Eckman, Rick Hamilton, Perry Hartswick, Jayant Kalagnanam, Jurij Paraszczak y Peter Williams. "Foundations for smarter cities". En: *IBM Journal of Research and Development* 54.4 (2010), págs. 1-16.

[115] Guy Harrison y Steven Feuerstein. *MySQL stored procedure programming*. .ºReilly Media, Inc.", 2006.

[116] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi y S. Matsueda. "People count system using multi-sensing application". En: *Solid State Sensors and Actuators, 1997. TRANSDUCERS '97 Chicago., 1997 International Conference on*. Vol. 2. Jun. de 1997, 1291-1294 vol.2.

[117] Kazuhiko Hashimoto, Chihiro Kawaguchi, Satoshi Matsueda, Katsuya Morinaka y Nobuyuki Yoshiike. "People-counting system using multisensing application". En: *Sensors and Actuators A: Physical* 66.1-3 (1998), págs. 50-55.

[118] S. Haykin. *Neural Networks: A Comprehensive Approach*. I. Piscataway, USA: EEE Computer Society Press, 1994.

[119] HS Hippert, DW Bunn y RC Souza. "Large neural networks for electricity load forecasting: Are they overfitted?" En: *International Journal of forecasting* 21.3 (2005), págs. 425-434.

[120] Bart Hobijn, Philip Hans Franses y Marius Ooms. "Generalizations of the KPSS-test for stationarity". En: *Statistica Neerlandica* 58.4 (2004), págs. 483-502.

[121] Hochbaum y Shmoys. "A best possible heuristic for the k-center problem". En: *Mathematics of Operations Research* 10.2 (1985), págs. 180-184.

[122] Bruce Hopkins y Ranjith Antony. *Bluetooth for Java*. Vol. 20003. Springer, 2003.

[123] Richard G. Hoptroff. "The principles and practice of time series forecasting and business modelling using neural nets". En: *Neural Computing & Applications* 1.1 (1993), págs. 59-66.

[124] Peter J Huber. *Robust statistics*. Springer, 2011.

[125] Chih-Lyang Hwang y Li-Jui Chang. "Trajectory tracking and obstacle avoidance of car-like mobile robots in an intelligent space using mixed H 2/H infinite decentralized control". En: *IEEE/ASME Transactions on Mechatronics* 12.3 (2007), págs. 345-352.

[126] R.J. Hyndman, A.B. Koehler, J.K. Ord y R.D. Snyder. *Forecasting with Exponential Smoothing*. Springer, 2008. ISBN: 978-3-540-71916-8.

[127] Rob J Hyndman y Yeasmin Kh. "Automatic time series forecasting: The forecast package for R". En: *Journal of Statistical Software* (2008).

[128] Rob J Hyndman, Anne B Koehler, Ralph D Snyder y Simone Grose. "A state space framework for automatic forecasting using exponential smoothing methods". En: *International Journal of Forecasting* 18.3 (2002), págs. 439-454.

[129] "IEEE BT (2013). The IEEE public BT OUI listing. Retrieved Dec 30, 2013, from `http://standards.ieee.org/develop/regauth/oui/oui.txt`". En:

[130] Apple INC. *Apple EULA*. URL: `http://www.apple.com/legal/sla/`.

[131] Facebook INC. *Facebook EULA*. URL: `https://www.facebook.com/terms`.

[132] Google INC. *Google EULA*. URL: `https://www.google.com/intl/es/policies/privacy/`.

[133] Satu Innamaa. "Short-term prediction of travel time using neural networks on an interurban highway". En: *Transportation* 32.6 (2005), págs. 649-669.

[134] *Interactive: The Top Programming Languages 2018*. 2018. URL: https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018.

[135] Landt Jerry y C Barbara. "Shrouds of Time: The history of RFID". En: *AIM Publication* (2001).

[136] Xiaomo Jiang y Hojjat Adeli. "Dynamic wavelet neural network model for traffic flow forecasting". En: *Journal of transportation engineering* 131.10 (2005), págs. 771-779.

[137] Ana Luz Jiménez Ortega. *Gestión Técnica Tráfico - Temario Oposiciones*. Inf. téc. Dirección General de Tráfico España, 2013.

[138] Young-Kee Jung y Yo-Sung Ho. "A feature-based vehicle tracking system in congested traffic video sequences". En: *Pacific-Rim Conference on Multimedia*. Springer. 2001, págs. 190-197.

[139] Joseph M Kahn, Randy H Katz y Kristofer SJ Pister. "Next century challenges: mobile networking for "Smart Dust"". En: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM. 1999, págs. 271-278.

[140] Olavi Kärner. "ARIMA representation for daily solar irradiance and surface air temperature time series". En: *Journal of Atmospheric and Solar-Terrestrial Physics* 71.8-9 (2009), págs. 841-847.

[141] V Kastrinaki, Michalis Zervakis y Kostas Kalaitzakis. "A survey of video processing techniques for traffic applications". En: *Image and vision computing* 21.4 (2003), págs. 359-381.

[142] JL Kay. "Measures of Effectiveness". En: *Proceeding of the International Symposium on Traffic COntrol Systems* (1976).

[143] A Kejariwal. *Twitter Engineering: Introducing practical and robust anomaly detection in a time series*. 2015.

[144] *Ken Olsen*. Fundador de Digital Equipment Corportation. 1977.

[145] Maurice George Kendall y col. "The advanced theory of statistics." En: *The advanced theory of statistics.* 2nd Ed (1946).

[146] A Keranen y M Kovatsch. *RESTful Design for Internet of Things Systems*. 2015.

[147] Akiba Kevin Townsend Carles Cufí y Robert Davidson. *Getting Started with Bluetooth Low Energy Tools and Techniques for Low-Power Networking*. OREALLY, 2014. URL: http://gen.lib.rus.ec/book/index.php?md5=32E42D5BCF6B150AFAB24AB74BB6383D.

[148] So-Hyeon Kim, Do-Hyeun Kim y Hee-Dong Park. "Animal situation tracking service using RFID, GPS, and sensors". En: *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. IEEE. 2010, págs. 153-156.

[149] ALS King, AM Valença, ACO Silva, T Baczynski, MR Carvalho y AE Nardi. "Nomophobia: Dependency on virtual environments or social phobia?" En: *Computers in Human Behavior* 29.1 (2013), págs. 140-144.

[150] Lawrence A Klein. *Sensor technologies and data requirements for ITS*. 2001.

[151] T. Kohonen, J. Hynninen, J. Kangas y J. Laaksonen. *SOM PAK: The Self-Organizing Map Program Package*. Technical Report A31. http://www.cis.hut.fi/nnrc/nnrc-programs.html. Helsinki University of Technology, 1996.

[152] Teuvo Kohonen. "The Self-Organizing Map". En: *Proceedings of the IEEE* 78.9 (sep. de 1990), págs. 1464-1480.

[153] Vassilis Kostakos. "Using Bluetooth to capture passenger trips on public transport buses". En: *arXiv preprint arXiv:0806.0874* (2008).

[154] Dennis Kügler. ""Man in the Middle" Attacks on Bluetooth". En: *International Conference on Financial Cryptography*. Springer. 2003, págs. 149-161.

[155] Cheng-Hao Kuo y Ram Nevatia. "How does person identity recognition help multi-person tracking?" En: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, págs. 1217-1224.

[156] Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt y Yongcheol Shin. "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" En: *Journal of econometrics* 54.1-3 (1992), págs. 159-178.

[157] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. "Deep learning". En: *nature* 521.7553 (2015), pág. 436.

[158] Jin-Shyan Lee, Yu-Wei Su y Chung-Chou Shen. "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi". En: *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. Ieee. 2007, págs. 46-51.

[159] Joo-Yub Lee, Cheal-Hwan Yoon, Hyunjae Park y Jungmin So. "Analysis of location estimation algorithms for wifi fingerprint-based indoor localization". En: *Proc. 2nd Int. Conf. Softw. Technol*. Vol. 19. 2013, págs. 89-92.

[160] Lee y Brent Ware. *Open Source Development with LAMP: Using Linux, Apache, MySQL and PHP*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2002, 2002.

[161] Robert Harper Lees. *Inductive loop sensor for traffic detection, and traffic monitoring apparatus and method using such a loop sensor*. US Patent 6,337,640. Ene. de 2002.

[162] William Lehr y Lee W McKnight. "Wireless internet access: 3G vs. WiFi?" En: *Telecommunications Policy* 27.5 (2003), págs. 351-370.

[163] James E Lenz. "A review of magnetic sensors". En: *Proceedings of the IEEE* 78.6 (1990), págs. 973-989.

[164] *Ley 11/1998 General de Telecomunicaciones*. URL: https://www.boe.es/buscar/act.php?id=BOE-A-2014-4950.

[165] *Ley Orgánica de 15/1999 de Protección de Datos de Carácter Personal (LOPD)*. URL: https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750.

[166] Yifan Li, Jiawei Han y Jiong Yang. "Clustering moving objects". En: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2004, págs. 617-622.

[167] Chin-Heng Lim, Yahong Wan, Boon-Poh Ng y Chong-Meng Samson See. "A real-time indoor WiFi localization system utilizing smart antennas". En: *IEEE Transactions on Consumer Electronics* 53.2 (2007).

[168] Marco Lippi, Matteo Bertini y Paolo Frasconi. "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning". En: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), págs. 871-882.

[169] Hui Liu, Hong-qi Tian y Yan-fei Li. "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction". En: *Applied Energy* 98 (2012), págs. 415-424.

[170] Ramón López de Lucio. *Ciudad y urbanismo a finales del siglo XX*. Universitat de Valencia, Servicio de Publicaciones, 1993.

[171] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li y Fei-Yue Wang. "Traffic flow prediction with big data: a deep learning approach". En: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015), págs. 865-873.

[172] David JC MacKay. "Bayesian interpolation". En: *Neural computation* 4.3 (1992), págs. 415-447.

[173] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye y Dane Brown. "A study of MAC address randomization in mobile devices and when it fails". En: *Proceedings on Privacy Enhancing Technologies* 2017.4 (2017), págs. 365-383.

[174] Peter T Martin, Yuqi Feng, Xiaodong Wang y col. *Detector technology evaluation*. Inf. téc. Mountain-Plains Consortium, 2003.

[175] Friedemann Mattern. "From smart devices to smart everyday objects". En: *Proceedings of smart objects conference*. 2003, págs. 15-16.

[176] Matthew S. Gast Matthew Gast. *802.11 Wireless Networks: The Definitive Guide (O'Reilly Networking)*. 1.ª ed. O'Reilly Networking. O'Reilly Media, 2002. ISBN: 9780596001834,0596001835. URL: http://gen.lib.rus.ec/book/index.php?md5=4AC045A00275981C1EC4C12E60B41310.

[177] Garth P McCormick. "Exponential Forecasting: Some New Variations". En: *Management Science* 15.5 (1969), págs. 311-320.

[178] Jules G McNeff. "The global positioning system". En: *IEEE Transactions on Microwave theory and techniques* 50.3 (2002), págs. 645-652.

[179] Clark McPhail y John McCarthy. "Who counts and how: estimating the size of protests". En: *Contexts* 3.3 (2004), págs. 12-18.

[180] J.M Menéndez. "Kilómetros a precio de Oro". En: *DGT - Tráfico* Noviembre-Diciembre (2000), págs. 31-34.

[181] Juan-Julián Merelo-Guervós, Israel Blancas-Álvarez, Pedro A Castillo, Gustavo Romero, Pablo García-Sánchez, Victor M Rivas, Mario García-Valdez, Amaury Hernández-Águila y Mario Román. "Ranking Programming Languages for Evolutionary Algorithm Operations". En: *European Conference on the Applications of Evolutionary Computation*. Springer. 2017, págs. 689-704.

[182] Jean-Luc Meunier y Dave Snowdon. *Mobile device and method for determining location of mobile device*. US Patent 7,042,391. Mayo de 2006.

[183] David L Mills. "Internet time synchronization: the network time protocol". En: *IEEE Transactions on communications* 39.10 (1991), págs. 1482-1493.

[184] David L Mills. "Network time protocol version 4 reference and implementation guide". En: *Electrical and Computer Engineering Technical Report* (2006), págs. 06-06.

[185] Esmond Mok y Günther Retscher. "Location determination using WiFi fingerprinting versus WiFi trilateration". En: *Journal of Location Based Services* 1.2 (2007), págs. 145-159.

[186] David Molnar y David Wagner. "Privacy and security in library RFID: Issues, practices, and architectures". En: *Proceedings of the 11th ACM conference on Computer and communications security*. ACM. 2004, págs. 210-219.

[187] Shunji Mori, Hirobumi Nishida e Hiromitsu Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999.

[188] Alistair Morrison, Marek Bell y Matthew Chalmers. "Visualisation of spectator activity at stadium events". En: *Information Visualisation, 2009 13th International Conference*. IEEE. 2009, págs. 219-226.

[189] Hans-Hellmut Nagel y Wilfried Enkelmann. "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (1986), págs. 565-593.

[190] Taewoo Nam y Theresa A Pardo. "Conceptualizing smart city with dimensions of technology, people, and institutions". En: *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times*. ACM. 2011, págs. 282-291.

[191] Nic Newman. "Apple iBeacon technology briefing". En: *Journal of Direct, Data and Digital Marketing Practice* 15.3 (2014), págs. 222-225.

[192] Lionel M Ni, Yunhao Liu, Yiu Cho Lau y Abhishek P Patil. "LANDMARC: indoor location sensing using active RFID". En: *Wireless networks* 10.6 (2004), págs. 701-710.

[193] Tom Nicolai y Holger Kenn. "About the relationship between people and discoverable Bluetooth devices in urban environments". En: *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*. ACM. 2007, págs. 72-78.

[194] Nils J Nilsson. "Probabilistic logic". En: *Artificial intelligence* 28.1 (1986), págs. 71-87.

[195] Sandra Norman-Eady. *The use of Thermal Imaging and the Fourth Amendment*. 2001.

[196] Eurostat Oecd. "Oslo Manual". En: *Guidelines for Collecting and Interpreting Innovation Data*, (2005).

[197] Ajoy K Palit y Dobrivoje Popovic. *Computational intelligence in time series forecasting*. 2006.

[198] JC Palomares-Salas, JJG De La Rosa, JG Ramiro, J Melgar, A Aguera y A Moreno. "ARIMA vs. Neural networks for wind speed forecasting". En: *2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. IEEE. 2009, págs. 129-133.

[199] Gang Pan, Guande Qi, Zhaohui Wu, Daqing Zhang y Shijian Li. "Land-use classification using taxi GPS traces". En: *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013), págs. 113-123.

[200] Gang Pan, Guande Qi, Wangsheng Zhang, Shijian Li, Zhaohui Wu y Laurence Tianruo Yang. "Trace analysis and mining for smart cities: issues, methods, and applications". En: *IEEE Communications Magazine* 51.6 (2013).

[201] Dong C Park, MA El-Sharkawi, RJ Marks, LE Atlas y MJ Damborg. "Electric load forecasting using an artificial neural network". En: *IEEE transactions on Power Systems* 6.2 (1991), págs. 442-449.

[202] E. Parras-Gutierrez, M. Garcia-Arenas, V. Rivas y M. del Jesus. "Coevolution of lags and rbfns for time series forecasting: L-co-r algorithm". En: Soft Computing 16 (6) (2012) 919-942.

[203] E. Parras-Gutierrez, V.M. Rivas, M. Garcia-Arenas y M.J. del Jesus. "Short, medium and long term forecasting of time series using the L-Co-R algorithm". En: *Neurocomputing* 128 (2014), págs. 433-446.

[204] Helen L Partridge. "Developing a human perspective to the digital divide in the'smart city'". En: (2004).

[205] Rahul Patel. *Plug-and-play*. US Patent 5,999,989. Dic. de 1999.

[206] Onkar Pathak, Pratik Palaskar, Rajesh Palkar y Mayur Tawari. "Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points—A Trilateration Approach". En: *International Journal of Scientific & Engineering Research* 5.4 (2014), pág. 1234.

[207] David A. Patterson, Garth Gibson y Randy H. Katz. *A case for redundant arrays of inexpensive disks (RAID)*. 1998.

[208] Peter CB Phillips y Pierre Perron. "Testing for a unit root in time series regression". En: *Biometrika* 75.2 (1988), págs. 335-346.

[209] Louis J Pignataro, Edmund J Cantilli, John C Falcocchio, KW Crowley, WR McShane, RP Roess y B Lee. *Traffic engineering: theory and practice*. Inf. téc. 1900.

[210] Jean-Christophe Plantin, Carl Lagoze, Paul Edwards y Christian Sandvig. "Big data is not about size: When data transform scholarship". En: (2017).

[211] John Platt. "Sequential minimal optimization: A fast algorithm for training support vector machines". En: (1998).

[212] RS Popovic, JA Flanagan y PA Besse. "The future of magnetic sensors". En: *Sensors and actuators A: Physical* 56.1 (1996), págs. 39-55.

[213] *Popular Mechanics*. Nov. de 1949. URL: https://books.google.es/books?id=WdkDAAAAMBAJ&hl=es&source=gbs_all_issues_r&cad=1.

[214] Stefan Poslad. *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons, 2011.

[215] Jacob Poushter. "Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies". En: *Pew Research Center: Global Attitudes & Trends* (2016).

[216] *Presupuestos ayuntamiento de Granada*. 2016. URL: transparencia.granada.org/public/Documento.aspx?ID=3092.

[217] Chen Qiu y Matt W Mutka. "Cooperation among smartphones to improve indoor position information". En: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*. IEEE. 2015, págs. 1-9.

[218] Roya Rad y Mansour Jamzad. "Real time classification and tracking of multiple vehicles in highways". En: *Pattern Recognition Letters* 26.10 (2005), págs. 1597-1607.

[219] *Raspberry Pi Fundation*. https://www.raspberrypi.org.

[220] *Real Decreto de 14 de Septiembre de 1882*. URL: https://www.boe.es/buscar/pdf/1882/BOE-A-1882-6036-consolidado.pdf.

[221] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen y Mani Srivastava. "Using mobile phones to determine transportation modes". En: *ACM Transactions on Sensor Networks (TOSN)* 6.2 (2010), pág. 13.

[222] John R Reitz, Frederick J Milford y Robert W Christy. *Foundations of electromagnetic theory*. Addison-Wesley Publishing Company, 2008.

[223] Julian Reschke. *RFC 7617 - The'Basic'HTTP Authentication Scheme*. Inf. téc. 2015. URL: https://tools.ietf.org/html/rfc7617.

[224] Eric Rescorla. *RFC 2818 - Http over tls*. Inf. téc. 2000. URL: https://tools.ietf.org/html/rfc2818.

[225] Eric Rescorla y A Schiffman. *RFC 2660 - The secure hypertext transfer protocol*. Inf. téc. 1999. URL: https://tools.ietf.org/html/rfc2660.

[226] Trond Riise y Dag Tjozstheim. "Theory and practice of multivariate ARMA forecasting". En: *Journal of Forecasting* 3.3 (1984), págs. 309-317.

[227] Patrice Rios. "Creating"The Smart City"". Tesis doct. 2012.

[228] Chris M Roberts. "Radio frequency identification (RFID)". En: *Computers & security* 25.1 (2006), págs. 18-26.

[229] Darren Robinson. *Computer modelling for sustainable urban design: Physical principles, methods and applications*. Routledge, 2012.

[230] Jean-Paul Rodrigue, Claude Comtois y Brian Slack. *The geography of transport systems*. Routledge, 2013.

[231] Bernard Rosner. "Percentage points for a generalized ESD many-outlier procedure". En: *Technometrics* 25.2 (1983), págs. 165-172.

[232] Jeanne W Ross, Cynthia M Beath y Anne Quaadgras. "You may not need big data after all". En: *Harvard Business Review* 91.12 (2013), págs. 90-+.

[233] "RPi SD cards". En: (2017). URL: https://elinux.org/RPi_SD_cards#SD_card_performance.

[234] Manuel de Solà-Morales i Rubió. *Las formas de crecimiento urbano*. Vol. 10. Univ. Politèc. de Catalunya, 1997.

[235] David E Rumelhart, Geoffrey E Hinton y Ronald J Williams. *Learning internal representations by error propagation*. Inf. téc. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[236] John Sacco y Brett Greenky. *RFID tracking of anesthesiologist and patient time*. US Patent App. 10/752,070. Ene. de 2004.

[237] Arthur L Samuel. "Some studies in machine learning using the game of checkers". En: *IBM Journal of research and development* 3.3 (1959), págs. 210-229.

[238] *SavariNetworks*. http://www.SavariNetworks.com/. [Online; accessed 28-Nov-2015].

[239] Paul-Andre Roland Savoie y Andre Eric Boulay. *Vehicle tracking system using cellular network*. US Patent 5,895,436. 1999.

[240] Robert Schneider, Robert Patten y Jennifer Toole. "Case study analysis of pedestrian and bicycle data collection in US communities". En: *Transportation Research Record: Journal of the Transportation Research Board* 1939 (2005), págs. 77-90.

[241] A.J. Schofield, P.A. Mehta y T.J. Stonham. "A system for counting people in video images using neural networks to identify the background scene". En: *Pattern Recognition* 29.8 (1996), págs. 1421-1428.

[242] Baron Schwartz y Preetam Jinka. *Anomaly Detection for Monitoring*. O'Relly Media Inc, 2016. ISBN: ISBN: 9781492042341.

[243] Baron Schwartz, Peter Zaitsev y Vadim Tkachenko. *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly.

[244] Gildo Seisdedos, Borda Richart, Gema Gallego, Javier de Paz, José Esponera y Olga Kolotouchkina. *Smart Cities: La transformación digital de las ciudades*. Telefónica y PwC, 2015.

[245] V Series. "Data Communication Over the Telephone Network". En: *Interfaces and voiceband modems* (1996).

[246] Eliezer A Sheffer y Marco J Thompson. *Vehicle tracking system*. US Patent 5,218,367. Jun. de 1993.

[247] William Simpson. "The point-to-point protocol (PPP) for the transmission of multi-protocol datagrams over point-to-point links". En: (1992).

[248] Sherry L Skszek. *"State-of-the-ArtReport on Non-Traditional Traffic Counting Methods*. Inf. téc. Arizona Department of Transportation, 2001.

[249] Alex J Smola y Bernhard Schölkopf. "A tutorial on support vector regression". En: *Statistics and computing* 14.3 (2004), págs. 199-222.

[250] Peter Snyder. "tmpfs: A virtual memory file system". En: *Proceedings of the Autumn 1990 EUUG Conference*. 1990, págs. 241-248.

[251] IEEE Computer Society. *802.3 Ethernet Working Group*. URL: http://www.ieee802.org/3/.

[252] IEEE Computer Society. *Bluetooth Assigned Numbers - Baseband*. URL: https://www.bluetooth.com/specifications/assigned-numbers/baseband.

[253] IEEE Computer Society. *802.11-1999 - Standard for Information Technology*. 1999. URL: http://standards.ieee.org/findstds/standard/802.11-1999.html.

[254] IEEE Computer Society. *521-2002 - IEEE Standard Letter Designations for Radar-Frequency Bands*. 2002. URL: https://standards.ieee.org/findstds/standard/521-2002.html.

[255] IEEE Computer Society. *802.15.1-2002 - IEEE Standard for Telecommunications and Information Exchange Between Systems*. 2002. URL: http://standards.ieee.org/findstds/standard/802.15.1-2002.html.

[256] IEEE Computer Society. *802-2014 - IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*. 2014. URL: http://standards.ieee.org/findstds/standard/802-2014.html.

[257] Chaoming Song, Zehui Qu, Nicholas Blumm y Albert-László Barabási. "Limits of predictability in human mobility". En: *Science* 327.5968 (2010), págs. 1018-1021.

[258] M. Soyturk, M.C. Bodur, A.B. Bakkal y S. Ozturk. "Estimating the number of people in a particular area using WiFi". En: *Signal Processing and Communications Applications Conference (SIU), 2015 23th*. Mayo de 2015, págs. 2541-2544.

[259] William J Stanton. "Fundamentals of marketing". En: (1967).

[260] Rainer Steffen, Jörg Preißinger, Tobias Schöllermann, Armin Müller e Ingo Schnabel. "Near field communication (NFC) in an automotive environment". En: *Near Field Communication (NFC), 2010 Second International Workshop on*. IEEE. 2010, págs. 15-20.

[261] Gordon L Stüber. *Principles of mobile communication*. Springer Science & Business Media, 2011.

[262] Johan AK Suykens y Joos Vandewalle. "Least squares support vector machine classifiers". En: *Neural processing letters* 9.3 (1999), págs. 293-300.

[263] Kingston Technology. *Flash Memory Guide: Portable Flash memory for computers, digital cameras, mobile phones and other devices*. 2015. URL: https://media.kingston.com/pdfs/MKF_283.1_Flash_Memory_Guide_EN.pdf.

[264] Telefónica. *Informe Anual sobre la Sociedad de la Información en España*. http://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/itempubli/483/. 2015.

[265] *The Computer Language Benchmarks Game*. 2017. URL: https://benchmarksgame.alioth.debian.org/u64q/java.html.

[266] *Thomas J. Watson*. Presidente de la Junta Directiva de IBM. 1943.

[267] TierpointLLC. *Trafficnow*. http://www.trafficnow.com/.

[268] *TrafficCast*. http://www.TrafficCast.com/.

[269] Herbert Alker Tripp. *Road traffic and its control*. Vol. 7. Arnold, 1950.

[270] Ruey S Tsay. "Time series and forecasting: Brief history and future research". En: *Journal of the American Statistical Association* 95.450 (2000), págs. 638-643.

[271] John R Tuttle. *Anti-theft method for detecting the unauthorized opening of containers and baggage*. US Patent 5,406,263. Abr. de 1995.

[272] S. Ultsch. "Kohonen's Self-organizing maps for exploratory data analysis." En: *INNC'90*. IASTED IMCAI. Berlin: Kluwer Academic, 2000, págs. 305-308.

[273] Eben Upton y Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[274] *Urban Audit*. http://ec.europa.eu/regional_policy/en/policy/themes/urban-development/audit/.

[275] Jose I Uriol Salcedo. *Historia de los caminos de España*. Colegio de Caminos, Canales y Puertos, 2001.

[276] Owen Vallis, Jordan Hochenbaum y Arun Kejariwal. "A Novel Technique for Long-Term Anomaly Detection in the Cloud." En: *HotCloud*. 2014.

[277] C Van Der Malsburg. "Frank Rosenblatt: principles of neurodynamics: perceptrons and the theory of brain mechanisms". En: *Brain theory*. Springer, 1986, págs. 245-248.

[278] Lelitha Vanajakshi y Laurence R Rilett. "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed". En: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE. 2004, págs. 194-199.

[279] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S Cardoso y Frank Piessens. "Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms". En: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM. 2016, págs. 413-424.

[280] Vladimir Vapnik, Isabel Guyon y Trevor Hastie. "Support vector machines". En: *Mach. Learn* 20.3 (1995), págs. 273-297.

[281] Rafael G Vieira, Marcos A Leone Filho y Robinson Semolini. "An Enhanced Seasonal-Hybrid ESD Technique for Robust Anomaly Detection on Time Series". En: *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC. 2018.

[282] Ignasi Vilajosana. *BitCarrier. Go With the Flow*. http://www.bitcarrier.com/.

[283] Athanasios S Voulodimos, Charalampos Z Patrikakis, Alexander B Sideridis, Vasileios A Ntafis y Eftychia M Xylouri. "A complete farm management system based on animal identification using RFID technology". En: *Computers and Electronics in Agriculture* 70.2 (2010), págs. 380-388.

[284] Vladimir Vujovic y Mirjana Maksimovic. "Raspberry Pi as a wireless sensor node: performances and constraints". En: *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE. 2014, págs. 1013-1018.

[285] Yiyan Wang, Yuexian Zou, Hang Shi y He Zhao. "Video image vehicle detection system for signaled traffic intersection". En: *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*. Vol. 1. IEEE. 2009, págs. 222-227.

[286] Roy Want. "An introduction to RFID technology". En: *IEEE pervasive computing* 5.1 (2006), págs. 25-33.

[287] Brett Warneke, Matt Last, Brian Liebowitz y Kristofer SJ Pister. "Smart dust: Communicating with a cubic-millimeter computer". En: *Computer* 34.1 (2001), págs. 44-51.

[288] Doug Washburn, Usman Sindhu, Stephanie Balaouras, Rachel A Dines, N Hayes y Lauren E Nelson. "Helping CIOs understand "smart city" initiatives". En: *Growth* 17.2 (2009), págs. 1-17.

[289] Mark Weiser. "The computer for the 21st century". En: *Scientific american* 265.3 (1991), págs. 94-104.

[290] Mark Weiser. "Some computer science issues in ubiquitous computing". En: *Communications of the ACM* 36.7 (1993), págs. 75-84.

[291] Mark Weiser. "Ubiquitous computing". En: *ACM Conference on Computer Science*. 1994, pág. 418.

[292] Mark Weiser. "The Computer for the 21st Century". En: *SIGMOBILE Mob. Comput. Commun. Rev.* 3.3 (jul. de 1999), págs. 3-11. ISSN: 1559-1662. DOI: 10.1145/329124.329126. URL: http://doi.acm.org/10.1145/329124.329126.

[293] Mark Weiser, Rich Gold y John Seely Brown. "The origins of ubiquitous computing research at PARC in the late 1980s". En: *IBM systems journal* 38.4 (1999), págs. 693-696.

[294] Sholom M Weiss y Casimir A Kulikowski. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Vol. 123. Morgan Kaufmann San Mateo, CA, 1991.

[295] J. Weppner y P. Lukowicz. "Bluetooth based collaborative crowd density estimation with mobile phones". En: *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*. Mar. de 2013, págs. 193-200.

[296] Paul Werbos. "Beyond regression: New tools for prediction and analysis in the behavioral sciences". En: (1974).

[297] Billy M Williams y Lester A Hoel. "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results". En: *Journal of transportation engineering* 129.6 (2003), págs. 664-672.

[298] Peter R Winters. "Forecasting sales by exponentially weighted moving averages". En: *Management science* 6.3 (1960), págs. 324-342.

[299] Hsien-Chung Wu. "The Karush–Kuhn–Tucker optimality conditions in an optimization problem with interval-valued objective function". En: *European Journal of Operational Research* 176.1 (2007), págs. 46-59.

[300] Wei Xi, Jizhong Zhao, Xiang-Yang Li, Kun Zhao, Shaojie Tang, Xue Liu y Zhiping Jiang. "Electronic frog eye: Counting crowd using WiFi". En: *INFOCOM, 2014 Proceedings IEEE*. Abr. de 2014, págs. 361-369.

[301] Robert Xiao, Gierad Laput, Yang Zhang y Chris Harrison. "Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances". En: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, págs. 4000-4008.

[302] Ming Xie, Laurent Trassoudaine, Joseph Alizon y Jean Gallice. "Road obstacle detection and tracking by an active and intelligent sensing strategy". En: *Machine Vision and Applications* 7.3 (1994), págs. 165-177.

[303] Youngjin Yoo. "It is not about size: a further thought on big data". En: *Journal of Information Technology* 30.1 (2015), págs. 63-65.

[304] Jing Yuan, Yu Zheng, Xing Xie y Guangzhong Sun. "T-drive: Enhancing driving directions with taxi drivers' intelligence". En: *IEEE Transactions on Knowledge and Data Engineering* 25.1 (2013), págs. 220-232.

[305] George Udny Yule. "VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers". En: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226.636-646 (1927), págs. 267-298.

[306] Kalid Yunus, Torbjörn Thiringer y Peiyuan Chen. "ARIMA-based frequency-decomposed modeling of wind speed time series". En: *IEEE Transactions on Power Systems* 31.4 (2016), págs. 2546-2556.

[307] Giancarlo Zaccone. *Getting Started with TensorFlow*. Packt Publishing Ltd, 2016.

[308] Lofti A Zadeh. "Fuzzy logic, neural networks, and soft computing". En: *Communications of the ACM* 37.3 (1994), págs. 77-85.

[309] Lotfi A Zadeh. "Fuzzy sets". En: *Information and control* 8.3 (1965), págs. 338-353.

[310] Lotfi A Zadeh. "What is soft computing?" En: *Soft computing* 1.1 (1997), págs. 1-1.

[311] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun y Shijian Li. "iBAT: detecting anomalous taxi trajectories from GPS traces". En: *Proceedings of the 13th international conference on Ubiquitous computing*. ACM. 2011, págs. 99-108.

[312] Guoqiang Zhang, B Eddy Patuwo y Michael Y Hu. "Forecasting with artificial neural networks:: The state of the art". En: *International journal of forecasting* 14.1 (1998), págs. 35-62.

[313] Yang Zhang y Yuncai Liu. "Traffic forecasting using least squares support vector machines". En: *Transportmetrica* 5.3 (2009), págs. 193-213.

[314] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips y Azriel Rosenfeld. "Face recognition: A literature survey". En: *ACM computing surveys (CSUR)* 35.4 (2003), págs. 399-458.

[315] Yang Zhao-sheng, Wang Yuan y Guan Qing. "Short-term traffic flow prediction method based on SVM [J]". En: *Journal of Jilin University (Engineering and Technology Edition)* 6 (2006), pág. 009.

[316] Xiuyan Zhu y Yuan Feng. "RSSI-based algorithm for indoor localization". En: *Communications and Network* 5.02 (2013), pág. 37.

[317] Andrew Zola. *12 Popular Programming Languages for IOT Development*. 2017. URL: https://www.upwork.com/hiring/for-clients/programming-languages-for-iot-development/.