

UNIVERSIDAD DE GRANADA

Departamento de Lenguajes y Sistemas
Informáticos



Programa Oficial de Doctorado en Tecnologías de la
Información y la Comunicación

TESIS DOCTORAL

**Hacia el Internet de los Agentes: Un Modelo
Semántico Para Agentificar Ecosistemas de
Internet de las Cosas**

Realizado por:

Pablo Antonio PICO VALENCIA

Dirigida por:

Juan Antonio HOLGADO TERRIZA

Octubre, 2018

Editor: Universidad de Granada. Tesis Doctorales
Autor: Pablo Antonio Pico Valencia
ISBN: 978-84-1306-107-8
URI: <http://hdl.handle.net/10481/71706>

A mi madre, por ser el pilar fundamental en todas las etapas de mi vida.

Resumen

El Internet de las Cosas (Internet of Things, IoT por sus siglas en inglés) es uno de los paradigmas emergentes que está revolucionando el Internet tanto a corto como a largo plazo, estableciendo las bases de lo que será el Futuro Internet. Dentro de dicho contexto, y tomando como base el IoT, se han identificado cuatro paradigmas complementarios que están contribuyendo a consolidar el IoT y, en consecuencia, el Futuro Internet. Estos paradigmas son: el Internet de los Servicios (IoS), Internet de las Personas (IoP), Internet del Contenido/Recursos (IoC) y el Internet de los Agentes (IoA).

Entre dichos paradigmas, el IoA se posiciona como un novedoso paradigma que puede aportar capacidad de inteligencia al IoT. Para ello, este enfoque introduce el concepto de agente software como mecanismo para la construcción de ecosistemas inteligentes de IoT donde los objetos tradicionalmente pasivos puedan llegar a comportarse de manera proactiva e inteligente.

Este trabajo de tesis presenta las contribuciones científicas que se han desarrollado dentro del contexto del IoA. De manera general, se definieron las directrices, concepto y terminología, alcance, aplicaciones potenciales, una especificación conceptual, métodos de agentificación, un marco de trabajo, un instrumento metodológico de diseño y un modelo de agente específico para proveer de inteligencia, proactividad, sociabilidad, adaptabilidad, y capacidad de colaboración al IoT.

En primer lugar se identificaron las bases en las que se sustenta el IoA, y la relación que debe tener con el resto de paradigmas de IoT. Esto nos ha permitido definir formalmente el concepto de agente software.

Posteriormente se realizó un estudio sistemático de los métodos planteados hasta la fecha para modelar el IoT a partir de agentes software. A partir de este estudio se planteó el “proceso de agentificación” como el procedimiento que permite integrar los agentes en los ecosistemas de IoT y que se puede aplicar de dos formas distintas: embebiendo agentes sobre los objetos y gestionando los recursos de redes de IoT a través de sistemas multiagentes. El modelo de cognición basado en reglas resultó ser el más empleado en la creación de ecosistemas de IoT agentificados, luego de compararlo con otros mecanismos como los basados en componentes

deliberativos, lógica semántica, probabilidades y técnicas de Inteligencia Artificial.

El agente deben que tener una identidad propia, y ser autónomo y proactivo, lo que le permite coexistir, así como compartir o colaborar con otros agentes del ecosistema de IoA. Las directrices que determinan las interacciones entre agentes en IoA se resuelven mediante procesos de razonamiento semántico en base a una descripción semántica de cada agente. Se creó una ontología denominado IoA-OWL para recoger e integrar todos los conceptos requeridos en la descripción semántica de cada agente organizados en seis sub-ontologías a las que se denominó: *AgentProfile*, *AgentContext*, *AgentService*, *AgentModel*, *AgentSmartThing* y *AgentSocial*, es decir, un perfil de agente, sensibilidad al contexto, ámbito social, ecosistema de servicios, artefactos de agentes y recursos de IoT, respectivamente.

En base al IoA-OWL se desarrolló un nuevo tipo de agente, el agente *Linked Open Agent* (LOA), extendiendo el modelo de agente JADE (*Java Agent DEvelopment Framework*). El modelo de agente propuesto es compatible con la agentificación del IoT a través de sistemas multiagentes y está descrito mediante un contrato semántico (*Linked Agent Contract*, LAC) y un flujo de control (*Workflow for Agent Control*, WAC). El primero de ellos se construye a partir de descriptores semánticos de acuerdo a la ontología IoA-OWL y el segundo se construye a partir de un conjunto de bloques IFTTT (If-This Then-That) donde se definen las instancias de los agentes contrapartes que el LOA empleará para la ejecución de tareas colaborativas.

A nivel experimental, el proceso de agentificación del IoT modelado a partir de entidades LOAs y guiado por el marco de trabajo Framework-IoA —desarrollado mediante la herramienta IoA-Building Tool— proporcionó un mejor grado de precisión y exactitud en la recuperación de contrapartes en comparación con los mecanismos de descubrimiento sintácticos de agentes implementados mediante JADE y de repositorio de servicios implementados en jUDDI (*Java implementation for the Universal Description, Discovery, and Integration*). Con respecto a JADE y jUDDI, el modelo basado en LOAs proporcionó una mejora en la precisión del 24 %.

Por otro lado, en vista de que el enfoque propuesto no correspondía

a un enfoque convencional fue necesario definir una herramienta metodológica que guiara el proceso a seguir para agentificar un ecosistema de IoT. En este sentido, se desarrolló un método sistemático en el que el proceso de agentificación es abordado en dos niveles, microscópico (a nivel de LOA) y macroscópico (a nivel de sociedad). Cada nivel incluyó un grupo de etapas, fases y pasos lógicos fáciles de ser ejercidos y que cumplidos a cabalidad permiten la creación de ecosistemas de IoA.

Finalmente, el modelo de entidad LOA así como el método sistemático propuesto fueron evaluados en el área de Inteligencia Ambiental. Se plantearon dos casos de estudio orientados a proveer confort en comunidades y oficinas inteligentes. En ambos casos los sistemas modelados se comportaron de acuerdo a lo planificado. Las entidades LOAs de los ecosistemas de IoA lograron gestionar los recursos de IoT de manera inteligente, adaptativa, pro-activa y colaborativa. Asimismo, el modelo basado en LOAs proporcionó un mejor nivel de interoperabilidad, adaptación, recuperación ante fallos, proactividad y flexibilidad a nivel de agente en comparación con el modelo genérico de agentes JADE.

Agradecimientos

El desarrollo de esta Tesis Doctoral fue posible gracias a la *Secretaría de Educación Superior, Ciencia, Tecnología e Innovación del Ecuador (SENESCYT)*, que con el Programa de Doctorado para Docentes Universitarios me brindó la oportunidad de cursar los estudios de postgrado en un centro de educación de prestigio internacional como es la Universidad de Granada. De igual manera, agradezco a la *Pontificia Universidad Católica del Ecuador Sede Esmeraldas (PUCESE)*, institución que me auspició para iniciar el proceso de postulación y adjudicación de la beca SENESCYT. También agradezco el apoyo del MSc. Xavier Quiñónez, Director de la Escuela de Sistemas y Computación de la misma institución, quien gestionó algunos de los recursos que me permitieron participar en varios eventos académicos internacionales especializados en la línea de investigación en la que se enmarcó este trabajo.

Por otro lado, agradezco el apoyo incondicional de mi supervisor, el Dr. *Juan Holgado Terriza*, quien mostró interés en mi propuesta de tesis doctoral desde el primer contacto realizado por medios digitales. Sus consejos y experiencia coadyuvaron para el desarrollo de las diferentes destrezas que he adquirido durante el proceso de formación, ejecución y difusión de resultados de la investigación realizada, tanto en revistas de alto impacto como en conferencias especializadas relacionadas con la temática abordada en esta investigación. Así mismo, mis agradecimientos especiales al Dr. *José Senso* y a la Dra. *Patricia Paderewsky*, profesores de la Universidad de Granada, quienes enriquecieron con sus críticas constructivas dos de las propuestas planteadas, que a su vez generaron dos de los artículos enviados a medios científicos de alto impacto.

Adicionalmente, agradezco infinitamente al Dr. *Sarvapali Ramchurn* quien me dio la oportunidad de realizar una estancia de investigación y ser parte del Grupo de Agentes, Interacción y Complejidad de la Universidad de Southampton, del Reino Unido. Finalmente, agradezco a la *Escuela Internacional de Postgrado de la Universidad de Granada*, institución española de prestigio y calidad, que con sus propuestas de cursos especializados me han permitido abordar y culminar con éxito este proceso de formación a nivel doctoral y enfocarla desde diferentes aristas—científico y personal—a fin de que pueda aportar a la sociedad en general desde la academia, la ciencia y el emprendimiento.

Índice general

Resumen	VII
Agradecimientos	XI
Introducción	1
Antecedentes y motivación	1
Objetivos	5
Estructura	5
1. Perspectivas actuales y futuras del Internet	9
1.1. Evolución del Internet	10
1.2. Paradigmas relacionados	15
1.3. Internet de las Cosas	16
1.4. Internet de los Servicios	21
1.5. Internet de las Personas	24
1.6. Internet del Contenido/Recursos	27
1.7. Internet de los Agentes	33
1.8. Discusión	37
1.9. Contribuciones relacionadas	38
2. Agentificación del IoT: Una revisión de la literatura	39
2.1. Revisión sistemática	40
2.1.1. Definición de las preguntas de investigación	40
2.1.2. Selección de las fuentes de información	41
2.1.3. Selección de estudios	42
2.1.4. Resultados	43
2.2. Modelado de objetos con agentes	45
2.2.1. Agente embebido en objetos de IoT	47
2.2.2. Agentes web orientados a objetos de IoT	49
2.2.3. Agentes móviles embebidos en objetos de IoT	50

2.3.	Modelado de redes de IoT con MASs	51
2.4.	Agentificación del IoT en la práctica	55
2.5.	Modelos de cognición para agentificar el IoT	55
2.5.1.	Basada en reglas	57
2.5.2.	Basada en componentes deliberativos	58
2.5.3.	Basada en lógica semántica	58
2.5.4.	Basada en probabilidades	59
2.5.5.	Basada en Inteligencia Artificial	60
2.6.	Tendencias de la agentificación del IoT	61
2.6.1.	Modelos de agentes ligeros para IoT	61
2.6.2.	Objetos de IoT <i>open hardware</i>	62
2.6.3.	Agentes amigables con el usuario	62
2.6.4.	Plataformas de agentes interoperables	63
2.6.5.	Servicios basados en la nube	63
2.6.6.	Metodologías especializadas	63
2.6.7.	Software especializado	64
2.6.8.	Repositorios de servicios públicos	64
2.6.9.	Definición de patrones y anti-patrones	65
2.6.10.	Descentralización de la gestión de datos	65
2.6.11.	Mecanismos de seguridad y privacidad	65
2.7.	Discusión	66
2.8.	Contribuciones relacionadas	67
3.	Internet de los Agentes: Un enfoque semántico	69
3.1.	Internet de los Agentes	70
3.2.	Un enfoque semántico del IoA	72
3.3.	Modelando el IoA	73
3.3.1.	Principales aspectos	74
3.3.2.	Información del perfil de agente	76
3.3.3.	Información de sensibilidad al contexto	77
3.3.4.	Información del ámbito social	78
3.3.5.	Información del ecosistema de servicios	79
3.3.6.	Información de los artefactos de agentes	79
3.3.7.	Información de los recursos de IoT	80
3.4.	Potenciales aplicaciones	81
3.4.1.	Ciudades inteligentes	81
3.4.2.	Cuidado de la salud	82
3.5.	Discusión	84

3.6. Contribuciones relacionadas	84
4. IoA-OWL, una ontología para describir el IoA	85
4.1. Modelos semánticos para IoT	86
4.1.1. Ontologías relacionadas con el IoA	86
4.1.2. Vocabularios enlazados y reutilizados	89
4.1.3. Ontologías reutilizadas	90
4.2. Ontología IoA-OWL	91
4.2.1. Ontología de nivel superior para IoA	92
4.2.2. Componente <i>AgentProfile</i>	94
4.2.3. Componente <i>AgentContext</i>	95
4.2.4. Componente <i>AgentService</i>	97
4.2.5. Componente <i>AgentModel</i>	99
4.2.6. Componente <i>AgentSmartThing</i>	100
4.2.7. Componente <i>AgentSocial</i>	101
4.3. Discusión	103
4.4. Contribuciones relacionadas	104
5. Framework-IoA: Un marco de trabajo para agentificar el IoT	105
5.1. IoA basado en datos enlazados	106
5.2. Linked Open Agent	107
5.2.1. Arquitectura de agente	110
5.2.2. Particiones de agentes	113
5.3. Linked Agent Contract	114
5.3.1. Estructura general del contrato	116
5.3.2. Estructura de las secciones contractuales	118
5.4. Workflow for Agent Control	119
5.4.1. Recetas IFTTT orientadas a objetos de IoT	120
5.4.2. Estructura de un flujo de control	122
5.5. Arquitectura de referencia para IoA	124
5.6. Procesos y mecanismos del modelo	127
5.6.1. Proceso de construcción de LOAs	127
5.6.2. Proceso de descubrimiento semántico	128
5.7. Calidad de servicio del modelo	132
5.7.1. Mecanismos de calidad de servicio	132
5.7.2. Mecanismos de tolerancia a fallos	135
5.8. Beneficios potenciales del modelo	136

5.9. Limitaciones del modelo	139
5.10. Discusión	140
5.11. Contribuciones relacionadas	141
6. Método sistemático para agentificar el IoT	143
6.1. Visión general del método	144
6.1.1. Método sistemático	144
6.1.2. Proceso	146
6.2. Visión microscópica	148
6.2.1. Planificación de los recursos del objeto de IoT (MI-1)	148
6.2.2. Descubrimiento de agentes públicos especializados asociados al objeto de IoT (MI-2)	150
6.2.3. Planificación de ecosistema de servicios del objeto de IoT (MI-3)	153
6.2.4. Planificación de la arquitectura y el comportamien- to del LOA (MI-4)	156
6.2.5. Enriquecimiento semántico del LAC (MI-5)	161
6.2.6. Generación del LAC del LOA (MI-6)	162
6.3. Visión macroscópica	163
6.3.1. Planificación de las particiones de agentes 1, 2 y 3 (MA-7)	164
6.3.2. Planificación social del ecosistema global (MA-8)	164
6.3.3. Planificación del comportamiento del ecosistema de IoA (MA-9)	167
6.3.4. Validación de los elementos del LOA (MA-10)	169
6.3.5. Ejecución del proceso de modelado basado en con- tratos (MA-11)	171
6.4. Beneficios del método	173
6.5. Discusión	174
6.6. Contribuciones relacionadas	174
7. Evaluación del modelo de IoA basado en LOAs	177
7.1. Definición del escenario de IoT	178
7.2. Despliegue de la infraestructura IoA	179
7.3. Proceso	181
7.4. Ecosistema de IoA: Comunidad inteligente	181
7.4.1. LOAs no colaborativos	184
7.4.2. LOAs colaborativos	184

7.5.	Evaluación experimental	187
7.5.1.	Desempeño de los LOAs	188
7.5.2.	Desempeño del proceso de descubrimiento	191
7.5.3.	Precisión y exactitud del proceso de descubrimiento	195
7.6.	Discusión	198
7.7.	Contribuciones relacionadas	199
8.	Aplicación del método sistemático de agentificación del IoT	201
8.1.	Descripción del escenario de IoT	202
8.2.	Despliegue de la infraestructura de IoA	202
8.3.	Modelado a nivel microscópico	202
8.3.1.	Planificación de recursos del objeto de IoT (MI-1)	203
8.3.2.	Descubrimiento de agentes públicos especializados asociados al objeto de IoT (MI-2)	205
8.3.3.	Planificación del ecosistema de servicios del objeto de IoT (MI-3)	205
8.3.4.	Planificación de la arquitectura y el comportamiento del LOA (MI-4)	206
8.3.5.	Enriquecimiento semántico del LAC (MI-5)	208
8.3.6.	Generación del LAC del LOA (MI-6)	209
8.4.	Modelado a nivel macroscópico	209
8.4.1.	Planificación de las particiones de agentes 1, 2 y 3 (MA-7)	210
8.4.2.	Planificación social del ecosistema global (MA-8)	212
8.4.3.	Planificación del comportamiento del ecosistema de IoA (MA-9)	212
8.4.4.	Validación de los elementos del LOA (MA-10)	214
8.4.5.	Ejecución del proceso de modelado basado en contratos (MA-11)	215
8.5.	Evaluación cualitativa del sistema	215
8.5.1.	Formulación del instrumento de evaluación	215
8.5.2.	Aplicación del instrumento de evaluación propuesto	216
8.6.	Discusión	218
8.7.	Contribuciones relacionadas	219

9. Conclusiones y trabajos científicos futuros	221
9.1. Conclusiones	222
9.2. Trabajos futuros	225
9.3. Contribuciones científicas derivadas	226
9.3.1. Revistas indexadas con impacto JCR	226
9.3.2. Revistas indexadas sin impacto JCR	229
9.3.3. Publicaciones en congresos internacionales	230
9.3.4. Publicaciones en congresos nacionales	232
9.3.5. Publicaciones en preparación	233
Bibliografía	249

Índice de figuras

1.1. Número de internautas en todo el mundo de 2005 a 2017 según datos del Portal Statista.	11
1.2. Número de usuarios de teléfonos inteligentes a nivel mundial entre 2014 y 2020 según proyecciones del Portal Statista.	12
1.3. Objetos cotidianos interconectados a Internet en un escenario de IoT.	13
1.4. Principales paradigmas y entidades emergentes relacionados con el Futuro Internet.	16
1.5. Tendencia de aplicaciones de IoT según el Portal IoT Analytics.	17
1.6. Dispositivos de IoT conectados alrededor del mundo de 2015 a 2025 según el Portal Statista.	18
1.7. Arquitectura de referencia de IoT según Fremantle.	19
1.8. Principales entidades de una Arquitectura Orientada a Servicios.	23
1.9. Los usuarios y su rol en el Internet de las Personas.	25
1.10. Evolución de la web y herramientas usadas en Internet.	28
1.11. Rol de la Computación en la Nube en IoT.	30
1.12. Big Data generada en entornos de IoT.	32
1.13. Usos de la Inteligencia Artificial en IoT.	33
1.14. Siguiete generación del IoT.	35
2.1. Tendencias en el proceso de agentificación del IoT	43
2.2. Naturaleza de los estudios (izquierda) y tipos de publicaciones (derecha) que abordan el proceso de integración de IoT y agentes	44
3.1. Áreas que dan origen al IoA.	72
3.2. Áreas que dan origen al IoA semántico	73
3.3. Aspectos asociados con el IoA.	74

4.1.	Ontología superior y componentes ontológicos de IoA-OWL	93
4.2.	Gráfico ontológico del componente <i>AgentProfile</i>	94
4.3.	Gráfico ontológico del componente <i>AgentContext</i>	96
4.4.	Gráfico ontológico del componente <i>AgentService</i>	98
4.5.	Gráfico ontológico del componente <i>AgentModel</i>	100
4.6.	Gráfico ontológico del componente <i>AgentSmartThing</i>	101
4.7.	Gráfico ontológico del componente <i>AgentSocial</i>	102
5.1.	Ecosistema de IoA dirigido por entidades LOAs.	106
5.2.	Proceso, y principales mecanismos y herramientas para modelar el IoA basado en datos enlazados.	107
5.3.	Arquitectura de un LOA.	111
5.4.	Especificación UML de una micro-receta IFTTT simple y anidada orientada a objetos de IoT.	121
5.5.	Especificación UML de una macro-receta IFTTT orientada a objeto de IoT.	122
5.6.	Arquitectura de referencia propuesta para el IoA	125
5.7.	Diagrama de secuencia UML para la creación de un LOA.	127
5.8.	Algoritmo de descubrimiento ejecutado por los LOAs.	131
6.1.	Esquema general del método propuesto.	145
6.2.	Visión general del proceso seguido por el método.	147
6.3.	Flujo, insumos y resultados de la etapa MI-1.	149
6.4.	Flujo, insumos y resultados de la etapa MI-2.	151
6.5.	Flujo, insumos y resultados de la etapa MI-3.	154
6.6.	Flujo, insumos y resultados de la etapa MI-4.	156
6.7.	Flujo, insumos y resultados de la etapa MI-5.	162
6.8.	Flujo, insumos y resultados de la etapa MI-6.	163
6.9.	Flujo, insumos y resultados de la etapa MA-8.	165
6.10.	Flujo, insumos y resultados de la etapa MA-9.	167
6.11.	Flujo, insumos y resultados de la etapa MA-10.	169
6.12.	Flujo, insumos y resultados de la etapa MA-11.	171
7.1.	Diagrama general del escenario de IoT.	179
7.2.	Descriptorios usados para el enriquecimiento semántico de los LOAs del escenario modelado.	182
7.3.	Esquema del ecosistema de IoA.	183
7.4.	Tiempos requeridos para el procesamiento de un LAC	189

7.5.	Tiempos requeridos para el procesamiento de tripletas RDF.	190
7.6.	Tiempos requeridos para el procesamiento de un LOA.	190
7.7.	Tiempos requeridos para el descubrimiento en las Páginas Amarillas de JADE.	193
7.8.	Tiempos requeridos para el descubrimiento en jUDDI.	193
7.9.	Tiempos requeridos para el descubrimiento en el repositorio de LACs.	194
7.10.	Tiempos requeridos para el descubrimiento en JADE, jUDDI y el repositorio de LACs.	195
8.1.	Mapa de la red física de IoT.	203
8.2.	Principales descriptores empleados para el enriquecimiento semántico de los LOAs.	204
8.3.	Diagrama de estado UML del LOA AHVAC.	207
8.4.	Diagrama de actividad UML del LOA AHVAC.	208
8.5.	Representación del ecosistema de IoA.	211
8.6.	Relaciones entre los LOAs del ecosistema de IoA.	213
8.7.	Sociograma del ecosistema de IoA.	214
A.1.	Esquema de un sistema multiagente basado en la definición de Wooldridge.	237
A.2.	Arquitectura de agente (a) reactiva y (b) deliberativa.	238
D.1.	Interfaz para crear un WAC.	243
E.1.	Interfaz de enriquecimiento semántico	245
F.1.	Especificación UML usada para el desarrollo de la herramienta <i>IoA-Building Tool</i> .	247

Índice de tablas

2.1. Resultados de la búsqueda realizada	43
2.2. Principales propuestas que modelan objetos de IoT mediante agentes.	46
2.3. Principales modelos de objetos modelados mediante sistemas multiagentes.	52
2.4. Principales aplicaciones reales modeladas a partir de IoT y agentes	56
3.1. Uso de los aspectos asociados al IoA.	75
4.1. Ontologías que describen aspectos relacionados con el IoA.	88
7.1. Nivel de precisión y exactitud del proceso de descubrimiento semántico.	196
8.1. Principales aspectos evaluados en sistemas IoA.	217
B.1. Fuentes primarias de información.	241

Glosario

ACL	Agent Communication Language <i>Lenguaje de Comunicación para Agentes</i>
AmI	Ambient Intelligence <i>Inteligencia Ambiental</i>
AI	Artificial Intelligence <i>Inteligencia Artificial</i>
AOP	Agent Oriented Paradigm <i>Paradigma Orientado a Agentes</i>
BDI	Belief, Desire, Intention <i>Creencia, Deseo e Intención</i>
BLE	Bluetooth Low Energy <i>Bluetooth de Baja Energía</i>
DbC	Design by Contract <i>Diseño por Contrato</i>
DOHA	Dynamic Open Home Automation <i>Automatización Domótica Abierta y Dinámica</i>
DPWS	Devices Profile for Web Services <i>Perfil de Dispositivos para Servicios Web</i>
FIPA	Foundation for Intelligent Physical Agents <i>Fundación para Agentes Físicos Inteligentes</i>
GSM	Global System for Mobile communications <i>Sistema Global para las Comunicaciones Móviles</i>
GPRS	General Packet Radio Service <i>Servicio General de Paquetes Vía Radio</i>
HVAC	Heating, Ventilation, and Air Conditioning <i>Sistema de Calefacción, Ventilación y Aire Acondicionado</i>
IaaS	Infraestructura como Servicio (IaaS), <i>Infraestructura como Servicio</i>
IoA	Internet Of Agents <i>Internet de los Agentes</i>

IoC	Internet Of Content <i>Internet del Contenido</i>
IoE	Internet Of Everything <i>Internet del Todo</i>
IFTTT	If-This Then-That <i>Si-Esto, Entonces-Aquello</i>
IoP	Internet Of People <i>Internet de las Personas</i>
IoS	Internet Of Services <i>Internet de los Servicios</i>
IoT	Internet Of Things <i>Internet de las Cosas</i>
IP	Internet Protocol <i>Protocolo de Internet</i>
IRI	Internationalized Resource Identifier <i>Identificador de Recursos Internacionalizado</i>
JADE	Java Agent DEvelopment Framework <i>Marco de Desarrollo de Agentes Java</i>
JSON	JavaScript Object Notation <i>Notación de Objetos JavaScript</i>
JSON-LD	JavaScript Object Notation for Linked Data <i>Notación de Objetos JavaScript para Datos Enlazados</i>
jUDDI	Java implementation of the UDDI <i>Implementación Java de UDDI</i>
LAC	Linked Agent Contract <i>Contrato Enlazado de Agente</i>
LOA	Linked Open Agent <i>Agente Abierto Enlazado</i>
LOD	Linked Open Data <i>Datos Abiertos Enlazados</i>
MAS	Multi-Agent System <i>Sistema Multiagente</i>
NFC	Near Field Communication <i>Comunicación de Campo Cercano</i>
OWL	Ontology Web Language <i>Lenguaje Web de Ontología</i>
openHAB	Open Home Automation Bus

	<i>Bus Domótico Abierto</i>
PaaS	Platform as a Service <i>Plataforma como Servicio</i>
RDF	Resource Description Framework <i>Marco de Descripción de Recursos</i>
REST	Representational State Transfer <i>Transferencia de Estado Representacional</i>
RFID	Radio Frequency Identification <i>Identificación por Radiofrecuencia</i>
ROA	Resource Oriented Architecture <i>Arquitectura Orientada a Recursos</i>
SaaS	Software as a Service <i>Software como Servicio</i>
SOA	Service Oriented Architecture <i>Arquitectura Orientada a Servicios</i>
SOAP	Simple Object Access Protocol <i>Protocolo Simple de Acceso a Objetos</i>
SOC	Service Oriented Computing <i>Computación Orientada a Servicios</i>
SPARQL	SPARQL Protocol and RDF Query Language <i>Protocolo SPARQL y Lenguaje de Consulta RDF</i>
UUID	Universally Unique Identifier <i>Identificador Único Universal</i>
URI	Uniform Resource Identifier <i>Identificador de Recursos Uniforme</i>
WAC	Workflow for Agent Control <i>Flujo para el Control del Agente</i>
WADL	Web Application Description Language <i>Lenguaje de Descripción de Aplicaciones Web</i>
WiFi	Wireless Fidelity <i>Fidelidad Inalámbrica</i>
WS	Web Service <i>Servicio Web</i>
WSDL	Web Services Description Language <i>Lenguaje de Descripción de Servicios Web</i>
XML	eXtensible Markup Language <i>Lenguaje de Marcado Extensible</i>

Introducción

Antecedentes y motivación

Nikola Tesla y Mark Weiser han sido dos grandes científicos visionarios en el campo de la Computación Ubicua y lo que hoy se conoce como el Internet de las Cosas (*Internet of Things*, IoT siglas en inglés) [119]. Por un lado, en el año 1926, Nikola Tesla visionaba un “*mundo conectado*” donde sería factible aplicar tecnología inalámbrica para que toda la Tierra se convirtiera en un enorme cerebro. Además, en aquella época, el mismo científico afirmó que las acciones de cómputo en aquel mundo conectado se llevarían a cabo mediante dispositivos que entrarían fácilmente en un bolsillo, y de una manera más simple de lo en aquella época resultaba usar un teléfono [134]. Por otro lado, décadas más tardes, en el año 1991, Mark Weiser pronosticó también un mundo que interconectaría cosas cotidianas de una manera no intrusiva para ayudar a las personas en sus actividades diarias [267]. Esta previsión dio origen a un nuevo término denominado “*Computación Ubicua*”.

A partir de la concepción de Weiser han surgido varios conceptos emergentes. Entre los conceptos más relevantes figuran la computación pervasiva, computación móvil, computación invisible e inteligencia ambiental. Estos conceptos, en conjunto con el desarrollo de la Microelectrónica—miniaturización de los dispositivos—, así como los avances en las Telecomunicaciones—Internet—han sido significativos para que las previsiones de Tesla y Weiser se hayan convertido en una realidad en términos prácticos. Adicionalmente, dichos conceptos han coadyuvado para el desarrollo del Futuro Internet, un paradigma cuya actual generación está centrada en el IoT [23].

El IoT es un concepto novedoso usado por primera vez por Kevin Ashton en el año 1995 [20]. Su amplia aceptación en diferentes dominios lo ha convertido en un paradigma emergente en proceso de consolidación (aún requiere superar retos como la interoperabilidad y la estandarización,

gestión de escalabilidad y flexibilidad para integrar nuevas tecnologías, gestión de nombres e identidades, gestión de servicios, gestión de heterogeneidad, gestión de la privacidad y seguridad, movilidad, eficiencia de consumo de energía, soporte de tiempo real, y la fiabilidad [6, 140]). Dicho paradigma introduce el concepto de objeto, dispositivo o cosa (“*thing*”) como principal elemento activo en procesos empresariales, industriales, informativos y sociales. Esto implica que a través de este tipo de elementos, y gracias a su soporte de conectividad a Internet, es posible la interconexión de personas y objetos físicos en cualquier momento, cualquier lugar, con cualquiera, a través de cualquier red y mediante el uso de cualquier servicio disponible [6, 23, 106, 203]. A raíz de ello se pueden formar ecosistemas heterogéneos donde personas y máquinas interactúan entre sí para percibir y controlar el mundo real.

Considerando que las interacciones en ecosistemas de IoT no solo se dan entre cosas o dispositivos compatibles con las interfaces y protocolos de comunicación usados en Internet, en este tipo de ecosistemas las personas también desempeñan un papel importante. Los usuarios constituyen entidades activas y singularmente inteligentes que manejan y supervisan el funcionamiento de los objetos conectados. Por este motivo el término más apropiado para referirse a este tipo de tecnologías sería Internet del Todo (*Internet of Everything*, IoE por sus siglas en inglés) [2, 89, 248].

Aunque hasta ahora se ha relacionado directamente a los objetos con las personas que los manejan, en los ecosistemas de IoE intervienen entidades adicionales. Algunas de estas entidades, como los servicios y el contenido generado, se complementan con las personas y los objetos interconectadas y han originado nuevos enfoques. Entre estos enfoques figuran los siguientes: el Internet de las Personas (*Internet of People*, IoP por sus siglas en inglés) enfocado en gestionar la interacción de los usuarios con el IoT [66], Internet del Contenido (*Internet of Content/Resource*, IoC por sus siglas en inglés) orientado a gestionar y procesar los datos masivos generados en ecosistemas de IoT [257] y el Internet de los Servicios (*Internet of Services*, IoS por sus siglas en inglés) centrado en la gestión de servicios especializados en IoT [47]. En conjunto estos enfoques han ayudado a gestionar y optimizar las tareas ejecutadas en ecosistemas de IoT, y así, llevar al Futuro Internet hacia una siguiente generación donde los objetos con los que las personas interactúen se comporten de manera más inteligente que en la actualidad. Las directrices de cómo estos enfoques

apoyan y complementan al IoT son abordadas en esta investigación.

La actual generación del Futuro Internet gira alrededor del IoT [140]. Sin embargo, la próxima generación del Futuro Internet estará estrechamente ligada a la web inteligente (Web 4.0) la cual propone la creación de ecosistemas gestionados por agentes inteligentes. Por tanto, la próxima generación del Futuro Internet, y del IoE, estará caracterizada por ecosistemas donde las personas interactúen con entidades gestoras de las cosas, sin que éstas puedan identificar fácilmente si dicha entidad gestora con la que interactúan constituye un humano o una máquina. Esto implica que la próxima generación del IoT estará determinada por objetos realmente inteligentes (*smart things/smart objects*) que poseen características inherentes a los agentes software como adaptabilidad, proactividad, reactividad, racionalidad, sociabilidad y autonomía [26, 101, 271]. Esta ha sido una de nuestras motivaciones que nos llevó a plantear esta Tesis e investigar un nuevo paradigma que incluyera modelos y mecanismos de agentes software especializados para crear sistemas de IoT más inteligentes, pro-activos, auto-adaptables, sociales y colaborativos que los sistemas de IoT genéricos.

A raíz de la idea de integrar las tecnologías orientadas a agentes sobre las tecnologías de IoT se originaron una serie de interrogantes entre las que figuran las siguientes: (i) ¿serían los agentes de software un mecanismo adecuado para que los objetos de IoT generalmente pasivos pueda llegar a modelar comportamientos pro-activos y autónomos?, (ii) ¿de qué forma se podría modelar un ecosistema de IoT a través de agentes, usando agentes simples o sistemas multiagente?, (iii) ¿cuál sería el modelo de agente que mejor se adapte a gestionar ecosistemas de IoT, reactivo o deliberativo?, (iv) ¿cuál debería ser el comportamiento genérico que un agente que opera en IoT debe modelar para ejecutar procesos de manera inteligente, colaborativa y adaptativa?, (v) ¿qué tipo de inteligencia debería modelar un agente para llevar a cabo sus tareas de una manera precisa y sin exigencia de grandes recursos de cómputo?. Estas interrogantes han sido respondidas en esta investigación.

Los agentes software deben tener una identidad propia, y ser autónomos y pro-activos, lo que le permite coexistir, así como compartir o colaborar con otros agentes de ecosistemas heterogéneos de IoT. Por tanto, esto implica que debe satisfacer un cierto grado de interoperabilidad funcional para que establezcan comunicación con agentes heterogéneos,

independientemente de la plataforma en que se encuentren alojados y de la herramienta en la que haya sido creado. Ante estas necesidades, FI-PA (Foundation for Intelligent Physical Agents, por sus siglas en inglés) planteó hace algunos años un lenguaje estándar para agentes denominado lenguaje ACL (*Agent Communication Language* [93], por sus siglas en inglés) que planteaba una solución a esta problemática. Sin embargo, propuestas prometedoras encaminadas a mejorar la interoperabilidad en los ecosistemas de agentes están centradas en el uso de tecnologías semánticas [127, 234]. Se plantea entonces, la necesidad de crear nuevos modelos de agentes que sean capaces de ejecutar procesos de razonamiento semántico y proceso de descubrimiento semántico encaminados a gestionar el IoT de manera más precisa. Esto nos motivó a plantear un novedoso modelo de agente semántico que gestione los recursos de IoT en función del significado de las cosas.

Seguidamente, es importante resaltar que para realizar la integración de agentes con las tecnologías de IoT no existen metodologías especializadas que guíen al desarrollador durante el diseño y desarrollo de este tipo de sistemas. Existen actualmente más de una decena de Metodologías Orientadas a Agentes entre las que sobresalen: MASCommonKads, Gaia, Tropos, Prometheus, Passi, Adelfe, RAP, Message, Ingenias, Desire y MaSE [5, 238, 259]. Pero, la mayoría de ellas, únicamente soportan dominios específicos y consecuentemente, poca o ninguna metodología toma en cuenta el modelado integral de aspectos importantes que gestionan comúnmente los sistemas de IoT modelados usando agentes (i.e., contexto, círculos sociales y colaboración, inteligencia, soporte de ecosistemas de servicios, y objetos y recursos de IoT). Por tanto, se requiere disponer de instrumentos metodológicos que indiquen el camino a seguir para crear agentes especializados en gestionar el IoT y que así, este tipo de sistemas puedan llegar a consolidarse en el terreno práctico como prescribe la Ingeniería de Software. Un método sistemático especializado en el diseño de ecosistemas de IoT inteligentes y colaborativos, aplicando la tecnología de agentes ha sido formulado como complemento al modelo de agente planteado en esta memoria. Se espera que los instrumentos planteados en este trabajo sirvan de base para que los desarrolladores puedan crear aplicaciones de IoT inteligentes y colaborativas en escenarios reales, los cuales son escasos, ya que no existen las suficientes herramientas consolidadas.

Objetivos

El objetivo principal de este trabajo de tesis se centra en investigar cómo se pueden integrar los agentes software como entidades software que permiten dotar de inteligencia, proactividad, autonomía, sociabilidad o la capacidad de colaboración, entre otros, a los ecosistemas de Internet de las Cosas. Dicho trabajo está enmarcado en el desarrollo de un enfoque emergente novedoso (“Internet de los Agentes”) que puede constituir las bases del Futuro Internet.

A un nivel más detallado, los objetivos específicos que han dirigido la ejecución de la presente investigación son descritos como sigue:

- Analizar el impacto de las tecnologías orientadas a agentes (como alternativa a la computación inteligente) en el campo de los sistemas distribuidos de control implementados a través de tecnología IoT.
- Formular un marco de trabajo que describa las bases conceptuales y terminología básica orientadas a crear sistemas híbridos de control inteligente basados en tecnologías de IoT y agentes software.
- Desarrollar un modelo especializado de agente orientado a gestionar los recursos de ecosistemas de IoT de manera inteligente, pro-activa, adaptativa y colaborativa.
- Formular un método sistemático de diseño que provea las directrices para la creación de sistemas de IoT basados en agentes software.
- Definir métricas de operación y calidad de servicio para evaluar los principales métodos y modelos propuestos aplicados en casos de estudios prácticos donde se planteen acciones de monitorización en entornos cotidianos de Inteligencia Ambiental.

Estructura

De manera general, las tareas desarrolladas para alcanzar los objetivos plantados en esta Tesis se encuentran organizados en nueve capítulos adicionales a la Introducción de la memoria. Dichos capítulos son descritos como se detalla a continuación:

El Capítulo 1 hace un acercamiento a los ejes temáticos sobre los que se fundamenta la investigación. Esto es, tópicos asociados al Futuro Internet y los principales enfoques relacionados tales como: Internet de las Cosas, Internet de los Servicios, Internet de las Personas, Internet del Contenido e Internet de los Agentes. Se plantea también una discusión en relación al Futuro Internet a corto y largo plazo, en términos de IoT y el IoA respectivamente.

A continuación, el Capítulo 2 provee una revisión detallada de la literatura actual entorno al proceso de agentificación del IoT a fin de identificar los principales retos, tendencias y oportunidades para el desarrollo de sistemas inteligentes de IoT. Los resultados obtenidos en este capítulo permitieron responder las preguntas de investigación planteadas en relación al paradigma del IoA y su aporte en términos de inteligencia, autonomía y adaptación de los objetos conectados a ecosistemas de IoT.

El Capítulo 3 define y detalla las principales directrices en relación al enfoque del IoA. Considerando que se trata de un enfoque emergente del que no se tienen definiciones formales, se lo ha formalizado considerando la fusión de tres áreas tales como IoT, agentes y Web Semántica para alcanzar un IoT inteligente. A modo complementario se detallan los principales aspectos que un ecosistema de IoA debe modelar y se describen también 4 casos de aplicación prácticos donde se puede emplear el IoA para mejorar la inteligencia en las actuales aplicaciones de IoT.

El Capítulo 4 parte de los aspectos relevantes a ser modelados por el IoA, descritos en el capítulo anterior desde un nivel de abstracción alto, para construir una ontología especializada que describe ecosistemas de IoA. El resultado obtenido fue la ontología IoA-OWL y es descrita en detalle en este apartado.

El Capítulo 5 presenta una visión general del marco de trabajo Framework-IoA que describe un modelo orientado al desarrollo de sistemas inteligentes de IoT, donde los objetos y sus recursos son gestionados a través de Sistemas Multiagentes (MASs) que incorporan entidades LOA (*Linked Open Agent*) de tipo *Agent of Thing*, *Agent of Coordination*, *Agent of Data* y *Agent of People*. Estos agentes son descritos mediante un contrato semántico (*Linked Agent Contract*, LAC) y dirigidos por un flujo de control especializado (*Workflow for Agent Control*, WAC). Y además, dichos agentes modelan su comportamiento a través de tareas de agente específicas tales como: *Asynchronous Agent Response Task*,

Workflow Execution Task, Injection Task y *Agent Discovery Task*. A través de ellas los agentes son capaces de establecer comunicación, ejecutar acciones sobre el ecosistema de IoT y descubrir agentes contrapartes basados en procesos de razonamiento semántico para llevar a cabo tareas colaborativas, respectivamente.

Posteriormente, el Capítulo 6 define la secuencia de pasos lógicos propuestos a través de un método sistemático especializado para materializar el proceso de agentificación del IoT en término de herramientas metodológicas de la Ingeniería del Software. El método propuesto plantea dividir el proceso de modelado en dos niveles, microscópico y macroscópico. A nivel micro se define el comportamiento del agente a nivel individual; mientras que a nivel macro el agente es modelado considerando su interacción con el ecosistema de IoA.

En el Capítulo 7 y Capítulo 8 se presentan dos casos de estudio donde se modela una comunidad y una oficina inteligente a partir de los métodos propuestos en esta investigación. En términos generales, el modelado de ambos sistemas describen aspectos como el alcance, descripción del escenario de IoT e IoA, el conjunto de descriptores empleados por los agentes de tipo *Agent of Thing*, *Agent of Coordination*, *Agent of Data* y *Agent of People*, y el proceso seguido para crear cada sistema. Adicional a los aspectos de diseño, se presentan también, los resultados cualitativos y cuantitativos obtenidos en relación a la evaluación de la funcionalidad del modelo de la entidad LOA, su desempeño de operación y los niveles de precisión del proceso de razonamiento semántico llevado a cabo por los agentes colaborativos para la recuperación de contrapartes y datos.

Finalmente, el Capítulo 9 presenta las principales conclusiones y trabajos futuros que surgen a raíz del desarrollo de la investigación basadas en los resultados obtenidos en los capítulos anteriores y las discusiones planteadas en cada uno de ellos. Además, se listan las aportaciones científicas resultantes de la investigación y se presenta un breve resumen de cada una de ellas.

El autor de esta memoria han creado un recurso en la web para listar las posibles erratas incluidas en esta Tesis Doctoral. La información es accesible desde la sección "DOCTORAL THESIS" de la siguiente dirección electrónica <http://core.ugr.es/ioaweb/>.

Capítulo 1

Perspectivas actuales y futuras del Internet

“La ciencia es la estética de la inteligencia”—**Gastón Bachelard**.

En los años sesenta se desarrollaron las bases de Internet, hoy en día considerada como la red descentralizada de ordenadores más grande a nivel mundial que ha supuesto una revolución sin precedentes en el campo de las Telecomunicaciones [140]. Una variedad de servicios tradicionales han sufrido un proceso de transformación hacia las Nuevas Tecnologías de la Información y la Comunicación (NTICs) asociadas a Internet. El alto grado de accesibilidad a Internet a nivel mundial ha permitido que la sociedad actual pueda sacar provecho de innovaciones tecnológicas tales como: correo electrónico, navegación web, servicio de transmisión de datos, comercio electrónico, redes sociales, entre otros [142]. En este contexto, la aceptación de Internet ha conllevado a que exista un alto nivel de conectividad entre dispositivos y personas, surgiendo así el término denominado Futuro Internet [130].

Algunos de los paradigmas más actuales soportadas por el Futuro Internet giran en torno a la Web Inteligente, Web Semántica, Datos Enlazados, Big Data, Computación en la Nube y el Internet de las Cosas (IoT) [283]. Éste último emerge como un novedoso paradigma para dar soporte a entornos con alto grado de conectividad donde personas y objetos pueden conectarse a Internet en cualquier momento, en cualquier lugar, y con cualquiera [38, 220]. A raíz de éste, otros paradigmas complementarios tales como: el Internet de los Servicios (IoS), Internet del Contenido (IoC), Internet de las Personas (IoP) e Internet de los Agentes (IoA), han sido propuestos. En este capítulo, se describen las bases conceptuales, alcance y tecnologías empleadas por estos paradigmas.

1.1. Evolución del Internet

El Internet es un paradigma constantemente emergente. Desde su aparición como ARPANET (*Advanced Research Projects Agency Network*) en los años sesenta, el Internet y las tecnologías vinculadas a éste, han evolucionado progresivamente. En un principio servicios ligados al Internet como el correo electrónico y la web, estuvieron orientadas principalmente a potentes ordenadores que solo eran usados para fines militares [156]. No obstante, con el pasar de los años, su uso se fue popularizando progresivamente hasta alcanzar el ámbito empresarial, educativo, científico e incluso el uso personal.

En las últimas décadas, las tecnologías vinculadas con Internet han evolucionado a pasos agigantados. Esto se ha debido principalmente a que el ordenador pasó de ser un bien orientado a grandes organizaciones, a ser un producto orientado al usuario común. Así, la actual sociedad de la información ha tenido mayor accesibilidad a Internet y sus correspondientes tecnologías y servicios. Entre estas tecnologías y servicios figuran las siguientes: correo electrónico, mensajería instantánea, navegación web, streaming, redes sociales, entre otros [64].

Según el portal *Statista*, el número de usuarios de Internet (internautas) se ha ido incrementando de manera lineal en el mundo. Como se ilustra en la Figura 1.1, del año 2005 al 2017 se ha producido un incremento del 71 %. Esto implica que el número de internautas se ha incrementado a un aproximado de 2554 millones de usuarios. Adicionalmente a este portal, el Portal *eMarket* proyectan que el número de usuarios de Internet siga incrementándose de manera gradual en los siguientes años y a medida que la brecha tecnológica vaya disminuyéndose en países en desarrollo. Este fenómeno de crecimiento será posible también gracias a que los usuarios de Internet no solo se conectarán desde ordenadores personales para llevar a cabo tareas cotidianas y de trabajo; sino que también accederán a Internet con fines de entretenimiento y ocio. En este último caso los usuarios en general se interconectan desde dispositivos móviles, tabletas y relojes inteligentes. Dichos dispositivos son cada vez más sofisticados y muchos de ellos soportan algunos de los actuales estándares de comunicación inalámbricos (i.e. RFID, NFC, ZigBee, Bluetooth, BLE, UWB, WiFi, Wi-Max [163, 231]) y/o celular (i.e. GSM, GPRS, 3/4/5G [38, 231]), a través de los cuales pueden establecer

conexión a Internet desde cualquier punto geográfico y sin necesidad de un medio físico (cable).

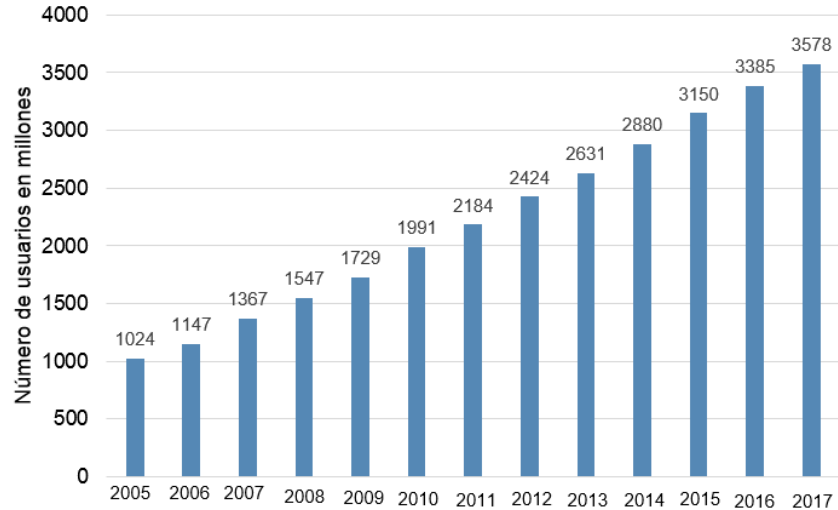


FIGURA 1.1: Número de internautas en todo el mundo de 2005 a 2017 según datos del Portal Statista.

Proyecciones adicionales realizadas por el Portal *Statista* detalladas en la Figura 1.2, muestran al igual que el caso anterior, un incremento progresivo del número de usuarios que tienen acceso a un teléfono inteligente. Al año 2017 se proyectó que 2.32 mil millones de personas tendrían acceso de un teléfono inteligente. Así mismo, se espera que al año 2020 el número de usuarios de teléfonos inteligentes aumentará en un 11,85%. Esto corresponde a un total de 2,87 mil millones de usuarios, muchos de los cuales accederán también a Internet para aprovechar al máximo las prestaciones y servicios que ofrecen los dispositivos móviles inteligentes actuales y futuros.

De manera general, la mayoría de internautas emplean actualmente sus teléfonos inteligentes para realizar tareas de comunicación (texto, voz y vídeo digital) y tareas de ocio (redes sociales, juegos y aplicaciones específicas) a través de Internet. No obstante, en la actualidad los usuarios de teléfonos inteligentes también lo están empleando para controlar dispositivos electrónicos comercializados como objetos inteligentes que mejoran su calidad de vida. Entre estos objetos, fácilmente asequibles por los usuarios en general, figuran artefactos como: bombillas inteligentes, cámaras

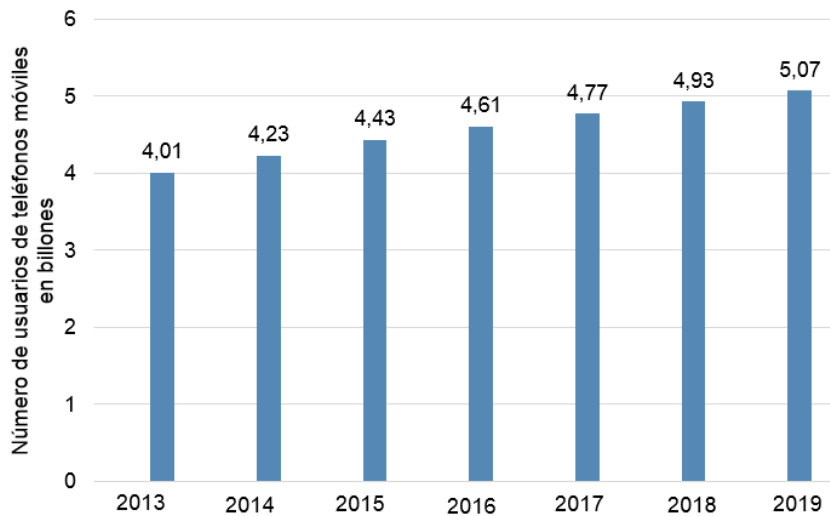


FIGURA 1.2: Número de usuarios de teléfonos inteligentes a nivel mundial entre 2014 y 2020 según proyecciones del Portal Statista.

de vídeo, persianas, puertas inteligentes, entre otros. En la misma línea, electrodomésticos de uso cotidiano en el hogar como televisores, refrigeradores y reproductores de audio inteligentes también pueden conectarse a Internet con la finalidad de acceder a servicios y contenido multimedia online (i.e., series, novelas, noticias, pistas, listas de reproducción, catálogos de productos, entre otros). De esta manera, como se ilustra en la Figura 1.3, dispositivos que anteriormente eran analógicos, han llegado a convertirse en dispositivos digitales con capacidades de para ser programados a través de Internet. Esto les habilita para que puedan acceder a servicios distribuidos en la Nube—un novedoso paradigma emergente donde se emplea al Internet como medio para ofrecer “todo” como servicio (*Everything as a Service*, XaaS por sus siglas en inglés) [78].

El impacto que ha tenido el Internet en la sociedad actual ha conllevado a la introducción de un novedoso concepto al que se ha denominado Futuro Internet. Aunque este término ha estado intrínseco con el Internet desde sus orígenes y en las diferentes etapas de su evolución [140], en la actualidad el Futuro Internet guarda una estrecha relación con las personas, y sus dispositivos móviles y objetos, a partir de los cuales se conectan a Internet. De esta manera, se habilita la interconexión



FIGURA 1.3: Objetos cotidianos interconectados a Internet en un escenario de IoT.

de entidades en espacios físicos abiertos (i.e., atmósfera, ríos, carreteras, parques) y cerrados (i.e., hogares, edificios, plantas industriales, centros educativos, hospitales, coches) para convertirlos en espacios inteligentes (i.e., hogares, industria, vehículos y hospitales inteligentes) donde las tareas cotidianas se realizan de manera automática y con el menor grado de interacción humana posible.

En la Literatura actual, el Futuro Internet gira en torno a la creación de un mundo ampliamente conectado y controlado desde cualquier punto, a cualquier momento y por cualquiera que se encuentre interconectado al ciberespacio. Por tanto, visto desde la óptica de las actuales tecnologías, el Futuro Internet está íntimamente vinculado con un novedoso paradigma denominado *Internet de las Cosas* (*Internet of Things*, IoT por sus siglas en inglés) [254]. Bajo esta premisa, y considerando que no existe una única definición formal y aceptada que defina el *Futuro Internet* [123], seis aproximaciones validadas por la comunidad científica sobre este enfoque son citadas a continuación. Dichas definiciones no son las únicas planteadas; pero han sido consideradas como las que más se acercan a las últimas previsiones y proyecciones sobre el futuro del Internet a corto, mediano y largo plazo.

Definición 1. Según Eloff et al. [84], el *Futuro Internet* consiste en “miles de millones de dispositivos digitales, personas, servicios y otros

objetos físicos que tienen el potencial de conectarse, interactuar e intercambiar información sobre sí mismos y su entorno”.

Definición 2. Para Boniface, SurrIDGE y Upstill [36], el Futuro Internet es un “*sistema socio-técnico que incluye información y servicios accesibles a través de Internet, acoplados al entorno físico y al comportamiento humano, y que apoya aplicaciones inteligentes de importancia social*”.

Definición 3. Desde la óptica Canale et al. [49], el Futuro Internet incluye la presencia de “*tecnología capaz de controlar eficientemente los recursos de red que se encuentran disponibles de forma contextualizada y en bucle cerrado y, al mismo tiempo, capaz de garantizar la satisfacción del usuario*”.

Definición 4. D’Angelo y Caporuscio [82] ven al Futuro Internet como un paradigma que modela “*un entorno de ejecución mundial que interconecta una miríada de entidades heterogéneas a lo largo del tiempo, apoya la diseminación de información, permite el surgimiento de dominios de aplicación prometedores y estimula nuevas oportunidades de negocio e investigación*” .

Definición 5. Derhab, Guerroumi y Younis [73] conciben al *Futuro Internet* como una “*red de dispositivos de detección inalámbricos y móviles, donde un gran conjunto heterogéneo de estos dispositivos está estrechamente acoplado para permitir una gestión inteligente de muchos aspectos de nuestras actividades diarias*”.

Definición 6. Samad y Memon [230] definen al Futuro Internet como el proceso de “*transición entre el IPv4 (Internet actual) y el IPv6 (Futuro Internet)*”.

Las definiciones previamente descritas guardan un aspecto común. En general, se visiona al Futuro Internet como un ecosistema altamente conectado donde máquinas y humanos interactúan e intercambian datos, información y conocimiento entre sí. Estos elementos son gestionados por las redes de comunicación tales como: redes de ordenadores, redes de sensores inalámbricos y redes celulares [230]. Por tanto el Futuro Internet está estrechamente vinculado con el IoT.

En torno al IoT, otros paradigmas complementarios que dan soporte al Futuro Internet han surgido acorde a cómo el IoT se ha ido consolidando y popularizando. En la actualidad multitud de aplicaciones basadas en el IoT han sido desplegadas para crear un mundo altamente conectado y así, ayudar en la automatización de las actividades diarias que los seres humanos realizan en el mundo real. En vista de que los seres humanos son parte activa del Futuro Internet, se requieren por tanto, mecanismos especializados para llevar a cabo la comunicación e interacción hombre-máquina [114, 146]. Dichos mecanismos son adicionales a los orientados a la comunicación máquina-máquina (*Machine 2 Machine*, M2M por sus siglas en inglés) [272] y la ejecución de procesos inteligentes, autónomos, pro-activos y colaborativos.

Adicionalmente, en vista de que la interacción entre personas y dispositivos de IoT generan grandes volúmenes de datos heterogéneos que describen el mundo real, el comportamiento y relaciones humanas, y el conocimiento en general, se han requerido también tecnologías y herramientas especializadas en la captura, almacenamiento y análisis de dichos datos para darle un valor añadido; es decir, convertirlos en información y conocimiento útil que sirva de soporte a los procesos de toma de decisiones en ecosistemas basados en el Futuro Internet.

1.2. Paradigmas relacionados

Las implicaciones del IoT no solo están orientadas a conectar dispositivos para monitorizar condiciones del mundo real, y ejecutar acciones de control que mejoren la calidad de vida de las personas y la productividad de las empresas. Con el fin de proveer una mejor experiencia de usuario y coadyuvar en el desarrollo de sistemas ubicuos con un grado de cognición que se asimile a la inteligencia humana, el *Futuro Internet*, y consecuentemente el Internet de las Cosas, integra paradigmas emergentes especializados [82]. Entre estos paradigmas, ilustrados a continuación en la Figura 1.4, se incluyen los siguientes: Internet de los Servicios (*Internet of Services*, IoT) [249], Internet del Contenido (*Internet of Content*, IoC) [256], Internet de las Personas (*Internet of People*, IoP) [179], y recientemente el paradigma Internet de los Agentes (*Internet of Agents*, IoA) [278].

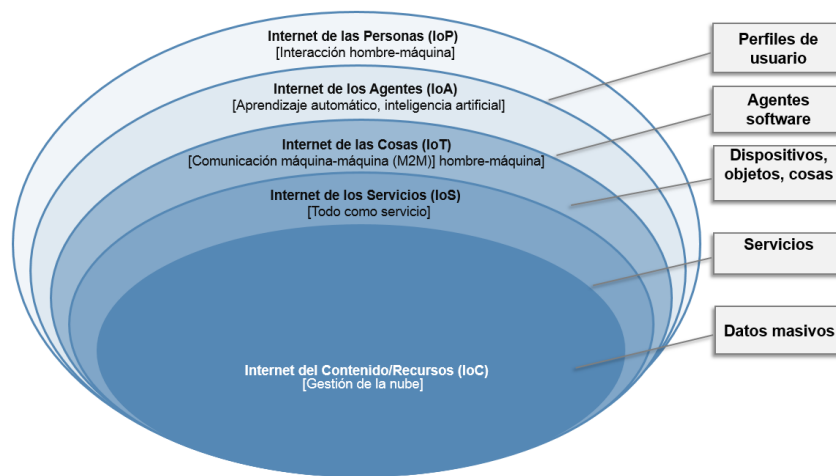


FIGURA 1.4: Principales paradigmas y entidades emergentes relacionados con el Futuro Internet.

En las siguientes sub-secciones se describen de manera detallada los principales paradigmas que guardan una relación estrecha con el Futuro Internet [82], posicionando al IoT como el paradigma emergente del Internet. De allí que se plantea una revisión general del IoT (dispositivos), el IoS (servicios), el IoP (usuarios), el IoC (datos masivos) y el IoA (agentes software), en términos de su respectivo alcance, definición formal, objetivo, situación actual, herramientas que lo soportan, y la forma en cómo se relacionan unos con otros. A partir de estas directrices se espera que el contenido de los siguientes capítulos que compone esta Tesis, y que hacen referencia a estos paradigmas, sean de fácil comprensión para el lector.

1.3. Internet de las Cosas

El Internet de las Cosas (Internet of Things, IoT por sus siglas en inglés) es un paradigma emergente que surge con el objetivo de controlar el mundo físico-real mediante el uso de una red global de objetos que se encuentran interconectados a través de Internet [6, 23, 245].

A raíz del paradigma del IoT, una amplia gama de objetos “inteligentes” han empezado a ser comercializados actualmente a fin de cubrir un

amplio espectro de aplicaciones reales en diferentes dominios del conocimiento tales como: cuidado de la salud [91, 148], gestión del confort en hogares inteligentes [252], gestión de los servicios públicos en ciudades inteligentes [280], gestión de procesos industriales [71], entre otros. Como ilustra la Figura 1.5, las aplicaciones de IoT están incrementándose progresivamente a medida que oferta de dispositivos heterogéneos (i.e., electrodomésticos inteligentes, cámaras de vídeo IP, sensores, actuadores, objetos tecnológico de vestimenta, robots, teléfonos inteligentes, tabletas y micro ordenadores) que adoptan diferentes estándares de comunicación inalámbrica aumentan [188]. En función de su alcance, dichos estándares habilitan el desarrollo de sistemas que cubren redes de área extendida (i.e., GPRS, GSM, 3G, 4G y LORA), redes de área local (i.e., LORA, Wavenis, M-Bus, SigFox) y redes de área doméstica (i.e., NFC, WiFi, ZigBee, RFID). Así, es factible desde el desarrollo de sistemas a pequeña escala orientado a proveer confort en hogares domóticos hasta el desarrollo de sistemas a una escala mayor enfocados en gestionar los servicios públicos de una ciudad, metrópolis, país o continente.

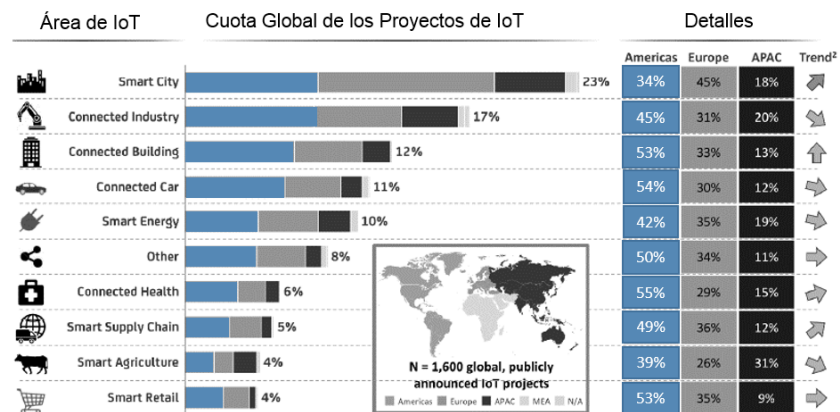


FIGURA 1.5: Tendencia de aplicaciones de IoT según el Portal IoT Analytics.

Según proyecciones del Portal *Statista*, el número de dispositivos compatibles con las tecnologías relacionadas con el IoT ha crecido exponencialmente desde el año 2015 hasta la actualidad. Estas tendencias de crecimiento, ilustradas en la Figura 1.6, visionan que al año 2025 el número total de objetos con soporte de interfaces compatibles con el IoT

que estarán interconectados a Internet será de aproximadamente 75,44 mil millones de dispositivos. Esto equivale a un incremento sustancial del 69,23 %, con respecto al 2018, año en que se ha realizado este análisis.

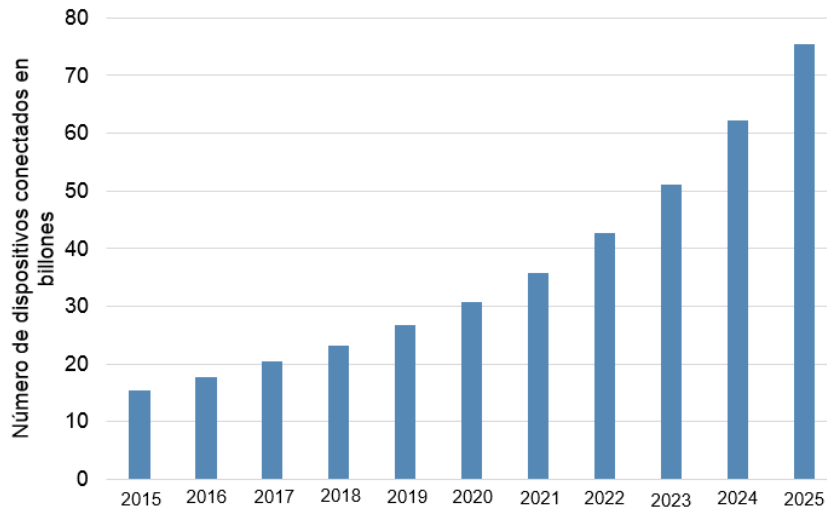


FIGURA 1.6: Dispositivos de IoT conectados alrededor del mundo de 2015 a 2025 según el Portal Statista.

El alto grado de conectividad existente hoy en día ha coadyuvado en la consolidación de las aplicaciones basadas en IoT. Gracias a los avances de las telecomunicaciones, el Internet, los teléfonos inteligentes y objetos compatibles con interfaces máquina-máquina han hecho posible que los actuales objetos—también denominados “cosas”—puedan conectarse a Internet en cualquier momento, en cualquier contexto, con cualquier servicio, con cualquier proveedor y a cualquier red [6, 23, 106, 203] independientemente de su grado de heterogeneidad. Sin embargo, el IoT sigue siendo un campo de investigación que aún presenta varios retos y que se encuentra en proceso de desarrollo y madurez.

Para que el IoT alcance un nivel de mayor consolidación, la comunidad científica ha planteado propuestas significativas enfocadas en la gestión de aspectos de infraestructura de los dispositivos, comunicación, identificación, gestión de eventos y analítica de datos, interfaces, protocolos y estándares. Algunos de ellos están relacionados con: la identificación, detección, comunicación, computación, servicio y aspectos asociados a la semántica [6, 23, 140]. Además, en torno al IoT se han planteado algunos

retos que buscan mejorar otros aspectos vinculados con: la interoperabilidad y la estandarización, gestión de escalabilidad y flexibilidad para integrar nuevas tecnologías, gestión de nombres e identidades, gestión de servicios, gestión de heterogeneidad, gestión de la privacidad y seguridad, movilidad, eficiencia de consumo de energía, soporte de tiempo real, y la fiabilidad [6, 140].

Algunos de los aspectos previamente especificados han sido abordados por middlewares especializados tales como: *TinySOA*, *IFTTT*, *ubiSOAP*, *SOCRADES*, *CarrIoT*, *Ubiware*, *ThingSpeak*, *Everyware* [177, 220]. Sin embargo, el soporte de todos los aspectos requeridos para el modelado de IoT no son generalmente contemplados por una misma herramienta. Las herramientas existentes en la actualidad son desarrolladas en base a una misma arquitectura de referencia como la propuesta por Fremantle [102]. Dicha arquitectura de referencia es ilustrada en la Figura 1.7 y no cubre aspectos específicos ya que estos dependen de la aplicación a desarrollar.



FIGURA 1.7: Arquitectura de referencia de IoT según Fremantle.

De manera general, la arquitectura mostrada en la Figura 1.7 constituye un modelo arquitectónico basado en capas. En el nivel más bajo se encuentra la capa de dispositivos de IoT (i.e., Arduino, Raspberry PI, dispositivos domésticos) que permite establecer comunicación entre los objetos y los servicios especializados disponibles en la nube a través de una segunda capa llamada la capa de comunicaciones. Esta capa gestiona el envío de mensajes a través del protocolo MQTT o HTTP/HTTPS. En este nivel, tanto las comunicaciones como los dispositivos se gestionan a través de una tercera capa especializada llamada capa de gestión de dispositivos.

En un nivel siguiente, una cuarta capa llamada capa de agregación es responsable de enrutar las comunicaciones a dispositivos específicos permitiendo el soporte para un servidor HTTP y/o un bróker MQTT desde el cual se activan los actos de comunicación con los dispositivos en la capa de dispositivos. Una quinta capa, llamada capa de procesamiento y análisis de eventos, se encarga de tomar los eventos del bus y luego de procesarlos en función de ellos; y, en segundo lugar, de almacenar los datos en medios persistentes para su posterior análisis y acceso por parte de los clientes a través de una sexta capa llamada capa de comunicación cliente/externo. Por último, una séptima capa, denominada capa de identificación y gestión de acceso, se encarga de la gestión de la seguridad y el acceso en la red de IoT en función de una política de control de acceso. Para ello, se establecen mecanismos enfocados en gestionar la identificación y el acceso a los dispositivos de entidades externas (i.e., usuarios, agentes).

Aunque existen arquitecturas de referencia generales para el desarrollo de sistemas de IoT como la descrita anteriormente. En la mayoría de los casos, las herramientas informáticas especializadas en el desarrollo de este tipo de sistemas adaptan estos modelos arquitectónicos para mantener compatibilidad con las necesidades, el ámbito de aplicación y la tecnología de hardware y software disponible. Esto se demuestra claramente en el trabajo realizado por Razzaque et al. [220], quienes han llevado a cabo una evaluación exhaustiva de los principales middlewares especializados para crear sistemas de IoT.

En su trabajo, Razzaque et al. [220] agrupan dichas herramientas en siete categorías tales como: basadas en eventos, orientadas a servicios, basados en máquinas virtuales, basados en tuplas, orientados a base de

datos, orientados a agentes y orientados a aplicaciones específicas. Sin embargo, las más ampliamente implementadas son los middlewares basados en eventos, servicios y agentes. Los primeros modelan los componentes, aplicaciones y todos los demás participantes interactúan a través de eventos. Los orientados a servicios modelan el suministro de funcionalidades mediante servicios y los correspondientes procesos de composición y descubrimiento automático de servicios en catálogos distribuidos. Finalmente, los basados en agentes modelan la lógica de las aplicaciones mediante el uso de entidades autónomas y móviles como son los agentes de software.

Los tres tipos de herramientas descritas anteriormente nos llevan a la conclusión de que la gestión de una capa de servicios en IoT es esencial para abstraer la complejidad de las comunicaciones y cualquier tipo de tarea que tanto los dispositivos como los usuarios requieran en un momento dado. Bajo esta premisa, el modelado de un ecosistema de servicios es beneficioso para el IoT.

1.4. Internet de los Servicios

Complementario al paradigma del IoT la introducción de un nuevo paradigma se ha planteado. Dicho paradigma, denominado Internet de los Servicios [194, 212] ha surgido gracias a las prestaciones de la Computación Orientada a Servicios (*Service-Oriented Computation*, SOC por sus siglas en inglés) en términos de reutilización de componentes, abstracción de la complejidad, y el uso de servicios como unidad de acoplamiento flexible en el desarrollo de sistemas distribuidos [237].

El Internet de los Servicios (*Internet of Services*, IoS por sus siglas en inglés) es considerado como la siguiente generación de los servicios proveídos en Internet [110]. Así, el IoS puede definirse como el paradigma que describe una infraestructura que usa Internet como medio para ofrecer y vender servicios especializados [47, 50]. Dicho paradigma es ampliamente usado a nivel empresarial y de gestión. No obstante, en los últimos años el IoS ha ido ganado terreno en el ámbito de las aplicaciones ubicuas y el IoT.

Algunos ejemplos de ecosistemas de servicios usados por las aplicaciones en general incluyen servicios especializados entre los que figuran los siguientes: *servicios de tráfico* (i.e., reporte de tráfico, control de tráfico,

seguimiento de vehículos), *servicios de administración pública* (i.e. pago de impuestos, servicios públicos), *servicios de salud* (i.e., servicios médicos, cuidado de la salud), *servicios meteorológicos* (i.e., monitorización ambiental, información meteorológica), y *servicios multi-media y redes sociales* (i.e., redes sociales, transferencia y almacenamiento masivo de datos) [277], entre otros.

La unidad de procesamiento empleada para el modelado de ecosistemas de IoS son los servicios. Un servicio, conocido también como servicio electrónico (*e-service*), es generalmente implementado a través de servicios web (*web services*). A la vez, un servicio web constituye un componente de software distribuido que posee una interfaz estándar bien definida la cual le permite interoperar con aplicaciones heterogéneas [87]. Su uso en el desarrollo de aplicaciones distribuidas orientadas a la web, dispositivos móviles, y objetos de IoT los ha convertido en uno de los principales medios para abstraer las comunicaciones entre dispositivos [118, 155] y usuarios independientemente de la plataforma, herramienta de programación o marco de trabajo (i.e., *JAX-WS*, *Microsoft .NET*, *WebLogic*, *WebSphere*, *Meteor-s*) empleado para su construcción. Esto ha sido alcanzado gracias a características asociadas a los servicios como: el uso de un contrato de servicio, bajo acoplamiento, encapsulación, abstracción, reusabilidad, composicionalidad y autonomía [86].

Los servicios web no son generalmente gestionados de manera individual; sino de manera agrupada o mediante colecciones. Una colección de servicios web puede ser gestionada mediante una Arquitectura Orientada a Servicios (*Service-Oriented Architecture*, SOA por sus siglas en inglés) [86], una Arquitectura Orientada a Recursos (*Resource-Oriented Architecture*, ROA por sus siglas en inglés) [259], o ambas [115]. Su uso depende del modelo de servicio web empleado para su implementación. En el primer caso, una arquitectura de tipo SOA es un estilo de diseño de software en la que los servicios se proporcionan a los demás mediante componentes de aplicación (autónomos, interoperables, disponibilidad, descubribles [221]) accesibles a través de un protocolo de comunicación y a través de una red. En el segundo caso, una arquitectura de tipo ROA cumple la misma función que SOA; pero los servicios son implementados a través de recursos [115] creados usando el modelo de servicios RESTful (*Representational State Transfer*) [168]—un modelo sin estado, accesible de manera uniforme, livianos y con mejor desempeño que los servicios

web genéricos [118].

En la Figura 1.8 es posible observar los principales actores que intervienen en una arquitectura SOA dirigida a IoT. De manera general, se ilustran los tres elementos básicos de una arquitectura orientada a servicios genérica tales como el proveedor del servicio (responsable de ofrecer una funcionalidad), consumidor del servicio (cliente que solicita una funcionalidad específica) y el directorio de servicios (descripción actualizada de los servicios disponibles). Sin embargo, a fin de añadir compatibilidad para gestionar los objetos de IoT se integra un nuevo componente (middleware) orientado a acceder a los objetos de IoT mediante el uso de servicios.

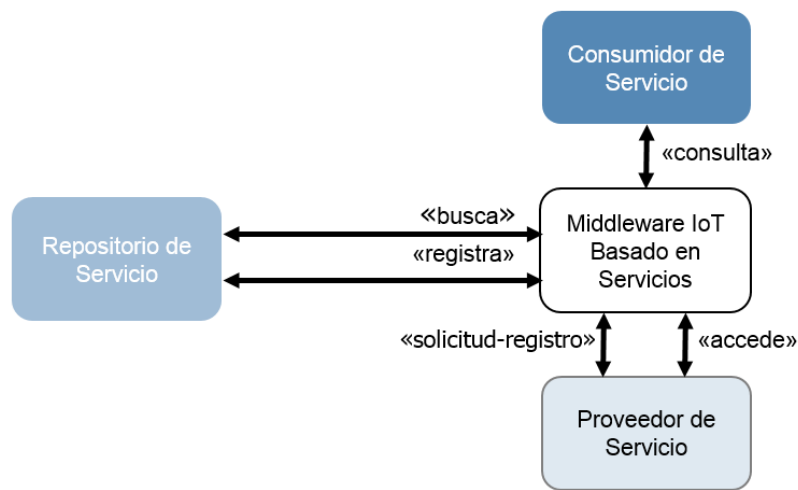


FIGURA 1.8: Principales entidades de una Arquitectura Orientada a Servicios.

Una plataforma o ecosistema de servicios se constituye habitualmente de servicios heterogéneos. La heterogeneidad de estos componentes está dada por el uso de servicios web que siguen diferentes criterios. Algunos de estos criterios son descritos en Pico-Valencia [206] e incluyen:

- Modelo de implementación: Servicios web SOAP [87], RESTful [259], DPWS (*Device Profile for Web Service*) [281] y DOHA (*Dynamic Open Home-Automation*) [223].
- Nivel de complejidad: Microservicio [76, 260], servicios compuestos [80].

- Herramienta de desarrollo: Plataformas (i.e Java, ASP .NET) y marcos de trabajo (i.e. Jersey, Apache Axis2).
- Contexto de operación: servidores web, tabletas, teléfonos inteligentes, dispositivos tecnológicos vestibles y sistemas embebidos [118, 181].
- Tipo de operación: Procesamiento de datos [225], procesos de razonamiento [25, 202], gestión de la nube [266].
- Manejo de elementos extra: sensibilidad al contexto [225], descripción semántica [174].

Con el propósito de soportar la heterogeneidad intrínseca de los ecosistemas de IoS, varias propuestas enfocadas en el consumo de servicios web heterogéneos han sido propuestas en la Literatura. Entre las más relevantes figuran los enfoques propuestos en Lee et al. [155], y Pico-Valencia et al. [212]. En ambas propuestas se plantea una herramienta con soporte para invocar servicios web de tipo SOAP, RESTful, y otros como DPWS y DOHA. Sin embargo, la herramienta propuesta por Pico-Valencia et al. [212] plantea el uso de los servicios web de manera uniforme desde agentes software que operan sobre ecosistemas de IoT.

El IoS y los servicios web han jugado un rol esencial en los ecosistemas de IoT [82, 194]. Al combinar estos dos enfoques los ecosistemas de IoT se han beneficiado de manera sustancial. Los actuales objetos de IoT emplean servicios web para manejar la interoperabilidad en IoT. Así, los objetos de IoT pueden establecer comunicación con otros dispositivos heterogéneos. Además, los usuarios pueden controlar estos dispositivos a través de interfaces amigables como es el caso de la web y teléfonos inteligentes.

Resumiendo, el IoS ha permitido a proveedores ofrecer servicios especializados que permiten gestionar los recursos de IoT. No obstante, dichos servicios tienen un comportamiento pasivo ya que su consumo depende de los propios dispositivos, los cuales son tradicionalmente no pro-activos. Esto implica que el grado de autonomía en IoT sigue siendo un reto aun introduciendo el IoS con el IoT.

1.5. Internet de las Personas

El término Internet de las Personas no es un paradigma reciente. Este paradigma ha sido utilizado anteriormente para referirse a los sistemas

web tradicionales diseñados específicamente a los usuarios [1, 179]. De manera general, el Internet de las Personas (*Internet of People*, IoP por sus siglas en inglés) hace hincapié en la conectividad digital de las personas a través de la infraestructura de Internet, formando una red de inteligencia colectiva y estimulando la comunicación interactiva entre las personas [179] (Figura 1.9). En este sentido, como afirman Aalst [1], los teléfonos inteligentes y las redes sociales desempeñan un rol muy importante en el ciberespacio.

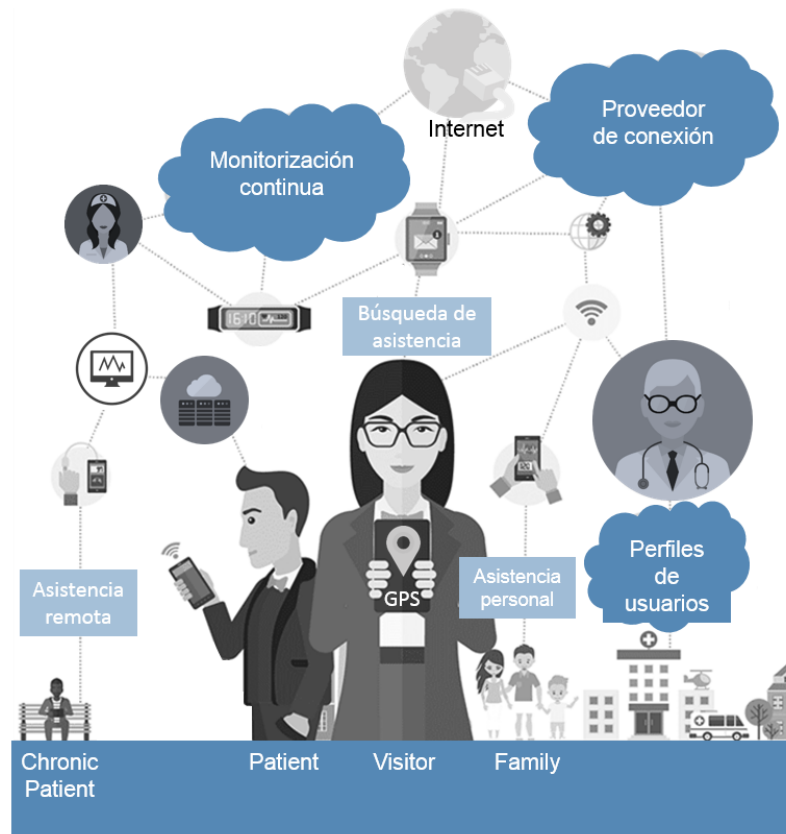


FIGURA 1.9: Los usuarios y su rol en el Internet de las Personas.

El enfoque del IoP, orientado hacia el IoT, es introducido de manera distinta a como lo hace la web tradicional. El IoP en entornos de IoT no ve a los usuarios y sus dispositivos personales como simple usuarios de

aplicaciones; sino como elementos activos de Internet [66]. Como muestra la Figura 1.9, a través de sensores vestibles no invasivos (*wereables*) y a través de teléfonos inteligentes, las personas llegan a convertirse en entidades que suministran datos de sus perfiles, sus condiciones fisiológicas y de geolocalización para adaptar de manera proactiva el comportamiento de los sistemas que usa [139].

Para que el IoP pueda alcanzar su objetivo en el Futuro Internet es necesario crear mecanismos especializados para que el IoT llegue a cumplir cuatro principios elementales que hagan más amigables la interacción humano-objetos de IoT [179]. Dichos principios son descritos a continuación:

- *Ser social*. Las interacciones sociales entre objetos de IoT y personas deben realizarse sin que su grado de heterogeneidad sea una limitación.
- *Ser personalizado*. Las interacciones con los dispositivos deben permitir personalización por parte de los usuarios en función de perfiles sociológicos y su contexto.
- *Ser proactivo*. Las acciones ejecutadas por los dispositivos deben ser puestas en marcha con mínima, y en el mejor de los casos con interacción nula de las personas.
- *Ser predecible*. Las acciones ejecutadas por los dispositivos se deben ejecutar en base a procesos de predicción del contexto, rutinas y comportamiento de los usuarios.

El IoP tiene asociado retos importantes que todavía deben ser abordados. Se requiere la definición de mecanismos de interacción que soporten soluciones específicas para cada dominio de aplicación particular de IoT centrado en las personas. Esto implica que las aplicaciones puedan ser ajustadas a la personalidad y estado de ánimo del usuario [35]. Por tanto, como afirma Boavida et al. [35], el IoP integrado con los objetos de IoT implica que los objetos tengan la capacidad de adaptar su comportamiento de acuerdo a perfiles y cambios de las rutinas de las personas, aprendiendo de sus experiencias y tratando de evitar al máximo la intervención del usuario [179]. Un claro ejemplo de este tipo de comportamiento puede involucrar el sistema de calefacción del hogar de un usuario. Dicho sistema podría estar programado para que se encienda antes de que el usuario llegue a casa y así, cuando éste llegue pueda

percibir una temperatura confortable. Sin embargo, si el usuario sufre un contratiempo o cambio de rutina, el sistema de calefacción considerará automáticamente la situación y se mantendrá apagado hasta que el usuario realmente esté de regreso a casa. Así, se puede alcanzar un mayor nivel de confort y una mejor administración de los recursos energéticos y ahorro financiero.

Las acciones relacionadas con el paradigmas del IoP requieren de mecanismos de identificación y manejo de perfiles específicos de los usuarios, y el modelado de aspectos específicos en relación a la adaptabilidad, la interacción hombre-máquina, la seguridad y la privacidad de los datos, y la analítica de datos [35]. Para cubrir algunos de estos aspectos, la Computación basada en la Nube [92] ha sido empleada como un puente entre el IoT y el IoP para establecer una plataforma de comunicación y colaboración eficiente que explote todas las fuentes de información disponibles en Internet a través del IoS. De esta manera, las personas tienen a su alcance los recursos y contenidos (i.e., temática, multimedia) adecuados para llevar a cabo acciones más inteligentes que apoyen a los retos emergentes del IoT y del IoP [204].

1.6. Internet del Contenido/Recursos

En relación al contenido generado en la web, en las últimas décadas el Internet ha experimentado cambios radicales. De un Internet de Contenidos/Recursos (*Internet of Content/Recourses*, IoC por sus siglas en inglés) estático se ha pasado a un Internet de Contenidos/Recursos más complejo y dinámico donde objetos, servicios y personas juegan un importante rol y se constituyen como entidades de insumo primario de datos. Esto provocó que la incipiente web, conocida como web 1.0 evolucionara a la web 2.0 [69] o lo que se conoce como la web social. Sin embargo, la evolución de las tecnologías asociadas a Internet ha conllevado a nuevas implicaciones y cambios en torno a la web lo cual ha conducido a una posterior evolución.

Como ilustra la Figura 1.10, herramientas como las ontologías y los agentes inteligentes han permitido a la web social evolucionar hacia la web 3.0 (web semántica) y posteriormente hacia la web 4.0. [4]. Esta última propone una web inteligente donde a partir de contenido semántico es

posible generar conocimiento el cual es gestionado por agentes inteligentes para la toma de decisiones en el ciberespacio. De esta forma, Internet y la web en general estarán inundadas de miles y millones de agentes, los cuales interactuarán con los usuarios alrededor del mundo.

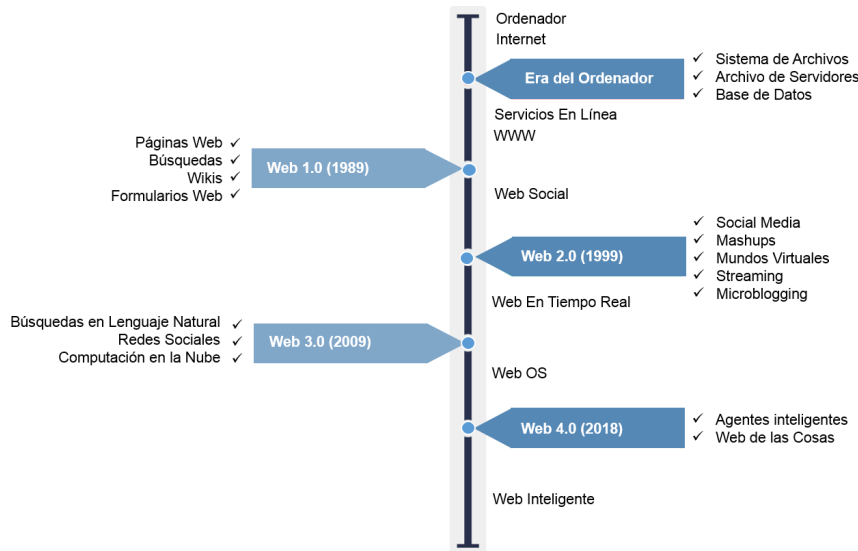


FIGURA 1.10: Evolución de la web y herramientas usadas en Internet.

El IoC como paradigma, hace referencia a toda la información consolidada, creada por entidades particulares (i.e., máquinas, personas) para aumentar el conocimiento sobre temáticas particulares [1, 257]. Aunque en los inicios de la web, para crear la información en ecosistemas de IoC, los usuarios se apoyaban de recursos como páginas web tradicionales, artículos, Wikipedia, vídeos, libros electrónicos y noticias; en la actualidad, el IoC no se limita únicamente a la generación de información por parte de las personas; sino que dichos contenidos pueden también ser generados por máquinas y agentes de software que están distribuidos en Internet.

El IoT es una de las tecnologías de soporte que ha permitido que los contenidos en la web dejaran de ser generados únicamente por personas. A través de millones de objetos de IoT conectados en escenarios del mundo real (i.e. hospitales, industrias, carreteras, puertos, centros educativos) la web actual está inundada de información. Multitud de sensores y objetos “inteligentes” obtienen datos, los procesan, almacenan

y los transmiten a través de Internet. Estos datos llegan a generar grandes bases de datos masivas [84] y son gestionados por el IoT, el IoS y el IoP, a través de los dispositivos, servicios y personas. Sin embargo, es importante resaltar que dichos datos son pocos significativos si no son procesados, integrados y fusionados de manera correcta. En este sentido, paradigmas emergentes como: la Computación en la Nube, el Big Data y la Inteligencia Artificial, han sido preponderantes para alcanzar una web inteligente como se visiona a largo plazo.

1.6.1. Computación Basada en la Nube

El aporte principal de este enfoque (conocido también como *Cloud Computing* por sus siglas en inglés) en el Futuro Internet gira alrededor de la definición de un modelo de computación que permite acceder a un conjunto de recursos informáticos compartidos y ofrecidos como servicios a través Internet [74]. Surgen así, modelos de servicios públicos y privados ofrecidos por gigantes tecnológicos (i.e., *Amazon*, *Microsoft* y *Google*) a través de la Nube. Algunos de estos servicios giran en torno a [53, 283]:

- Software como Servicio (*Software as a Service*, SaaS), que proporciona a los usuarios finales acceso a una aplicación específica como Microsoft Office 365, Youtube, Facebook.
- Plataforma como Servicio (*Platform as a Service*, PaaS)—que proporciona interfaces de programación de aplicaciones (API) y entornos de programación para desarrolladores de aplicaciones como Google Application Engine (GAE), Amazon Web Services y Microsoft Azure.
- Infraestructura como Servicio (*Infrastructure as a Service*, IaaS), que proporciona instalaciones de computación y almacenamiento a través de la virtualización utilizando marcos de trabajo como Amazon EC2 y S3.

Sin embargo, como ilustra la Figura 1.11, a partir de dichos servicios se han desarrollado nuevas prestaciones tecnológicas orientadas a entornos ubicuos (i.e., ecosistemas de IoT) que aprovechan la Computación en la Nube tales como: Sensado como Servicio (*Sensing as a Service*), Sensado

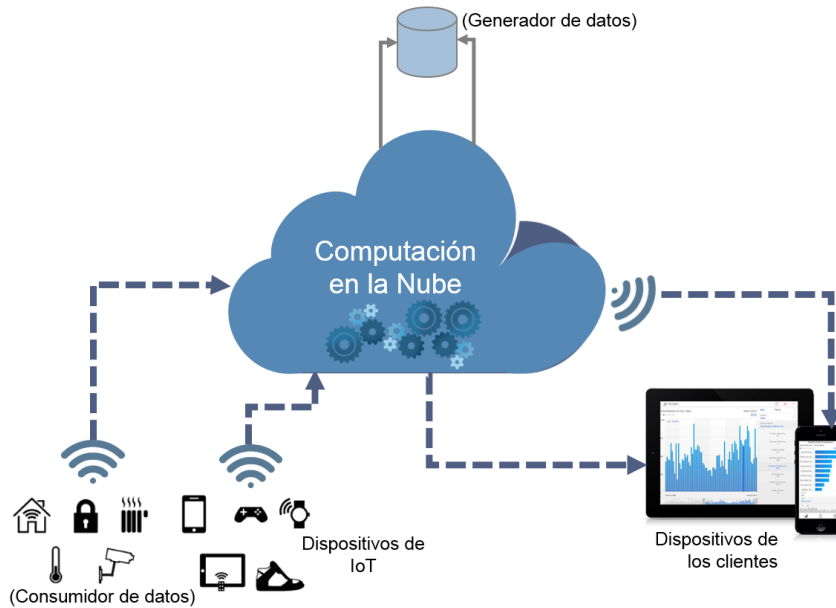


FIGURA 1.11: Rol de la Computación en la Nube en IoT.

y Control como Servicio (*Sensing and Actuation as a Service*), Objetos Inteligentes como Servicios (*Smart Object as a Service*), entre otros [53].

Los servicios anteriores proporcionan mecanismos sofisticados que los dispositivos generalmente no tienen para realizar tareas de almacenamiento, procesamiento y gestión de los datos capturados en los ecosistemas de IoT. Sin embargo, la aplicación de la computación en la nube en IoT tiene algunas limitaciones en términos de seguridad de datos, ancho de banda y latencia [243]. La computación en la nube proporciona una plataforma para centralizar servicios y datos. Esto significa que los sistemas son penalizados cuando tienen que acceder a la nube con frecuencia. Esto se convierte en un problema incontrolable en grandes ecosistemas como IoT. Por consiguiente, a fin de superar estas limitaciones y proporcionar servicios de IoT más eficientes y específicos para el usuario, han surgido dos nuevos enfoques para la gestión de datos en contextos de IoT. Dichos enfoques se denominan computación en la niebla (*Fog Computing* [37] por sus siglas en inglés) y computación en el borde (*Edge Computing* [229] por sus siglas en inglés).

Por un lado, en la computación en el borde la gestión de datos requerida por el cliente puede resolverse en el borde de la red cercana al cliente

integrando en la red local una pasarela encargada de procesar los datos sin necesidad de acceder directamente a la nube [243]. De esta manera, el nodo pasarela puede procesar datos de múltiples fuentes convirtiéndose en un productor de datos. Un ejemplo claro de este enfoque puede ser un teléfono inteligente, que puede desempeñar el papel de frontera entre los dispositivos de IoT y la nube. Por lo tanto, sólo los datos que deben estar disponibles deben ser cargados en la nube.

Por otro lado, en la computación en la niebla la gestión de datos necesaria para el cliente se realiza mediante infraestructuras informáticas en la red local cercana a las fuentes de datos [229]. De esta manera, las fuentes de datos pueden ser recogidas, analizadas y procesadas directamente por la infraestructura de niebla en lugar de por la nube. Solo en circunstancias específicas son enviadas a la nube. Por consiguiente, el bajo acceso a la nube reduce la latencia de estos sistemas [37]. Un ejemplo donde se aplica este enfoque puede ser un sistema de seguridad compuesto de varias cámaras que almacenan las imágenes capturadas de forma local y las procesan en la niebla para detectar anomalías (por ejemplo, la presencia de objetos peligrosos). De este modo, la conexión a la nube sólo se establece únicamente cuando es necesario enviar la información que justifica una incidencia.

1.6.2. Big Data

Por otro lado, el paradigma asociado a la Big Data provee un soporte tecnológico para el tratamiento eficiente de los datos masivos generados en Internet en un aceptable rango de tiempo [58]. De manera general, los datos masivos se generan a partir de transacciones en línea, correos electrónicos, vídeos, audios, imágenes, secuencias de clics, registros, mensajes, consultas de búsqueda, historias clínicas, interacciones en redes sociales, datos científicos, sensores y teléfonos móviles y sus aplicaciones [228]. Por tanto, el paradigma del *Big Data* debe proveer de mecanismos sofisticados que le permitan dar valor a los grandes volúmenes de datos distribuidos en *Terabyte*, *Pettabyte*, *Exabyte* y *Zettabyte*. Asimismo, dicho paradigma también debe ser capaz de gestionar la heterogeneidad de dichos datos; es decir, proporcionar mecanismos para almacenar y procesar datos estructurados, no estructurados y semi-estructurados. Adicionalmente, como se muestra en la Figura 1.12, son necesarios métodos y mecanismos automáticos orientados a realizar los siguientes procesos:

representación de datos distribuidos y escalables, control de redundancia y compresión de datos, análisis de datos y gestión de la confidencialidad de datos [58, 149, 190]. Estos mecanismos en conjunto con la Computación en la Nube aportan significativamente para la gestión de datos, generación de información y toma de decisiones en Internet.



FIGURA 1.12: Big Data generada en entornos de IoT.

1.6.3. Inteligencia Artificial

De forma complementaria al paradigma del Big Data empleado para realizar el análisis de datos (generalmente usando modelos estadísticos) [90], nuevos modelos fundamentados sobre las bases de la Inteligencia Artificial también se han empleado para modelar comportamientos inteligentes en el Futuro Internet, y específicamente en ecosistemas de IoT.

Algunas de las ramas de la Inteligencia Artificial que han sido empleadas en la optimización del comportamiento de objetos conectados a redes de IoT han sido las siguientes: Aprendizaje Automático [175, 217, 264], Computación Evolutiva [96, 189] y últimamente, los Sistemas Basados en Agentes [96, 98, 234]. Este último, ha sido planteado como una línea de investigación reciente entre los desarrolladores de sistemas para optimizar los recursos en ecosistemas de IoT [234] y que estos escalen hacia un comportamiento más proactivo, inteligente y colaborativo.

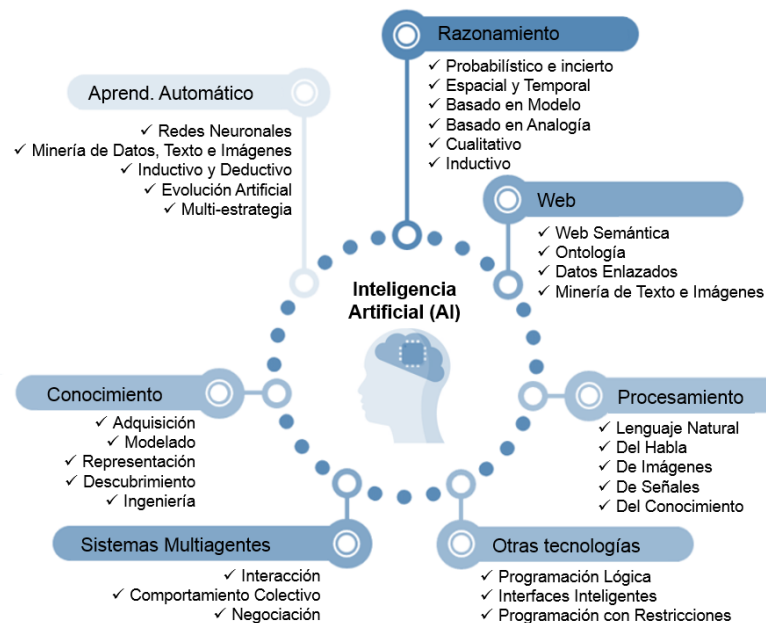


FIGURA 1.13: Usos de la Inteligencia Artificial en IoT.

1.7. Internet de los Agentes

Las tendencias relativas al IoT requieren que los objetos tengan la capacidad de comportarse de manera inteligente y autónoma. Sin embargo, en la actualidad, la mayoría de los objetos de IoT comercializados no soportan un mecanismo que permita definir de forma dinámica un comportamiento con estas características. Con el fin de suplir estas deficiencias en los últimos años una nueva línea de investigación dirigida a

lograr este objetivo, a través de la integración del IoT y las tecnologías de agentes, ha sido planteado [185]. Este nuevo enfoque se denomina Internet de los Agentes (*Internet of Agents*, IoA por sus siglas en inglés) [278].

Una contribución importante a esta línea de investigación es, sin duda, el concepto *Agent of Things* propuesto por Mzahm et al. [185]. En su estudio los autores sugieren que cada objeto conectado a una red de IoT tenga un agente software especial vinculado para añadirle capacidades como inteligencia y autonomía. Así, los objetos de IoT, tradicionalmente pasivos, pueden llegar a comportarse de manera proactiva [169]. A partir de esta idea se han propuesto otros enfoques relacionados, tales como objetos inteligentes basados en agentes [97, 160], modelos de IoT basados en agentes software [99, 135, 193] y Sistemas Multiagentes (MASs) para IoT [14, 158] (las bases de estos dos conceptos se describen en el Anexo A). Sobre las ideas planteadas en estas propuestas, muchas de ellas conceptuales, modelos más específicos también han sido implementado (i.e., *IoT-a* [52], *agent of thing* [187], y *smart object* [97]).

El Internet de los Agentes (Internet of Agents) constituye un paradigma en el que se plantea el modelado de la interacción de agentes inteligentes definida en un nivel superior que rige los recursos de IoT orientado a mitigar las deficiencias relacionadas con el razonamiento y las capacidades sociales en los escenarios de IoT [209]. El esquema lógico que implementa este planteamiento es ilustrado en la Figura 1.14.

Es importante rescatar que muchos de los trabajos en los que se ha analizado y abordado la integración de agentes e IoT ninguno enfatizaba específicamente en el término IoA. Fueron Yu et al. [278] quienes acuñaron el término *Internet of Agents* en su trabajo titulado “*From Internet of Things to Internet of Agents*”. Los autores plantearon al IoA como la evolución del IoT en la que los dispositivos inteligentes toman en cuenta el comportamiento humano. A partir de esta propuesta, las primeras aproximaciones que han tratado al IoA han sido planteadas. Los estudios más sobresalientes se han centrado esencialmente en el planteamiento de marcos conceptuales y de análisis [169] [278], modelado de mecanismos [198, 209] y desarrollo de aplicaciones prácticas [19, 46].

El IoA ha empezado a despertar interés entre los investigadores que trabajan en el área de IoT. Dos trabajos relevantes son los propuestos por Maamar et al. [169] y Predrag [215]. En el primero de ellos se presenta al

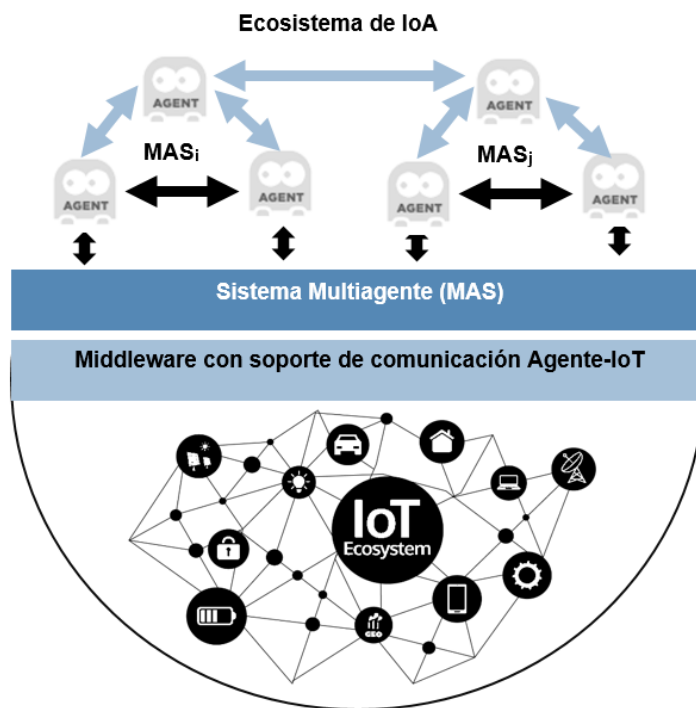


FIGURA 1.14: Siguiete generación del IoT.

IoA como el proceso de transición de los objetos de IoT, tradicionalmente pasivos, hacia agentes software los cuales conducen a un proceso de agenticación de dichos objetos empoderándolos para que puedan responder a los cambios de su entorno por sí mismo [169]. A partir de esta percepción, en el segundo estudio se ve al IoA como un paradigma orientado al desarrollo de programas informáticos y de inteligencia computacional del IoT para permitir que diferentes dispositivos, plataformas y tecnologías interactúen eficazmente entre sí, facilitando los procesos de comunicación, coordinación y diversas formas de colaboración [215].

En general, como afirma Gorodetsky [111], todos los trabajos que giran en torno al IoA están orientados hacia el modelado e implementación de sistemas de IoT en términos de agentes simples [101, 170]—y MASs [268]. Por ello, los agentes software y los MASs tienen un importante rol en la creación de máquinas inteligentes. No obstante, los modelos de agentes para IoT requieren que incorporen características adicionales

a las genéricas—adaptabilidad, proactividad, reactividad, racionalidad, sociabilidad y autonomía [26, 101, 271]—. Esto significa por tanto que los modelos de agentes orientados al IoA deben proveer mecanismos más sofisticados de seguridad, interoperabilidad, inteligencia, proactividad, autonomía, semántica, sociabilidad y colaboración orientados a dispositivos, soporte de movilidad, orientados al usuario, entre otras.

Gracias a las capacidades de los agentes software, el comportamiento de los objetos de IoT tradicionales adquiere mejoras significativas en términos de los siguientes aspectos [233, 234]: flexibilidad, colaboración, interoperabilidad, proactividad e inteligencia. Dichos beneficios son sólo algunos de los aspectos positivos que los agentes introducen en ecosistemas de IoT. El análisis realizado por Savaglio et al. [233] lo demuestra y afirma que “*la abstracción de agentes es un paradigma adecuado para infundir inteligencia y autonomía en un único objeto inteligente y, en consecuencia, para realizar sistemas de IoT distribuidos, auto-dirigidos y heterogéneos*”. Esta es una de las principales razones por la que los modelos de IoT basados en agentes se están empezando a popularizar y la comunidad científica cada vez más empieza a prestarle más atención.

1.7.1. Integración del IoP con el IoA

La interacción de las personas con el IoT requerida por el IoP puede simplificarse cuando se aplica el paradigma de IoA. El hecho de que los objetos agentificados de IoT puedan comportarse de manera pro-activa como ocurre con las personas, nos permitiría contar con procesos auto-personalizados y ejecutables sin necesidad de supervisión de personas en los ecosistemas de IoA. En este sentido, como el IoA propone el uso de agentes de tipo *agent of things* [186] para manejar objetos de IoT, también es factible introducir agentes de tipo *agent of people* [211] con el fin de establecer interacciones hombre-ordenador en los ecosistemas de IoT. Estos agentes son capaces de actuar como personas que abordan su comportamiento en base a perfiles de usuario (adaptables y personalizables en función del contexto, su estado emocional o las rutinas diarias). En consecuencia, los agentes pueden ejecutar tareas en los ecosistemas de IoT para realizar acciones exactas y precisas de forma similar a como lo hacen los humanos [273].

Por otro lado, la integración de ambos paradigmas permite establecer relaciones sociales y procesos de negociación entre personas y máquinas

en los ecosistemas de IoT sin demasiado esfuerzo. Esto se debe a que los agentes pueden utilizar protocolos de comunicación especializados, como el protocolo FIPA Contract Net Interaction Protocol [94], a partir del cual pueden negociar cómo utilizar los recursos de IoT. Por lo tanto, esto sería de considerable beneficio en el modelado y ejecución de procesos colaborativos sobre ecosistema de IoA.

1.8. Discusión

El Futuro Internet tiene una tendencia hacia un mundo real controlado a través de objetos con capacidad para conectarse a Internet. Por tanto, el Futuro Internet a corto plazo será soportado por un Internet de las Cosas (IoT) consolidado. Esto significa que los ecosistemas de IoT ofrecerán servicios orientados a dispositivos enfocados principalmente a mejorar la calidad de vida de las personas. Para ello se aprovecharán el máximo potencial de enfoques emergentes como son la Computación en la Nube, la *Big Data* y las Arquitecturas Orientadas a Servicios. No obstante, muchos de los servicios desarrollados a este nivel no podrían escalar hacia la pro-actividad, lo cual implica que el usuario juega un rol crucial en su configuración y operación en función de sus necesidades y nuevos requerimientos. Sin embargo, en los últimos años el Futuro Internet ha iniciado su tendencia hacia un IoT más autónomo, inteligente y colaborativo. Esta tendencia concibe a largo plazo ecosistemas de objetos inteligentes que adaptan de manera autónoma y proactiva su comportamiento de acuerdo a perfiles sociológicos de las personas. Así, los usuarios tendrán una mínima o nula intervención con la personalización de los dispositivos. Dichas acciones son gestionadas de manera automática mediante entidades autónomas e inteligentes como son los agentes software.

Si a corto plazo, el Futuro Internet visiona ecosistemas que interconectan millones (o billones) de objetos de IoT que generan datos masivos a partir de los cuales se toman las decisiones para controlar el mundo real; a largo plazo, el Futuro Internet visiona ecosistemas inteligentes en los que las personas van a interactuar y negociar con millones (o billones) de agentes software los cuales controlan los objetos de IoT y consecuentemente, el mundo real. Por tanto, el Futuro Internet tiene una tendencia marcada hacia el Internet de los Agentes (IoA)—donde las interacciones

usuario-máquina serán tan similares a las interacciones usuario-usuario que las personas no lo notarán.

1.9. Contribuciones relacionadas

Las ideas y planteamientos de este capítulo han sido evaluadas a través del proceso de evaluación por pares correspondiente a las siguientes contribuciones científicas:

P. Pico-Valencia, Managing the Heterogeneity in Multiagent Systems Based on Service Oriented Technologies. *Computación y Sistemas*, ISSN 2007-9737 (2018) [*En Revisión*].

P. Pico-Valencia and J.A. Holgado-Terriza, Integration of MultiAgent Systems with Resource-Oriented Architecture for Management of IoT-Objects. *Ambient Intelligence and Smart Environments*, 23, 2018. ISSN 978-1-61499-873-0.

J. A. Holgado-Terriza, P. Pico-Valencia and X. Quiñónez-Ku. Future Internet and the Next Generation of the Internet of Things: A Brief Survey. *IEEE Internet of Things Journal*. [*En Revisión*].

Capítulo 2

Agentificación del IoT: Una revisión de la literatura

“Investigar es ver lo que todo el mundo ha visto, y pensar lo que nadie más ha pensado”—A. Szent-Györgyi.

El Internet de las Cosas (IoT) es un paradigma emergente, que surge a final de los noventas con el objetivo de controlar el mundo físico mediante el uso de una red global de objetos que se encuentran interconectados a través de Internet [6, 38, 163]. A raíz de este novedoso enfoque, una amplia gama de objetos “inteligentes” compatible con interfaces de IoT son actualmente comercializados a fin de cubrir un amplio espectro de aplicaciones en diferentes dominios del conocimiento. No obstante, la alta oferta ha causado una proliferación de objetos (i.e., TVs, refrigeradoras, sistemas de ventilación)—aproximadamente 12.6 millones de dispositivos según proyecciones de Cisco [62]—que adoptan diferentes estándares y que al ser integrados son incompatibles. Por ello, el aspecto asociado a la interoperabilidad es un reto que el IoT aún debe afrontar [6].

Muchos de los objetos comercializados en la actualidad son cajas negras. Esto dificulta su adaptación y adición de mecanismos de inteligencia que cubra las necesidades específicas de las infraestructuras especializadas de IoT. En consecuencia, la comunidad científica ha prestado interés en investigar modelos, mecanismos y técnicas para llevar a cabo el modelado de objetos [97] y redes de IoT [14] basado en agentes. Este capítulo presenta una revisión de la Literatura en torno al proceso de agentificación del IoT visto como un proceso de integración para así crear redes de IoT inteligentes, pro-activas y colaborativas. Además, en este capítulo se definen las principales tendencias asociadas al proceso de agentificación del IoT.

2.1. Revisión sistemática

Un proceso de revisión sistemática consiste en identificar, evaluar e interpretar los estudios más relevantes de una temática, pregunta de investigación o fenómeno [138]. Por tanto, el objetivo general de un proceso de esta naturaleza es proporcionar evidencias que respondan a un conjunto de preguntas científicas a partir de estudios primarios en donde previamente se haya abordado la temática de estudio [144].

Para llevar cabo el proceso de revisión sistemática planteado en este capítulo, se siguieron las directrices metodológicas propuestas por Keele [138] para desarrollar revisiones de la Literatura en el ámbito de la Ingeniería de Software. Debido a la generalidad del proceso con estudios de la misma índole en el área de las Ciencias de la Computación como el propuesto por Kofod-Petersen [144] ha motivado su uso para dirigir el proceso de revisión sistemática planteado. La aplicación de dicho proceso se detalla a continuación.

2.1.1. Definición de las preguntas de investigación

En vista del reciente interés de la comunidad científica en torno al proceso de integración de las tecnologías de IoT con agentes para lograr un nivel de inteligencia/cognición en objetos y redes de objetos superior al alcanzado por el IoT genérico [234], se ha planteado una revisión sistemática de la literatura que intenta responder a seis preguntas de investigación (RQ) relevantes que no han sido respondidas por otros estudios. Dichas interrogantes, han orientado el proceso de revisión abordado en este trabajo y se detallan de la siguiente manera:

- *RQ*₁: ¿Cuál es la tendencia existente en torno al uso de agentes para modelar sistemas basados en tecnologías de IoT?
- *RQ*₂: ¿Cuáles son los principales enfoques propuestos para abordar el modelado de redes cognitivas basadas en tecnologías de IoT?
- *RQ*₃: ¿Cuáles son las principales propuestas en términos de modelos y herramientas tecnológicas que la comunidad científica ha planteado para integrar IoT y agentes software?
- *RQ*₄: ¿Existen aplicaciones de IoT basadas en agentes implementadas en escenarios para la solución de problemas reales?

- RQ_5 : ¿Qué aspectos adicionales al cognitivo se han experimentado a partir del proceso de agentificación del IoT?
- RQ_6 : ¿Cuáles son las principales técnicas de inteligencia usadas para el desarrollo de aplicaciones cognitivas de IoT modeladas usando agentes software?

2.1.2. Selección de las fuentes de información

Las preguntas de investigación previamente definidas, guiaron el proceso del estudio para determinar un conjunto de términos que han permitido ejecutar el proceso de búsqueda de los estudios primarios relacionados con el área investigada—proceso de integración del IoT con agentes software (agentificación). Los términos empleados se constituyen por las palabras (*'agent'*, *'software agent'*, *'multiagent'*), (*internet of things*) e (*'intelligent'*, *'smart'*). A partir de ellos, fueron definidas de manera estratégica cuatro cadenas de búsqueda (CB) que siguen la estructura detallada a continuación:

- CB_1 : (*'agent'* OR *'software agent'* OR *'multiagent'*) AND (*'internet of things'*),
- CB_2 : (*'agent'* AND *'integration'* AND *'internet of things'*),
- CB_3 : (*'internet of things'* AND *'based'* AND *'agent'*), y
- CB_4 : (*'smart'* OR *'intelligent'*) AND (*'internet of things'*).

Una vez definidas las cadenas de búsqueda, se procedió a determinar las fuentes de información empleadas para ejecutar las ecuaciones de búsqueda, y así recuperar los estudios más relevantes en los que el proceso de integración entre agentes e IoT ha sido abordado. Basado en los marcos metodológicos propuestos por Keele [138] y Kofod-Petersen [144], se consideró un total de cinco fuentes primarias de información (FPI), entre las que figuran tres librerías digitales especializadas en el área de Ciencias de la Computación (*IEEE Digital Library*, *ACM Digital Library*, *Science Direct*) y dos bases de datos documentales (*Web of Science* y *Scopus*). Las dos últimas fuentes han sido incluidas con el propósito de extender el rango de búsqueda y así, recuperar otro tipo de documentos del estilo patentes, reportes técnicos, entre otros.

2.1.3. Selección de estudios

El proceso de exploración llevado a cabo en octubre de 2017, se realizó sobre cada uno de los motores de búsqueda especificados anteriormente y se aplicó sobre los títulos de las publicaciones. El filtro sobre el título se aplicó debido a que existen publicados centenas de trabajos enmarcados en la línea de investigación de IoT y agentes que han sido aplicados para simular entornos que validan modelos propuestos en dicha área, los cuales para esta investigación no tienen relevancia alguna. No obstante, aspectos abordados por el IoT a través de agentes software como: sociabilidad, autonomía, proactividad, movilidad y adaptación, sí constituyen estudios de relevancia en este estudio siempre y cuando coadyuven en la ejecución de servicios y procesos inteligentes por parte de los objetos conectados a redes de IoT.

De manera general, el proceso de exploración realizado proporcionó un total de 239 estudios, de los cuales 92 no estaban repetidos (dato obtenido sin agrupar por cadena de búsqueda ni fuente de información). De estos 92 estudios, se redujo a 69 luego de analizar el título con detalle y descartar 23 estudios más. Finalmente, luego de analizar el resumen de cada artículo, se identificaron 13 propuestas que no constituían trabajos relevantes en el área de investigación y consecuentemente, el número de estudios considerados para ser analizados de manera integral fueron únicamente 56. De estos 56 estudios, considerados en primera instancia como primarios, luego de ser leídos completamente, 6 de ellos fueron considerados poco relevantes por no estar alineados a la temática investigada.

Los principales criterios de exclusión empleados en el proceso de selección de estudios primarios, que ayudaron a excluir los 6 artículos especificados previamente, se incluyen los siguientes: no disponibilidad del texto completo (1 manuscrito), no disponibilidad de la versión en inglés del manuscrito (2 en chino y 1 en ruso) y carencia de relación con la temática de estudio (los 2 restantes).

Finalmente, para el análisis de resultados en esta revisión sistemática, se consideraron un total de 50 estudios. Los resultados del proceso de selección se detallan en la Tabla 2.1 y la lista de artículos seleccionados están listados en el Anexo B. Dicho anexo, muestra el número de estudio, fuente, la cadena de búsqueda a partir de la cual fue recuperado, el año de publicación del estudio, el tipo de estudio (i.e., artículo de conferencia, revista y patente registrada) y su correspondiente referencia bibliográfica.

TABLA 2.1: Resultados de la búsqueda realizada

C. Búsqueda	Cad_1	Cad_2	Cad_3	Cad_4	Cad_5	Σ	Repetidos	Únicos
$Cadena_1$	24	22	5	50	44	145	58	87
$Cadena_2$	2	0	0	3	2	7	4	3
$Cadena_3$	6	1	3	21	20	51	23	28
$Cadena_4$	10	5	1	10	10	36	8	28
Σ	42	28	9	84	76	239		
Total repetidos	11	9	4	34	33			
Total únicos	31	19	5	50	43			

2.1.4. Resultados

A continuación, se analiza y discute el contenido de los estudios primarios listados en el Anexo B con el propósito de extraer información relevante en torno a la tendencia del enfoque de IoT basado en agentes en la última década. Los resultados, detallados en la Figura 2.1, demuestran que el número de publicaciones en las que el proceso de agentificación del IoT es abordado, tiene una tendencia creciente. Los inicios de este enfoque emergente tienen su primera aproximación en el año 2011. Sin embargo, su mayor repercusión se ha dado en los últimos cinco años donde se han publicado el 91 % de los estudios analizados.

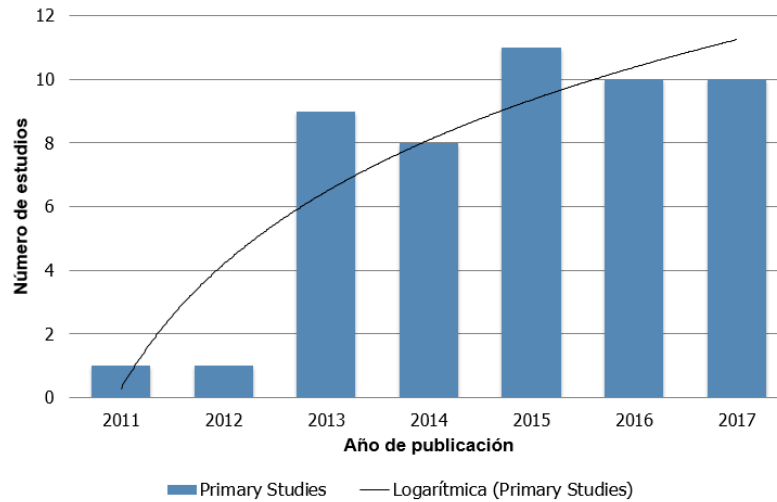


FIGURA 2.1: Tendencias en el proceso de agentificación del IoT

De los trabajos publicados a lo largo de periodo 2011-2017, como se detalla en la Figura 2.2 (b), el mayor número de estos estudios giran alrededor de modelos (experimentales y no experimentales) y aplicaciones implementadas en entornos reales (92 %). Dichos estudios fueron publicados en su mayoría en revistas y actas de conferencia científica (98 %) según muestra la Figura 2.2 (a).

Por otro lado, se evidencia también, que no existen hasta la fecha suficientes revisiones de la literatura que aborden esta temática (8 %). Asimismo, se observa que de la mayoría de estudios que giran alrededor de los modelos propuestos, el 36 % corresponden a modelos en los que se ha llevado a cabo un proceso de experimentación a través de presentación de resultados cuantitativos; mientras que otro 36 % de los modelos teóricos constituyen únicamente propuestas donde se plantean ideas, arquitecturas y marcos de trabajo que incluyen ideas innovadoras para integrar el IoT con agentes.

Adicionalmente, es posible apreciar que los modelos elaborados están empezando a ser adoptados en la solución de problemas reales (20 %) conllevando inclusive al desarrollo de patentes. Esto nos lleva a concluir que el proceso de agentificación del IoT empieza a tomar auge poco a poco y existe una alta probabilidad de que se fortalezca en los siguientes años debido al crecimiento progresivo del uso de dispositivos de IoT y la necesidad de que ellos puedan operar de manera autónoma, inteligente y adaptativa. De esta manera, la nueva generación de objetos de IoT constituirán un medio más efectivo para el desarrollo de las actividades del ser humano [235] de una manera más autónoma.

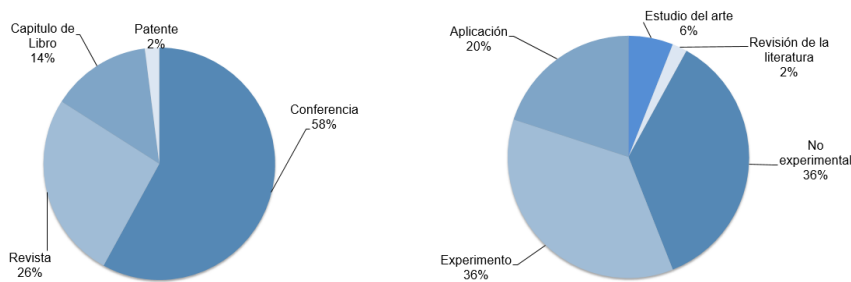


FIGURA 2.2: Naturaleza de los estudios (izquierda) y tipos de publicaciones (derecha) que abordan el proceso de integración de IoT y agentes

Finalmente, los modelos experimentales y no experimentales, publicados y enfocados en realizar la agentificación del IoT, demuestran el interés creciente de los investigadores para crear redes cognitivas de objetos de IoT. A partir de estos modelos, ha sido posible identificar dos enfoques para agentificar el IoT como se describe a continuación:

- *Modelado de objetos de IoT basado en agentes*. Este enfoque está orientado a modelar y diseñar objetos de IoT mediante la incorporación de un mecanismo de inteligencia y razonamiento que es embebido en el propio objeto. A partir de este artificio, es factible que el objeto pueda ejecutar directamente acciones que optimicen el uso de los recursos asociados a dichos objetos y su entorno [185].
- *Modelado de redes de IoT basado en Sistemas Multiagentes (MASs)*. Este enfoque consiste en modelar MASs para la gestión de redes de objetos compatibles con el enfoque del IoT. Así, este tipo de redes adquieren capacidades sociales y también de autonomía e inteligencia que de manera conjunta habilitan a los objetos de IoT para llevar a cabo de forma indirecta tareas colaborativas entre varios objetos de su mismo entorno [97].

2.2. Modelado de objetos con agentes

Este enfoque sugiere acercar algoritmos que definan comportamientos inteligentes al nivel del hardware usado por el IoT [52]. En otras palabras, el modelado de objetos de IoT basado en agentes consiste en embeber un agente software en el propio dispositivo para que éste adquiera las capacidades innatas inherente a los agentes y que así, el objeto pueda incorporar un comportamiento más dinámico, flexible y cognitivo que los actuales objetos genéricos de IoT comercializados.

En la Tabla 2.2 se resumen las principales propuestas planteadas que giran alrededor de este enfoque. En dicho resumen, se incluyen detalles como la unidad de procesamiento propuesta, el modelo de agente usado para su materialización, las tecnologías de IoT compatibles con el modelo propuesto y el alcance—en términos de inteligencia, sociabilidad, autonomía, proactividad, interoperabilidad, movilidad, adaptabilidad, desarrollo modular de los componentes, seguridad y privacidad, auto-adaptación y auto-organización—de las aplicaciones en cada una de las propuestas listadas.

TABLA 2.2: Principales propuestas que modelan objetos de IoT mediante agentes.

#	Enfoque	M. de agente	Tecnología IoT	Alcance	Referencia
1	Agent of Things	Teórico	Teórico	I, S, A	[185, 187]
2	IoT-a	Triskell3S	Raspberry Pi	S	[52]
3	Iota	Teórico	Teórico	I, S	[246]
4	Smart Object	JADE, JAdex	Omnet++	I, S, A, P	[98]
5	Moergestel et al.	JADE	Beagle Board	S	[180]
6	IoA	Teórico	Teórico	T, I	[278]
7	Manate et al.	Akka toolkit	General	T	[172]
8	Sol	Self-StarMAS	D. Android, motas SunSPOT,	AA, O, I	[24]
9	NFC-agent	Reactivo	NFC smartphones	S, P, R, I	[164]
10	AIoT device	Agente AIoT	LEGO Mindstorms NXT, Pa- rot AR. Drone	I, A, AO	[136]
11	REST agent	Agente REST	General	O	[158]
12	MAF	Agente MAF	No definido	S	[31]
13	REST agent	Agente REST	Smartphone, ATmega 1284P	I, O, M	[160]
14	HTML5 agent	Agente HTML5	Raspberry PI	M, T	[132]
15	AgentJS	Agente Java Script	Smartphone	T, M, D, I	[40]
16	PherCon	AgentSpace	Lego NXT	S, M, P	[109]
17	DLDA agent	Agente DLDA	Smartphone	S-P, I	[83]

I=Inteligencia, S=sociabilidad-colaboración, A=Autonomía, P=Proactividad, O=Interoperabilidad
T=Adaptabilidad, D=Modular, S-P=Seguridad y Privacidad, AA=Auto-Adaptación, AO=Auto-organización

2.2.1. Agente embebido en objetos de IoT

Los primeros diez ítem listados en la Tabla 2.2 corresponden a propuestas enfocadas a embeber un agente software en la arquitectura de los dispositivos u objetos compatibles con IoT. El proceso de embeber el agente en la arquitectura del propio dispositivo físico se puede realizar de dos formas. La primera consiste en el embebido directo de un agente y ocurre cuando el objeto es de naturaleza *open hardware*; es decir, que el objeto soporta la actualización del software que dirige su comportamiento. Esto implica la carga de un firmware que puede ser actualizado de forma dinámica (i.e., mangOH [173]).

Por otro lado, la segunda alternativa consiste en el embebido de agentes en los objetos de IoT por medio de mecanismos alternativos como los componentes SBC (*Single Board Computer*). Así, el micro-controlador usado en el primer caso, se cambia por un microprocesador más potente y con suficiente poder de cómputo y almacenamiento para ejecutar los algoritmos que usa un agente para controlar un conjunto de componentes físicos integrables en sus placas a través de entradas analógicas y digitales. Algunos proyectos relevantes en esta línea son *Raspberry PI*, *BeagleBoard*, entre otros [171].

En el intento de definir una unidad básica de procesamiento estándar para materializar el modelado del IoT basado en agentes se han planteado varias propuestas. Entre las unidades más relevantes figuran entidades del tipo *agent of things*, *IoT-a*, *Iota* y *AIoT*; acuñados respectivamente por Mzahm, Ahmad y Tang [185] y Mzahm et al. [187], Carlier y Renault [52], Singh y Chopra [246] y Kato, Takahashi y Kinoshita [136]. Es importante señalar que aunque estas entidades han sido planteadas bajo nombres distintos, comparten el mismo objetivo, el cual es integrar inteligencia en objetos para alcanzar redes de objetos cognitivos de IoT. Por este motivo, cada una de las unidades planteadas están siendo adoptadas por modelos, marcos de trabajo y herramientas tecnológicas que buscan alcanzar niveles de inteligencia y autonomía superiores a los conseguidos por las aplicaciones genéricas de IoT [102].

Los estudios previamente especificados, proponen de manera general el modelado de objetos para IoT mediante la vinculación de un agente individual. No obstante, aunque de manera similar, y en algunos casos de forma idéntica, cada autor define este proceso de integración desde su perspectiva. Por ejemplo, Mzahm et al. [187] y Mzahm, Ahmad

y Tang [185] lo conciben como un artificio que habilita a los objetos de capacidades para ejecutar por sí mismos procesos de razonamiento y de comunicación con objetos heterogéneos, proveyendo así un alto de nivel de interoperabilidad e inteligencia dinámica y flexible, personalizada a través de agentes software. Por su parte, Carlier y Renault [52], y Singh y Chopra [246] lo definen como una máquina unitaria conceptual que almacena la lógica de control del objeto enfocada también en la ejecución de tareas de razonamiento de manera similar a como plantean Mzahm et al. [187], y Mzahm, Ahmad y Tang [185]. En la misma línea, Kato, Takahashi y Kinoshita [136], y Knol et al. [143] proponen el diseño de objetos de IoT usando agentes para mejorar sus habilidades de comunicación y que de esta manera, los objetos puedan establecer procesos cooperativos con otros dispositivos [136] para compartir datos de forma coherente tanto en el ciberespacio como en el propio dispositivo [143].

Por otro lado, aunque las propuestas *agent of things* e *Iota* proveen de las directrices teóricas sobre cómo llevar a cabo el proceso de modelado de objetos basado en agentes sin definir el modelo del agente a usar. Kato, Takahashi y Kinoshita [136] recomiendan el uso de agentes basados en reglas *IF-THEN* a partir de las cuales los objetos son capaces de ejecutar acciones determinadas. Así, los objetos pueden ser configurados de manera similar a como operan las tecnologías IFTTT (IF-This, Then-That) en la web [124]. No obstante, es posible llegar a definir procesos cognitivos más complejos en donde se aplican capacidades deliberativas de naturaleza BDI (creencias, deseos e intenciones; *beliefs, desires e intentions* por sus siglas en inglés) [214], procesos de razonamiento semántico basados en ontologías y procesos de mayor complejidad basados en técnicas de Inteligencia Artificial (*Artificial Intelligence*, AI por sus siglas en inglés) (i.e., aprendizaje automático, redes neuronales artificiales, algoritmos genéticos, redes bayesianas, lógica difusa). En estos últimos casos, las propuestas están orientados a problemas más específicos que han considerado la naturaleza del problema y los recursos asociados a los objetos ya que de ello dependerá el uso de un determinado modelo de agente (reactivo, deliberativo, híbrido) [26] y la(s) técnica(s) de inteligencia a adoptar.

En la misma línea descrita previamente, en [98-100] se introduce un nuevo concepto denominado *smart object* el cual habilita a los objetos de IoT para que sean capaces de ejecutar acciones pro-activas, cooperativas

y la gestión inteligente de la sensibilidad al contexto. Esta última característica es alcanzada gracias a la capacidad reactiva asociada a los agentes. Además, con el fin de materializar el concepto propuesto, los autores incluyen mecanismos para que los objetos aprovechen de la Computación en la Nube [183] con la finalidad de dotarles de mayor capacidad de cómputo y de almacenamiento [98] los cuales constituyen recursos limitados de la mayoría de los actuales objetos de IoT. De manera análoga, Lin, Ho y Lin [164] proponen el desarrollo de un modelo de agente inteligente con capacidades para llevar a cabo la comunicación a través de la tecnología inalámbrica de campo cercano (*Near Field Communication*, NFC por sus siglas en inglés), que combina perfiles personales y sociales a partir de los cuales recomienda servicios relevantes que apoyan la acción reactiva, el logro pro-activo y la cooperación social por parte de los objetos de IoT—aspectos relevantes del enfoque denominado Internet Social de las Cosas (*Social IoT*, SIoT por sus siglas en inglés) [21].

Finalmente, Ayala, Amor y Fuentes [24] proponen embeber agentes en objetos heterogéneos de IoT con el propósito de que dichos agentes gestionen los recursos asociados a los objetos de IoT para proveer los servicios más adecuados dentro de ecosistemas de IoT. En resumen, la propuesta de Ayala, Amor y Fuentes [24], se centra en el uso de agentes de tipo *Self-StarMAS* que dan soporte a la interoperabilidad entre redes heterogéneas de IoT proveyendo comunicación flexible entre los objetos conectados, tal y como visiona Yu, Shen y Leung [278] quienes ven al IoT, no sólo como una red de objetos interconectados; sino como una red de agentes que gestionan dichos objetos de manera más flexible e inteligente a como se realiza en la actualidad.

2.2.2. Agentes web orientados a objetos de IoT

Generalmente, los objetos de IoT son dispositivos electrónicos que poseen capacidad limitada de recursos de cómputo y almacenamiento para soportar la ejecución de agentes sobre ellos. Para confrontar esta limitación, se han propuesto varios modelos de agentes alternativos a los descritos previamente y a los modelos tradicionales de agentes reactivos como JADE [28] y JADE Leap,[30] y agentes de tipo deliberativo como Jadex,[214] que son actualmente las plataformas más extendidas y populares para la creación de agentes. Dichos modelos de agentes están basados en tecnologías web.

La onceava propuesta detallada en la Tabla 2.2 constituye el principal modelo de agente web propuesto hasta la fecha para su uso en ecosistemas de IoT. Dicho modelo, propuesto por Leong y Lu [158], constituye un agente basado en tecnologías REST que utiliza el Protocolo de Transferencia de Hipertexto (HTTP) como protocolo de comunicación y mensajes codificados en lenguaje JSON (*JavaScript Object Notation*), lo cual habilita un mejor desempeño para llevar a cabo el intercambio de mensajes entre agentes que se ejecutan de manera distribuida. De esta manera, los agentes proveen de un mecanismo para el soporte de agentes livianos que no necesitan de un directorio central para encontrar otros objetos que le ayuden a cumplir sus objetivos. De allí que estos agentes provean de un alto nivel de interoperabilidad y facilidad para distribuir la inteligencia que el IoT puede aprovechar para el desarrollo de redes cognitivas basadas en agentes ligeros e interoperables en la web [122].

2.2.3. Agentes móviles embebidos en objetos de IoT

Bajo el mismo criterio de la necesidad de componentes ligeros que aportan en el alcance de un nivel superior de autonomía, cognición y flexibilidad de adaptación en IoT, varias propuestas de modelos de agentes ligeros con soporte de movilidad también han sido propuestas. En primer lugar, la propuesta de Leppänen et al. [159-161] está centrada en un modelo ligero de agente desarrollado a partir de tecnologías REST que están orientados principalmente a dispositivos de IoT heterogéneos de recursos de baja potencia y redes de amplia escala. Dichos agentes son expuestos en la web y así, la *Web of Things* [113, 282] resulta positivamente beneficiada en términos de operacionalización.

Por otro lado, bajo el mismo criterio de Leppänen et al., Jarvenpaa et al. [132] propusieron un modelo de agente móvil desarrollado a partir del uso de la tecnología web *HTML5* que le permite a los agentes moverse en diferentes dispositivos de IoT, habilitando su clonación para crear comportamientos de mayor complejidad y entornos mayormente distribuidos. Dichos comportamientos pueden ejecutarse de dos formas: mediante una interfaz de usuario dentro de un navegador o desde un servidor. El uso de una u otra alternativa depende de los recursos asociados a los objetos donde el agente debe migrar. En esta misma línea, Bosse [40] propone un modelo de agente móvil adicional denominado *AgentJS*. Dicho modelo ha

sido desarrollado mediante el lenguaje *Java Script* y por tanto es factible ejecutarlo en todas las plataformas incluyendo los navegadores web.

Finalmente, Godfrey, Jha y Nair [109] proponen un modelo de agente móvil que no solo se enfoca en habilitar las comunicaciones entre objetos conectados a varias redes de IoT; sino que también se orienta en habilitar a los objetos para que sean capaces de buscar los recursos adecuados y proporcionar servicios especializados a los dispositivos conectados a una red de esta naturaleza. En conclusión, los agentes móviles ligeros propuestos en este apartado aportan significativamente para que los objetos de IoT integren un nivel de experticia mediante componentes que no necesariamente deben estar incorporados indefinidamente en su arquitectura; pero que coadyuvan al cumplimiento de sus objetivos.

2.3. Modelado de redes de IoT con MASs

Este segundo enfoque propuesto para llevar a cabo el proceso de agenticación del IoT se centra en el modelado de MASs para gestionar los objetos conectados a infraestructuras IoT y sus recursos asociados. La gestión de las comunicaciones entre los agentes que formen parte de los MASs con los dispositivos físicos que componen la red de objetos es posible gracias al uso de middlewares especializados en proveer de una interfaz de comunicación con los objetos de IoT. Algunas herramientas de este tipo, detalladas por Razzaque et al. [220] son: TinySOA, ubiSOAP, ActorNet, UbiROAD, OpenHAB entre otros.

El modelado de MASs para la gestión de IoT, conocido también de forma genérica como Modelado del IoT Basado en Agentes (*Agent Based Modelling*, ABM por sus siglas en inglés), se enfoca básicamente en el modelado de capacidades sociales, de colaboración y de adaptación. Prueba de ello es la demostración hecha por Kaminski, Murphy y Marchetti [135] y Savaglio et al. [234] quienes en sus respectivos estudios y análisis determinan los principales beneficios del ABM como una técnica reconocida que puede ser considerado como un “*soporte eficaz y pleno para el desarrollo de sistemas de IoT abiertos, dinámicos, cooperativos y descentralizados*”.

En el proceso de revisión fueron recuperadas cinco propuestas relevantes (Tabla 2.3). Los dos primeros modelos son MASs que adoptan mecanismos sofisticados de AI.

TABLA 2.3: Principales modelos de objetos modelados mediante sistemas multiagentes.

#	Enfoque	M. de agente	Tecnología IoT	Alcance	Referencia
1	Machine Learning MASs	JADE, Jade, Leap	Ordenadores y Teléfonos inteligentes	I, AO, AA	[189]
2	Recommendation-MAS	General	General	I	[96]
3	LSMASs	General	General	O	[235]
4	Sensor Cloud	SeSAm	SoC microchip		[39]
5	MASMINC	JADE	Ordenador	B	[151]
6	DMA	Agente CMA	General	M	[60]

I=Inteligencia, O=Interoperabilidad, M=Movilidad, T=Adaptabilidad, AO=Auto-Organización
AA=Auto-adaptación, B=Escalabilidad

El primero modelo de la Tabla 2.3 detalla el enfoque propuesto por Nascimento y Lucena [189]. Su propuesta propone un modelo en el que los objetos adquieren capacidad de auto-organización y auto-adaptación a partir de agentes basados en técnicas de aprendizaje automático tales como redes neuronales y algoritmos evolutivos. En esta misma línea, el segundo enfoque propuesto por Forestiero [96] plantea un modelo de agente que usa un algoritmo multiagente de recomendación basado en el mapeo de objetos similares.

Respecto al modelado del componente social y cooperativo, también han sido planteadas algunas propuestas. Entre las propuestas más relevantes figuran los trabajos de Ruan, Durrezi y Alfantoukh [226], y Schatten, Ševa y Tomičić [235] (tercer ítem de la Tabla 2.3). Por un lado, Ruan et al. proponen un marco de trabajo orientado en gestionar el nivel de confianza de cada agente con el objetivo de ayudarles a evaluar la confiabilidad de sus homólogos. Por otro lado, Schatten, Ševa y Tomičić [235] proponen un estudio donde se plantean las directrices para llevar a cabo el diseño organizacional de MASs a gran escala que tienen limitaciones para comunicarse entre sí.

Adicional a los planteamientos anteriores, dos estudios extras enfocados en la gestión del componente social han sido presentados. En primer lugar, el estudio de Manate, Fortis y Negru [172] propone una plataforma multiagente híbrida donde los agentes pueden ser embebidos en los propios dispositivos o en la nube. De esta manera, los agentes pueden intercambiar información independientemente de su ubicación. De allí que la plataforma propuesta realiza una auditoría de seguridad sobre la arquitectura distribuida de agentes para determinar qué áreas geográficas ofrecen una mejor calidad de servicios para los usuarios finales. En segundo lugar, el trabajo presentado por Choi y Koh [60]—incluido como sexto ítem de la Tabla 2.3—propone un modelo de agente de movilidad distribuida (*Distributed Mobility Agent, DMA por sus siglas en inglés*) para gestionar la ubicación de los sensores móviles y realizar operaciones de control de traspaso con sus homólogos distribuidos en Internet.

Finalmente, en torno al componente cognitivo, como detalla el quinto ítem de la Tabla 2.3, Laghari y Niazi [151] exploran un modelado de redes de comunicaciones complejas basadas en agentes cognitivos a través de la plataforma MAS-MINC (*Multiagent System for Managing Carbon Footprint*) que también considera el aspecto de la cooperación. Sobre este

trasfondo, Singh y Chopra [246] proponen la integración de MASs como mecanismo de soporte a la distribución de la inteligencia apoyado en la Computación en la Nube, a fin de proveer procesos de razonamientos flexibles y optimizar el descubrimiento y selección de objetos de IoT y recursos asociados. Con un propósito similar, como se indica en el cuarto elemento de la Tabla 2.3, Bosse [39] propone un MAS basado en agentes móviles capaces de operar en redes heterogéneas de IoT de escala amplia. Esto ha dado lugar al desarrollo de nubes de sensores interoperables y con capacidades de razonamiento propio.

En síntesis, el modelado de MASs puede ser abordado combinándolo también con el primer enfoque de agentificación descrito previamente (modelado de objetos de IoT basado en agentes). El hecho de modelar objetos como agente habilita de forma automática el desarrollo de MAS para la gestión de redes de objetos de IoT y los recursos asociados a la infraestructura de manera global; es decir, contemplando el componente de cognición, capacidad social y de colaboración, adaptación y autonomía. No obstante, en vista de que previamente no se habían establecido estas bases conceptuales ni se había definido el artefacto de vincular un agente a cada dispositivo de IoT, varios de los middlewares especializados en abstraer la comunicación de bajo nivel con los dispositivos a través del uso de agentes han sido creados [220].

En el estudio realizado por Razzaque et al. [220], se presentaron 61 herramientas, de las cuales 10 de ellas han sido desarrollados para el soporte de agentes software. Específicamente, estas herramientas—*Impala*, *Smart messages*, *ActorNet*, *Agila*, *Ubiware*, *UbiROAD*, *AFME*, *MAPS*, *MASPO*T y *TinyMAPS*—soportan el uso de agentes móviles a fin de apoyar a dispositivos con recursos de computación limitados en la optimización del uso de la energía (30 %) y memoria (60 %), así como llevar a cabo la adaptación dinámica de redes heterogéneas de objetos de IoT (90 %), descubrimiento de servicios (80 %), descubrimiento de dispositivos distribuidos (70 %), gestión coherente del contexto (70 %), interoperabilidad (100 %) y comportamientos autónomos (100 %). En general, el objetivo se orienta en optimizar los procesos de asignación y gestión de recursos de las redes de IoT.

2.4. Agentificación del IoT en la práctica

El proceso de agentificación del IoT ha tenido una tendencia creciente durante los últimos cinco años. Aunque la Figura 2.2 muestra que varios de los estudios han sido modelos teóricos (8 %) y no experimentales (36 %); de manera contrapuesta, varios de los modelos han sido validados mediante experimentos y simulaciones (36 %), y otros han sido validados en entornos reales. A continuación, en la Tabla 2.4 se analizan varios de los experimentos que han modelado situaciones del mundo real (i.e., intersecciones de carreteras reales) mediante la agentificación del IoT.

En la Tabla 2.4, muestra 12 propuestas que resuelven problemas en 7 áreas distintas: gestión inteligente del transporte, gestión inteligente de servicios de entretenimiento, asistencia inteligente del cuidado de la salud, gestión inteligente del confort, procesos inteligentes de fabricación, gestión de actividades agrícolas y finalmente, gestión inteligente de logística. Dichas propuestas se apoyan en tecnologías de IoT—RFID, microordenadores (*Raspberry PI*), cámaras IP, teléfonos inteligentes, y sensores y actuadores a partir de los cuales manejan de manera inteligente, social y autónoma los recursos de objetos de IoT a través de agentes.

Es importante resaltar que los modelos de agentes usados son bastante heterogéneos. Dichos modelos van desde modelos reactivos basados en JADE hasta otros más especializados o específicos manejados por ciertas metodologías de agentes (i.e., *JASON*, *Gaia*, *MAPS* [166], *AFME* [184]) y otros modelos que surgen a raíz de la propia propuesta (i.e. *Self-SmartMAS*). Sin embargo, existen problemas de interoperabilidad entre ellos.

2.5. Modelos de cognición para agentificar el IoT

Las aplicaciones que introducen el concepto de agentificación en tecnologías de IoT, y resumidas en la Tabla 2.4, aprovechan de la capacidad de inteligencia inherentes a los modelos reactivo (propuestas 1, 3, 4, 5, 6, 8, 11) y deliberativo (propuestas 9, 12). No obstante, se pudo constatar que estos comportamientos genéricos se ven complementados de técnicas avanzadas de AI (propuestas 2, 3, 6), modelos basados en probabilidades (propuestas 2, 6, 12) y lógica semántica (propuestas 5, 10, 12).

TABLA 2.4: Principales aplicaciones reales modeladas a partir de IoT y agentes

#	Dominio	T. de inteligencia	M. de agente	Tecnología IoT	Referencia
1	Transporte	Reglas basada en contexto	Reactivo	RFID	[7]
2	Transporte	Técnicas de IA y modelo probabilístico		Wi-Fi, ZigBee, Bluetooth	[54]
3	Transporte	Técnicas de IA	JADE, Jade Leap	PC y teléfonos inteligentes	[189]
4	Entretenimiento	Reglas basada en contexto	Self-SmartMAS	NFC, RFID y teléfonos inteligentes	[24]
5	Salud	Lógica semántica	JADE	TCP/UDP	[258]
6	Salud	Modelo probabilístico	Reactivo	Sensor de peso	[104]
7	Confort	Técnicas de IA	No especificado	Dispositivos AlertMe	[11, 12]
8	Confort	Reglas	Reactivo	Raspberry PI	[3]
9	Confort	Razonamiento	JASON	Dispositivos Zigbee, Wifi y Bluetooth	[105]
10	Fabricación	Lógica semántica	Gaia	RFID	[14]
11	Agricultura	Reglas basada en contexto	JADE	RFID	[284]
12	Logística	Lógica semántica y modelo probabilístico	MAPS [166]	PCs, PDAs y cámaras	[265]

Es importante resaltar que cada vez es más común la combinación de varias de las técnicas previamente comentadas en un mismo sistema, como es el caso del uso de información contextual en el caso de la propuesta 12 que combina ontologías y un modelo de decisiones de Markov para recomendar de manera más acertada los servicios de IoT acorde a las necesidades de los usuarios. Un análisis más exhaustivo respecto a los mecanismos cognitivos empleados por las aplicaciones incluidas en la Tabla 2.4 se detalla a continuación.

2.5.1. Basada en reglas

El modelo de agente reactivo, en su mayoría implementado a partir de la herramienta JADE, emplea un mecanismo basado en eventos que es gestionado mediante un conjunto de reglas que se disparan de manera automática cuando se produce un cambio en el contexto donde operan los agentes. El uso de este tipo de agentes en redes de objetos de IoT está orientado básicamente en el modelado de reglas basadas en información contextual de los dispositivos a fin de coordinar de manera más eficaz los servicios ofrecidos. En esta línea, figuran las propuestas presentadas por Al-Sakran [7], Adhikaree, Makani y Yun [3], Ayala, Amor y Fuentes [24], y Zhou y Lou [284].

Por un lado, Adhikaree, Makani y Yun [3] emplean agentes domésticos inteligentes reactivos que actúan de manera autónoma a fin de minimizar los costos de electricidad para hogares inteligentes. Así mismo, Ayala, Amor y Fuentes [24] proponen el uso de agentes reactivos en la gestión de las actividades de un museo inteligente que se basan en información contextual proveída por sensores a fin de proveer servicios específicos tales como establecer rutas dentro del museo considerando la presencia de otros grupos en las diferentes áreas físicas de la instalación, y compartir y difundir información entre la guía y su grupo de visitantes.

Apoyado en las tecnologías RFID usadas por Adhikaree, Makani y Yun [3], Al-Sakran [7] propone también la combinación de éstas con agentes reactivos para monitorizar en tiempo real la identificación de vehículos en carreteras y minimizar así el número de accidentes. Finalmente, Zhou y Lou [284] propusieron un sistema de rastreo de carga inteligente que brinda al usuario de información en tiempo real sobre los vehículos y carga basado en agentes de reflejo que tienen una representación interna de su entorno que le ayuda a disparar las reglas de

acción-condición en función de la ubicación de la carga, además de establecer procesos de negociación cuando surgen problemas con la ruta de una carga específica.

2.5.2. Basada en componentes deliberativos

Únicamente se identificó una propuesta enmarcada en la introducción de componentes BDI para el modelado de la inteligencia en IoT. Esta propuesta constituye el trabajo de Gateau y Rykowski [105], quienes presentan un modelo para proveer confort inteligente basado en agentes deliberativos que formulan y negocian las necesidades y expectativas de los usuarios para proveer confort a través de la jerarquía de necesidades de Maslow (comodidad térmica, de ventilación, humedad, emisión de gases, iluminación y acústica).

Es importante señalar que el uso de este tipo de agentes en IoT es escaso debido a que son complejos de modelar y adaptar, y además no está orientado al modelado de ecosistemas de interacción en MASs de mediana y gran escala. Esta es una de las principales razones por las que el modelo de agente deliberativo no es el más empleado en la agentificación del IoT a pesar de que el nivel de cognición puede llegar a alcanzar un nivel superior a los agentes reactivos. De hecho, como afirma Moin [182], ni Jadex ni cualquier otro marco de agente de BDI de última generación podría operar de manera eficiente plataformas con recursos limitados como las de IoT.

2.5.3. Basada en lógica semántica

Este enfoque propone el diseño de agentes con capacidades para intercambiar información basada en ontologías las cuales representan la semántica de la información procesada por el sistema para ejecutar tareas automáticas a partir de procesos de razonamiento semántico. En esta línea, Alexakos y Kalogeras [14] han diseñado agentes con estas capacidades de manera que las decisiones se basan en el significado de los datos.

Para modelar agentes cognitivos basados en este tipo de estrategia se necesitan ontologías especializadas en el dominio en el que se enmarca la aplicación. Las ontologías empleadas permiten describir los principales

aspectos de IoT tales como dispositivos, recursos, entre otros. En este sentido, por ejemplo, para el caso propuesto por Alexakos y Kalogeras [14], limitado a un entorno de fabricación, se emplearon por un lado la ontología *P2Ontology* que organiza el conocimiento en entornos de fabricación y la ontología SSN (*Semantic Sensor Network Ontology*) especializada en organizar el conocimiento asociado al IoT.

En esta misma línea, Thangaraj et al. [258] proponen un enfoque similar de agente que actúa en base a lógica semántica y descriptores semánticos para llevar a cabo la gestión inteligente de pacientes (i.e., monitorización de signos vitales y diagnóstico de las enfermedades a partir de los datos clínicos del paciente). La ontología utilizada incluye los conceptos y propiedades de los signos vitales de los pacientes.

Finalmente, Wang, Zhu y Ma [265] propusieron otro modelo adicional donde los agentes basan su inteligencia a través de modelos semánticos. En este caso específico, los autores introdujeron ontologías con el propósito de organizar y administrar eficientemente diversos servicios proveídos por terceros en ecosistemas generales de IoT.

2.5.4. Basada en probabilidades

La cognición de los agentes bajo esta técnica se lleva a cabo a través de modelos probabilísticos y estocásticos tales como: modelos de decisiones de Markov, redes bayesianas, algoritmos genéticos, entre otros. De las aplicaciones listadas en la Tabla 2.4, Chamoso et al. [54] ha utilizado la técnica de redes bayesianas para estimar el flujo de trabajo más adecuado que debe ejecutarse en un entorno de IoT por parte de los agentes para optimizar la gestión del flujo de tráfico en ciudades inteligentes de forma automática. Por otro lado, los autores también incorporaron agentes de peso liviano que se basan en técnicas de inteligencia colectiva modelada a partir de las interacciones entre los agentes para ejecutar comportamientos colaborativos.

En la misma línea seguida por Chamoso et al. [54], Nascimento y Lucena [189] emplearon algoritmos genéticos los cuales se basan en procesos de evolución usando técnicas estocásticas en los procesos de cruce y mutación de la población a fin de que los objetos adquieran capacidad de auto-adaptación. Adicionalmente, García-Magariño, Lacuesta y Lloret [104] presentan una cama inteligente *ABS-BedIoT* que permite el reconocimiento de las posturas del sueño de un durmiente y genera archivos

de texto que son analizados mediante técnicas de Big Data para detectar ciertas posturas y la cantidad de movimiento durante el sueño. Así, es posible detectar posturas perjudiciales en pacientes crónicos. Finalmente, Wang, Zhu y Ma [265] emplean un Proceso de Decisión de Markov (PDM) que le permite a los agentes ejecutar, organizar y administrar eficientemente diversos servicios de IoT.

2.5.5. Basada en Inteligencia Artificial

La capacidad cognitiva modelada bajo esta técnica se consigue incorporando agentes inteligentes de peso pesado que imitan ciertas capacidades cognitivas humanas estudiadas en el área de la AI—contrarias a los agentes ligeros orientados a objetos de IoT.

Las aplicaciones 2, 3 y 6 listadas en la Tabla 2.4, adoptan esta filosofía con el fin de añadir el componente de inteligencia basado en la inteligencia humana sobre el IoT. Así, Chamoso et al. [54] proponen dos modelos de agentes, uno de peso pesado y otro de peso liviano (descrito previamente en la categoría de agentes que emplean modelos de probabilidades). El primero de ellos se basa en técnicas de AI. De esta manera, el agente constituye una entidad especializada en fusionar información automáticamente a través de aprendizaje supervisado; es decir, el modelo emplea una red neuronal artificial y clasificadores que combinan diferentes tecnologías de detección que proporcionan datos heterogéneos y brindan información más precisa sobre los recursos de IoT. En este tipo de modelos, la inteligencia reside en cada uno de los agentes que participan en el sistema por lo que desde el punto de vista del cómputo son menos eficientes y por tanto no están orientados a dispositivos de IoT; sino a infraestructuras de Computación en la Nube que estén dotadas de alto poder de cómputo para ejecutar este tipo de tareas.

En esta misma línea, otros trabajos que adoptan otras técnicas de AI han sido presentados. Algunos de ellos son las propuestas de Nascimento y Lucena [189], Ramchurn et al. [218] y Alan et al. [12]. En el primer caso, se combinan nuevamente las redes neuronales artificiales con los algoritmos evolutivos a fin de que los objetos adquieran capacidad de auto-organización y auto-adaptación. En el segundo caso, el modelo de agente planteado usa técnicas de aprendizaje automático para obtener el estado de la situación o estado de un entorno donde se produjo un

desastre natural, a partir de datos capturados por Vehículo Aéreo No Tripulado (*Inmanned Aerial Vehicle*, UAV por sus siglas en inglés).

Los dos casos anteriores representan aplicaciones prácticas donde las acciones cognitivas de los agentes son modeladas a través de técnicas sofisticadas de AI. No obstante, siguiendo el mismo enfoque pueden ser propuestos modelos adicionales donde no se combinen varias técnicas. Este es el caso de Alan et al. [12] quienes emplean técnicas de aprendizaje automático para predecir el consumo diario de energía en un hogar domótico basado en el consumo del día anterior y así, planear a qué tarifa y cuándo cambiarla para predecir la tarifa más económica.

2.6. Tendencias de la agentificación del IoT

El análisis de las tendencias y retos del proceso de agentificación se ha realizado desde dos aristas. El primer caso es aquel en donde se propone llevar a cabo la agentificación del IoT embebiendo agentes en los propios objetos (2.6.1 - 2.6.4) y el segundo caso es aquél en el que el proceso de agentificación del IoT es realizado por medio de MASs (2.6.5 - 2.6.7). Los detalles de las tendencias alrededor de estos enfoques se resumen como sigue:

2.6.1. Modelos de agentes ligeros para IoT

Las actuales plataformas de agentes no son lo suficiente ligeras para desarrollar componentes que deben ejecutarse en dispositivos con recursos limitados como los objetos de IoT. Esta es una de las principales debilidades y por ello algunas propuestas han desarrollado sus propios modelos de agentes adoptando tecnologías web ligeras. Por tanto, es necesario avanzar en el desarrollo de nuevas plataformas con agentes ligeros y ultraligeros, aprovechando por una parte la posibilidad de utilizar sistemas operativos con un *memory footprint* más pequeño gracias a la capacidad de componentización, y con posibilidad de ejecutar un entorno multi-tarea en dispositivos con procesadores de bajo rendimiento y limitada capacidad de almacenamiento.

En la línea de los agentes ligeros y ultra-ligeros, se visiona que los agentes del futuro tengan la capacidad de trabajar en entornos fijos y

móviles. Por ello, se debe dotar a los agentes de capacidades que le permitan moverse entre dispositivos heterogéneos de IoT para coadyuvar en procesos donde un objetivo común se cumple a partir de un conjunto de agentes especializados. Por tanto, se requiere también de mecanismos para el soporte de movilidad ligera y aunque algunas propuestas de esta naturaleza han sido ya abordadas, es necesario que se estandaricen y el código fuente esté disponible para que los científicos y programadores puedan usarlas y contribuir para su mejora. Es importante señalar, que los modelos creados también deben ahondar esfuerzos en proveer de mecanismos de seguridad y de la gestión de la privacidad de los datos a fin de brindar agentes de confianza.

2.6.2. Objetos de IoT *open hardware*

Una de las principales limitaciones de muchos de los objetos de IoT (i.e., electrodomésticos) comercializados hoy en día es que son distribuidos como cajas negras donde el usuario puede generalmente parametrizar el objeto para cambiar su comportamiento; pero no tiene la opción de adaptarlo o re-programarlo para que cumpla con sus requerimientos en su totalidad. En este sentido, y aprovechando de la madurez y la reducción de los costos de las tecnologías SBC (i.e., *Raspberry PI*, *BeagleBoard*) es posible una nueva línea de trabajo enfocada en la creación de objetos físicos de IoT distribuidos bajo licencia *open hardware*. De esta manera, el proceso de agentificación basado en el embebido de agentes en los propios objetos podrá materializarse en corto/mediano plazo ya que dichos objetos podrán manejar comportamientos flexibles, autónomos, reactivos, sociales e inteligentes, de manera innata e independiente de infraestructuras basadas en la nube.

2.6.3. Agentes amigables con el usuario

En el ámbito de las tecnologías orientadas a agentes la interacción entre los agentes software y el usuario es de suma importancia. Por tanto, la línea enfocada en cubrir este aspecto debe abordar aspectos como el desarrollo de mecanismos que permitan la creación y personalización de los comportamientos de los agentes en tiempo de ejecución, de manera sencilla y sin necesidad de que el usuario posea conocimientos técnicos para llevarlo a cabo. Sin embargo, aunque esta capacidad de adaptación

es inherente a los agentes, se debe ahondar esfuerzos para crear técnicas de resolución de los conflictos en entornos multi-usuarios, los cuales se originan luego de introducir cambios en el perfil de los usuarios.

2.6.4. Plataformas de agentes interoperables

En lo referente a la interoperabilidad de las comunicaciones entre agentes, una línea que abarque este aspecto puede desarrollarse a fin de se establezcan estándares que habiliten la interoperabilidad técnica (recursos compartidos e interfaces de comunicación), sintáctica (formato de mensajes compartidos) y semántica (ontología y representación de conocimiento compartidos) Savaglio et al. [234].

Aunque la interoperabilidad sintáctica ha sido conseguida gracias a FIPA, es importante plantear nuevas directrices en mira de la interoperabilidad total. Por ello, las tendencias deberán estar enmarcadas en la aplicación de estándares actuales como los empleados en la web junto con las tecnologías semánticas como ontologías y datos enlazados para que todos los agentes manejen el mismo vocabulario y se alcancen mejores niveles de interoperabilidad semántica en sistemas heterogéneos.

2.6.5. Servicios basados en la nube

Una de las principales limitaciones inherentes a los objetos de IoT es su falta de potencia de cálculo y capacidad de almacenamiento necesaria para ejecutar procesos inteligentes. En este sentido, es posible aprovechar las ventajas que ofrecen las tecnologías de Computación en la Nube para ejecutar procesos complejos y almacenar grandes volúmenes de datos. Los agentes pueden basar su comportamiento tanto en técnicas de inteligencia simples (i.e., reglas) como en otras más complejas (i.e., técnicas de inteligencia artificial, modelos estocásticos). Así, la complejidad de los agentes reside en la nube y los agentes sólo se encargarían de realizar peticiones a los servicios de la nube.

2.6.6. Metodologías especializadas

Complementario a los mecanismos y métodos propuestos para realizar la agentificación del IoT, es importante desarrollar instrumentos metodológicos tales como métodos sistemáticos, metodologías y ciclos de vida

que sirvan de directrices para llevar a cabo el proceso de agentificación desde una perspectiva práctica. En la actualidad, no se ha propuesto ninguna herramienta de este tipo. Por lo tanto, ha sido un factor limitante por el que el proceso de agentificación que no haya sido ampliamente adoptado para el desarrollo de aplicaciones cognitivas de IoT y su uso en entornos inteligentes.

2.6.7. Software especializado

De forma similar al aspecto anterior, es importante desarrollar modelos de agentes orientados al IoT y las respectivas herramientas de apoyo. Los modelos presentados en la Tabla 2.2 son significativos en este sentido. Sin embargo, muchos de ellos son sólo marcos conceptuales que describen los términos principales y los procesos asociados. Algunos modelos que se han desarrollado y probado mediante experimentos y estudios de casos no tienen una descripción de los middlewares empleados o desarrollados, lo que dificulta a otros programadores la puesta en marcha de proyectos de agentificación del IoT. En este contexto, se recomienda no sólo el desarrollo de herramientas tecnológicas, sino también que estén documentadas y disponibles bajo licencia de código abierto. Sólo mediante la publicación y documentación de las herramientas propuestas podrá consolidarse y extenderse el proceso de agentificación del IoT entre la comunidad de desarrolladores de sistemas de IoT.

2.6.8. Repositorios de servicios públicos

El uso de servicios y recursos útiles en ecosistemas de IoT agentificados, a través de arquitecturas SOA y ROA, es de suma importancia en la próxima generación del IoT. En este sentido, los ecosistemas públicos de servicios y recursos deben ser implementados para que cualquier persona que desee utilizarlos pueda hacerlo luego de la respectiva autenticación. Sin embargo, estos componentes son necesarios para proporcionar datos seguros y deben estar siempre disponibles. En resumen, esto ayudaría en términos de reutilización de componentes y también aceleraría el desarrollo de sistemas de IoT agentificados.

2.6.9. Definición de patrones y anti-patrones

La definición de patrones de interacción y anti-patrones que determinen los procesos de comunicación en ecosistemas de IoT agentificados puede ser pertinente para guiar a los desarrolladores en el modelado de tales sistemas. Por lo tanto, se recomienda que se creen patrones de comportamiento de los agentes sobre los objetos de IoT y los usuarios como base para las herramientas metodológicas y tecnológicas. Esto facilitaría la comprensión, el modelado y la actualización de los comportamientos micro y macroscópicos de los sistemas agentificados de IoT.

2.6.10. Descentralización de la gestión de datos

Para los grandes ecosistemas de IoT, es importante crear nodos pasarela (*gateway*) que soporten la computación de borde. Esto permite que los objetos de IoT pasen de ser meros consumidores de datos a productores y consumidores de datos. De esta manera, la gestión de datos y servicios puede optimizarse en términos de fiabilidad, seguridad y protección de la privacidad. Además, se pueden ofrecer servicios especializados cercanos a los usuarios en función de su contexto, perfil de usuario o estado emocional. Esta visión podría lograrse mediante la creación de pasarelas agentificadas específicas para IoT que distribuyan la inteligencia sobre ecosistemas de IoA, como se discute en Singh y Chopra [246].

2.6.11. Mecanismos de seguridad y privacidad

Finalmente, es importante cubrir el aspecto relacionado con la seguridad y la privacidad de los datos manejados. Por tanto, se plantea una nueva línea en la que se establezcan protocolos de comunicación de agentes a nivel de inter-plataforma e intra-plataforma a fin de proveer de mecanismos seguros a los usuarios que usan agentes en IoT. Además, es importante que los agentes trabajen en función de mecanismos que gestionen el consentimiento de los datos en función del usuario tomando en cuenta el contexto donde son ejecutadas las operaciones y procesos [192, 197].

La gestión del consentimiento del usuario es elemental en ecosistemas de agentes a gran escala y distribuidos debido a que los datos de los usuarios rigen sobre una legislación distinta acorde al país donde se ejecuten

los procesos. De allí, la importancia de que se establezcan mecanismos de consentimiento informado para el acceso y manipulación de los datos en cada momento que sea requerido y que el usuario tenga conocimiento de lo que está sucediendo con la información que él genera por si mismo o a través de dispositivos vestibles que lleve integrado a su cuerpo.

2.7. Discusión

A partir de la revisión sistemática realizada se pudo constatar la existencia de varias investigaciones en las que se han introducido las directrices para llevar a cabo el proceso de agentificación del IoT. Los principales estudios, propuesto en los últimos 5 años (2013-2017) muestran una tendencia creciente (RQ_1) y se espera que con mejoras en las tecnologías de agentes respecto a la comunicación a nivel de intra-plataforma e inter-plataforma, las aplicaciones que adopten este enfoque crezcan de forma exponencial en los próximos años. También, por medio de las fuentes de información analizadas en este estudio fue posible identificar dos enfoques elementales que se pueden usar para integrar IoT y agentes software. Dichos enfoques proponen básicamente el modelado de objetos embebiendo un agente en el propio dispositivo y el modelado de redes completas de objetos de IoT mediante el uso de MASs (RQ_2). A partir de dichos enfoques no solo ha sido factible modelar comportamientos cognitivos; sino también, crear comportamientos sociales y colaborativos, control de privacidad, adaptación y autonomía. Todos ellos, demandados por las redes inteligentes de IoT.

Se pudo constatar también que el IoT tiene una tendencia hacia un paradigma emergente denominado Internet de los Agentes (IoA). Se visiona el IoT como un enfoque apoyado por agentes inteligentes que gestionen de forma eficaz los recursos ofrecidos en IoT tanto para objetos como para los usuarios. También se espera que entidades como: *agent of things*, *IoT-a*, *Iota* y *smart-object* faciliten el desarrollo de redes de objetos de IoT con alto grado de inteligencia y que además sean capaces de interoperar con agentes heterogéneos (RQ_3). De allí, la importancia de que las propuestas conceptuales se materialicen en middlewares prácticos que permitan el desarrollo de aplicaciones cognitivas de IoT pues las actuales propuestas prácticas implementadas en entornos reales que han adoptado este enfoque son relativamente pocas, y muchas de ellas quedan en

experimentos y simulaciones (RQ_4). Por ello, se considera oportuno el desarrollo de marcos metodológicos y herramientas que adopten el proceso de agentificación del IoT para que los desarrolladores tengan las directrices que le orienten cuando empleen este paradigma. Así, el proceso de agentificación del IoT puede llegar a extenderse, popularizarse y lograr establecerse como un estándar para crear redes cognitivas de IoT.

Por otro lado, la investigación también ha permitido indagar cuáles son las principales técnicas usadas por el proceso de agentificación del IoT. Se identificaron principalmente cinco alternativas tales como: capacidad cognitiva modelada a partir de reglas, componentes deliberativos, lógica semántica, modelos probabilísticos y aplicando técnicas de IA, la cuales son empleadas de manera individual y de manera combinada (RQ_6). La estrategia empleada depende básicamente de los objetivos planteados, de la complejidad del problema, de la big data generada por los objetos de IoT y del tipo de tecnología IoT empleada. Además, es importante resaltar que la introducción de agentes software en arquitecturas IoT habilita capacidades para que los objetos sean más autónomos, proactivos, interoperables con objetos heterogéneos, adaptativos a nuevas necesidades del usuario, colaborativos con ecosistemas de IoT heterogéneos, y auto-adaptativos y auto-organización gracias a los mecanismos de aprendizaje que los agentes puedan incluir (RQ_5).

Finalmente, se presentó un análisis de los principales retos en torno al proceso de agentificación del IoT para que el IoA llegue a alcanzar un mayor nivel de consolidación y aceptación para el desarrollo de redes inteligentes de IoT.

2.8. Contribuciones relacionadas

El proceso de revisión sistemática abordada en este capítulo proporcionó como resultado un artículo de revista. Los detalles de esta contribución se detallan a continuación:

P. Pico-Valencia, and J.A. Holgado-Terriza, *Agentification of the Internet of Things: A Systematic Literature Review*, International Journal of Distributed Sensor Networks, ISSN 1550-1477 (2018).

Capítulo 3

Internet de los Agentes: Un enfoque semántico

“Lo bueno de la ciencia es que es verdad, creas o no en ella”—Neil deGrasse Tyson.

En los últimos años, se ha planteado una línea de investigación emergente asociada al Internet de las Cosas (IoT) enfocado en la creación de dispositivos que sean capaces de comportarse de manera inteligente, autónoma, pro-activa y social, mediante el uso de agentes software y sistemas multi-agentes [185]. Este paradigma, denominado Internet de los Agentes (IoA) [278], y materializado a través del proceso de agentificación del IoT constituye un paradigma del Futuro Internet que surge como una alternativa enfocada en mitigar las limitaciones inherentes al IoT concernientes a la autonomía, cognición y capacidades socio-colaborativas [186, 209]. Su principal objetivo está centrado en el modelado de objetos inteligentes compatibles con las interfaces de IoT usando agentes software.

En este capítulo se describen las principales directrices en relación al IoA. Se presentan las principales definiciones, así como los principales aspectos que los agentes deben describir para que los objetos de IoT vinculados a dichos agentes puedan ser gestionados, y además se presenta un análisis del modelo de agente reactivo frente al modelo deliberativo para determinar la arquitectura/modelo de agente más idóneo para la creación de ecosistemas de IoA. Finalmente, se presentan situaciones reales en donde los ecosistemas de IoA pueden ser aplicados para resolver problemas en dominios, en los que las tecnologías de IoT están ganando terreno (i.e., cuidado de la salud, ciudad e industria inteligente).

3.1. Internet de los Agentes

Hasta la fecha no ha sido planteada una definición formal que determine el objetivo y alcance del IoA. Aunque no existe una definición formal y aceptada a nivel global que describa este paradigma, se presentan en orden cronológico algunas de las primeras aproximaciones publicadas en la Literatura y que han sido recuperadas en esta investigación en mayo de 2018. Todas las definiciones planteadas, a continuación, han hecho énfasis al IoA ya sea de manera teórica o práctica, empírica o experimental.

Definición 1. El primer acercamiento al IoA fue realizado por Yu et al. (2013) [278] en su trabajo “*From Internet of Things to Internet of Agents*”. En su estudio, los autores establecen que el IoA es *evolución del IoT en la que los dispositivos y los agentes son personalizados y creados por los usuarios finales, aún sin tener conocimiento técnico en el área de las tecnologías orientadas a agentes*.

Luego, en el año 2017, cinco trabajos adicionales realizaron hincapié en el tema. Como resultado, en estos estudios se plantearon seis definiciones:

Definición 2. En primer lugar, Alqithami y Lützenberger (2017) [15] conciben al IoA como una “*tendencia en curso para que el IoT se adapte a la naturaleza/arquitectura inteligente de los MASs*” .

Definición 3. En segundo lugar, Aref y Tran (2017) [18, 19] visionan al IoA como “*un área emergente que pretende combinar las ventajas de los sistemas multiagentes y del Internet de las Cosas, añadiendo autonomía e inteligencia a los objetos tontos utilizados tradicionalmente en el IoT*”.

Definición 4. En tercer lugar, Bui y Jung (2017) [46], ven al IoA como el resultado de la *combinación entre el IoT y los MAS que permite al IoT adaptarse a la arquitectura inteligente para aumentar la autonomía y flexibilidad en los entornos y consecuentemente mejorar la integración y la interoperabilidad entre sistemas heterogéneos*.

Definición 5. En cuarto lugar, y desde la perspectiva de Gorodetsky (2017) [111], el IoA constituye la “*acción de modelar e implementar aplicaciones de IoT en términos de agentes autónomos y Sistemas Multi-agentes*”.

Definición 6. En quinto lugar, Predrag (2017) [215] define al IoA como *la base de programas informáticos y de inteligencia computacional del IoT para permitir que diferentes dispositivos, plataformas y tecnologías interactúen eficazmente entre sí, y así, facilitar la comunicación, la coordinación y diversas formas de cooperación.*

Definición 7. En sexto lugar, Tumminelli y Battle (2017) Tumminelli y Battle [262] encasillan al IoA como *una sociedad de agentes considerada como una extensión del paradigma del IoT para modelar sistemas autónomos en los que los agentes cooperan y negocian para resolver tareas dentro de una red abierta.*

Finalmente, las últimas aproximaciones en relación al IoA que han sido planteadas en 2018, año en el que se realizó este análisis, incluyen dos definiciones adicionales a las anteriores:

Definición 8. Desde la óptica de Maamar et al. (2018) [169], el IoA es el proceso de *transición de las cosas pasivas a agentes que conducen a un proceso de agentificación de las cosas que empodera a las cosas para que respondan a su entorno.*

Definición 9. Finalmente, Shen et al (2018) [241] establecen un símil del IoA con la definición propuesta por Yu et al. (2013) [278] respecto al acercamiento de los agentes a los usuarios para que puedan personalizarlos en tiempo de ejecución y crearlos sin conocimiento técnico.

Las definiciones especificadas anteriormente, obtenidas en estudios preliminares, contemplan la fusión de dos campos, el IoT y las tecnologías orientadas a agentes. En la Figura 3.1 se ilustra dos conjuntos, cada uno representando a los campos especificados. La intersección de los dos conjuntos Tecnologías Orientadas a Agentes (creación de sistemas auto-organizativos e inteligentes) e IoT (creación de redes de dispositivos interconectados a Internet) dan lugar al Internet de los Agentes. Considerando el alcance de estos dos campos interceptados, el IoA tiene como

objetivo principal la creación de redes de agentes con soporte de razonamiento los cuales tienen vinculados objetos de IoT para gestionarlos y controlarlos de la manera más inteligente y precisa posible.

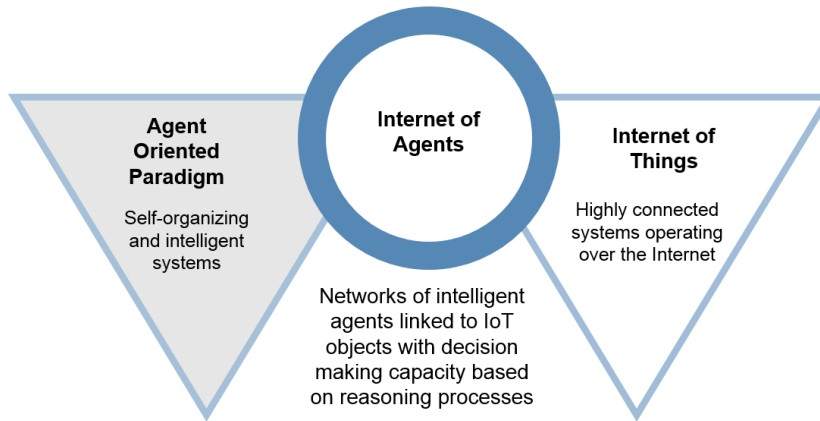


FIGURA 3.1: Áreas que dan origen al IoA.

3.2. Un enfoque semántico del IoA

Las definiciones planteadas previamente definen al IoA como un paradigma que fusiona las tecnologías del IoT y las tecnologías orientadas a agentes (Figura 3.1). Sin embargo, como se ilustra en la Figura 3.2, en esta investigación se integra un tercer paradigma, la Web Semántica. Entonces, se define al IoA como la “*interacción de agentes inteligentes, definida en un nivel superior que rigen los recursos de IoT aplicando procesos de razonamiento semántico para ejecutar acciones con un mayor nivel de inteligencia y precisión, aprovechando así las capacidades sociales intrínseca en escenarios de IoT*” [209].

El uso de tecnologías relacionadas con la Web Semántica en IoT habilita a los agentes para que los procesos de toma de decisiones sobre ecosistemas de IoT se realicen en función del significado de las cosas y no únicamente en base a la sintaxis (comparación de cadenas y valores nominales). De esta forma se habilita en IoA la interoperabilidad semántica; es decir, que los agentes que operan en los sistemas multiagentes

llegan a ser capaces de interactuar y comunicarse con otros agentes que se ejecutan en sistemas multiagentes heterogéneos [51, 128].

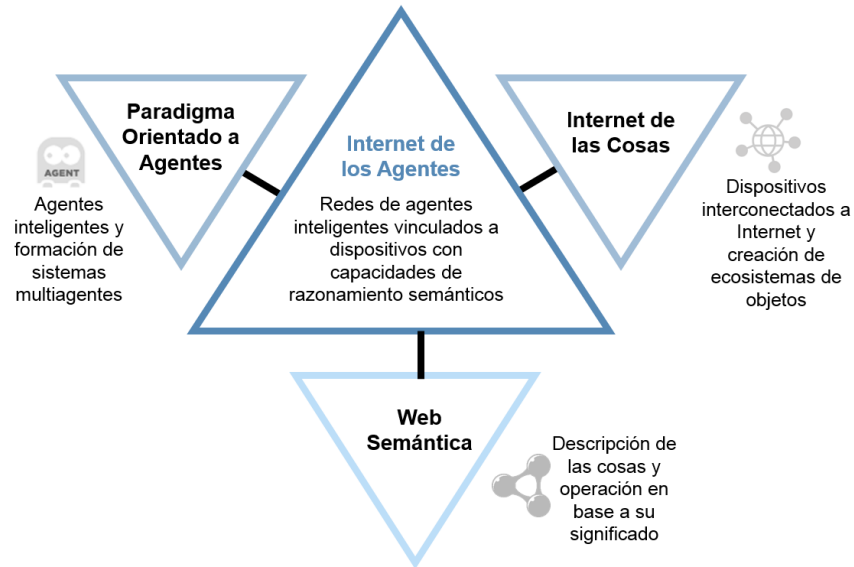


FIGURA 3.2: Áreas que dan origen al IoA semántico

La toma de decisiones en IoA pueden ser llevados a cabo mediante el uso de razonadores semánticos. No obstante, es necesario que los elementos principales que forman parte de los ecosistemas de IoA tales como agentes y objetos de IoT sean descritos semánticamente. Para realizar la descripción de dichos recursos es necesario definir una misma especificación para que las diferentes plataformas de IoA distribuidas en Internet puedan interoperar empleando el mismo vocabulario.

3.3. Modelando el IoA

El IoA se basa en el proceso de agentificación del IoT, a través del cual mejora el nivel de autonomía e inteligencia en entornos genéricos de IoT que generalmente operan bajo ciertas reglas predefinidas. Algunos trabajos relacionados como las propuestas de Mzahn et al. [185, 187], Fortino et al. [97, 99], Nieves et al. [193], Leong et al. [158], Al-Sakran et al. [7], Kaminski et al. [135] y Xu et al. [276] han modelado el proceso

de agentificación; pero no cubren todos los aspectos importantes que permiten alcanzar el objetivo perseguido por el IoA. Para definir estos aspectos, se ha analizado el alcance de las principales propuestas que están relacionadas con la integración de los agentes y el IoT.

3.3.1. Principales aspectos

Para mejorar el grado de autonomía e interoperabilidad sobre diferentes tipos de objetos de IoT y para poder actuar de forma pro-activa e inteligente sobre los ecosistemas de IoT, los agentes que controlan este tipo de ecosistemas necesitan tener un conocimiento detallado de las propiedades que caracterizan el modelo de agente, el contexto en el que se ejecuta, el tipo de objetos y recursos de IoT que gestiona, y las entidades con las que pueden colaborar para alcanzar sus objetivos, entre otros. Estos aspectos son ilustrados a continuación, en la Figura 3.3.

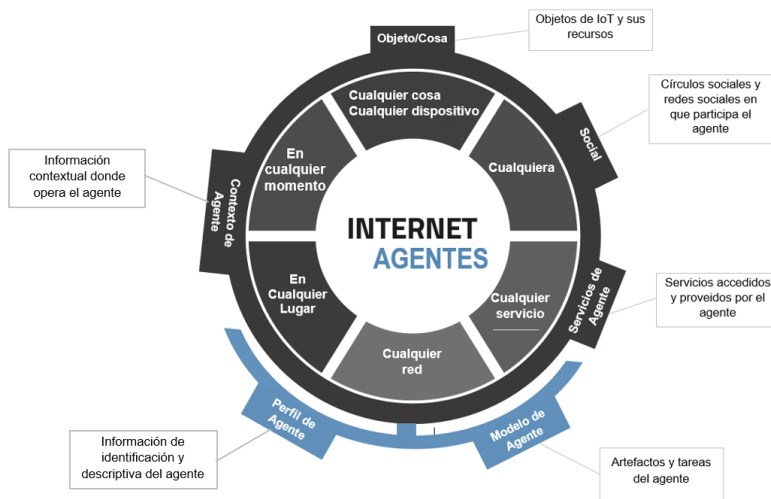


FIGURA 3.3: Aspectos asociados con el IoA.

Para identificar los principales modelos de objetos de IoT basados en agentes propuestos hasta la fecha hemos llevado a cabo un proceso de revisión de la Literatura. Los resultados obtenidos permitieron definir seis aspectos (perfil del agente, datos contextuales, contexto social, ecosistema de servicios, artefactos del agente y recursos de IoT). Algunos de estos aspectos son detallados en la Tabla 3.1.

TABLA 3.1: Uso de los aspectos asociados al IoA.

#	Modelo para agentificación	Profile	Context	Social	Service	Model	Thing
1	Agent of Things [186]	No	Sí	No	No	No	Sí
2	IoT-a [52]	No	Sí	Sí	Sí	No	No
3	Smart object [99]	No	Sí	Sí	Sí	No	No
4	Knol et al. [143]	No	No	No	No	No	Sí
5	Manate et al. [172]	No	Sí	No	Sí	No	Sí
6	Sol [24]	Sí	Sí	No	Sí	Sí	No
7	NFC-agent [164]	No	Sí	Sí	No	No	No
8	Alot device [136]	No	Sí	Sí	Sí	No	Sí
9	Rest agent [158]	No	Sí	No	No	No	No
10	MAF [32]	No	No	Sí	No	No	No
11	Rest agent [160]	No	Sí	Sí	Sí	No	Sí
12	HTML5 agent [132]	No	No	No	Sí	No	No
13	AgentJS [40]	No	No	Sí	Sí	No	No
14	PherCon [109]	No	Sí	Sí	Sí	No	No
15	DLDA agent [83]	Sí	Sí	No	No	No	No

El contenido de la Tabla 3.1 muestra qué aspectos han sido utilizados por los principales modelos de IoT basados en agentes. El análisis de los quince modelos estudiados confirmó que el uso de descriptores relacionados con los seis aspectos propuestos—*Profile*, *Context*, *Social*, *Service*, *Model* y *Thing*—ayuda a la ejecución de los procesos de toma de decisiones en ecosistemas de IoT. Por ejemplo, se han utilizado datos contextuales para ejecutar acciones en función de la ubicación de los objetos de IoT; se han utilizado datos de recursos de los objetos de IoT (i.e., nivel de batería) para supervisar el estado de estos objetos; entre otros. Sin embargo, ninguno de los modelos incluidos en la Tabla 3.1 cubre los seis aspectos de manera integrada. Esto se debe a que cada modelo ha sido diseñado para objetivos muy específicos.

Es evidente que los aspectos relacionados con el contexto, redes sociales y servicios son los más utilizados para describir a los agentes que gestionan el IoT. También destacamos que el cuarto aspecto, “cosa”, utilizado por este tipo de sistemas está relacionado con la descripción de objetos de IoT. Finalmente, los aspectos menos significativos son los relacionados con el perfil y el modelo de agente. Sin embargo, estos últimos aspectos han sido considerados potencialmente relevantes por esta propuesta porque, por un lado, los metadatos del perfil del agente contienen información útil para definir la identificación y el círculo social de un agente y, por otro, los metadatos almacenados por el aspecto modelo guardan información de utilidad con respecto a la estructura interna del agente (i.e., comportamientos y sus tiempos de activación) a partir de los cuales el comportamiento del agente puede ser definido y generado automáticamente sin necesidad de que sea programado manualmente.

Apoyados en las buenas prácticas adoptadas por los modelos listados en la Tabla 3.1, consideramos que a medida que se utilizan más descriptores para describir los ecosistemas de IoA y sus elementos, se proporcionan más datos que sirven de insumo y que los agentes pueden emplear para tomar decisiones de manera más precisas sobre dichos ecosistemas.

3.3.2. Información del perfil de agente

La adición de información no funcional es útil para describir las características y capacidades de los agentes. A partir de esta información pueden descubrirse agentes que estén ejecutándose en MASs heterogéneos [24]. Por lo tanto, consideramos que ésta es una información relevante que

debe incluirse para describir ecosistemas de IoA. Por tanto, esta cuestión se ha hecho necesario el modelado de un perfil de agente *AgentProfile*.

Ante los desafíos que plantea el IoT en relación con la confianza, la seguridad, la fiabilidad y la interoperabilidad semántica [23, 106, 178], el *AgentProfile* debe describir los siguientes aspectos:

- Identificación, estado y solicitudes que el agente ha completado con éxito.
- Credenciales del agente para establecer comunicaciones seguras cumpliendo con el protocolo de Autenticación, Autorización y Contabilidad (AAA) [199] de forma transparente, para que los agentes puedan definir su identidad cuando se inicia un proceso de comunicación
- Un conjunto de reglas que indican cómo crear agentes seguros.
- Un conjunto de parámetros de entrada que permiten al agente operar de forma controlada en la plataforma en la que se aloja.
- Un conjunto de restricciones de tiempo real que permiten conocer la calidad del servicio ofrecido por los agentes a la hora de proporcionar datos a sus contrapartes o al momento de cambiar el estado de uno o varios actuadores.

3.3.3. Información de sensibilidad al contexto

La gestión de los datos contextuales es muy importante en IoT y, consecuentemente en IoA. Para introducir información contextual de los agentes y sus objetos de IoT vinculados, hemos considerado necesario modelar el aspecto relacionado con su localización y geolocalización. Este aspecto que hemos llamado *AgentContext* es útil porque el proceso de descubrimiento de agentes puede ejecutarse con mayor precisión y además porque los agentes pueden adaptarse en función de sus condiciones contextuales. Teniendo en cuenta los desafíos, tales como la detección, la semántica y la inteligencia [23, 106, 107], los siguientes aspectos deben ser modelados:

- Información de geolocalización desde la que un agente puede actuar en función de su propia posición geográfica y de la geolocalización de los objetos que controla.

- Información de localización de los objetos gestionados por cada agente que le permite ejecutar acciones en función de la posición de los objetos respecto a otros objetos o entidades de referencia.
- Información asociada a la zona horaria en la que están instalados los agentes y los objetos para ejecutar las acciones temporales programadas.
- Información relativa a la plataforma tecnológica en la que se ejecutan los agentes.

Toda la información previamente especificada de manera conjunta, permite a cualquier entidad (i.e., agente, objeto de IoT, usuario) enviar solicitudes de información o acciones de detección/control con mayor precisión.

3.3.4. Información del ámbito social

Las redes sociales han conducido a la aparición de un nuevo enfoque llamado *Internet Social de las Cosas*. Este enfoque se ha propuesto para que los agentes puedan gestionar sus propias redes de colaboración, similares a las redes sociales humanas [22, 163, 270]. Basándonos en este supuesto, hemos encontrado útil modelar un nuevo aspecto al que hemos llamamos *AgentSocial*. Las propiedades asociadas a este aspecto, permitirán en los ecosistemas de IoA, modelar redes sociales a partir de las relaciones que surgen de la interconexión de dispositivos controlados por agentes. Así, el proceso de descubrimiento de agentes puede modelarse a partir del poder social de los agentes en cada una de las redes en las que participan.

La información relacionada con este aspecto, y que por consiguiente, debe ser tomada en cuenta para la definición de este aspecto en particular, se resume a continuación:

- Temas de interés relacionados con el objetivo del agente.
- Datos y unidades de medida que le interesan recibir o proporcionar al agente entre sus contrapartes potenciales que forman parte de la red social global de IoA.
- Lista de contrapartes o socios del agente que colaboran con el agente modelado para lograr sus objetivos de control sobre el objeto que gestiona.

- Conjunto de medidas de centralidad asociadas a los agentes que permiten conocer el poder de cada agente en la red social global de IoA en la que colabora.

3.3.5. Información del ecosistema de servicios

El aspecto *AgentService* cubre aspectos funcionales del IoT como la computación, la gestión de servicios y la composición de los servicios [6, 23, 163, 270] para que el IoA pueda mantener el soporte de SOA [87], ROA [168] o ambos [115].

En términos generales, consideramos que este aspecto debería permitir modelar las acciones de los agentes en base a los modelos de servicios web más extendidos en la actualidad:

- Servicios web SOAP (*Simple Object Access Protocol*) [87] desplegados principalmente por aplicaciones SOA para el desarrollo de procesos de negocio.
- Servicios web RESTful (*Representational State Transfer*) [259] desplegados en aplicaciones REST, objetos y marcos de trabajo especializados de IoT.
- Servicios web DPWS (*Device Profile for Web Service*) [275], generalmente desplegados en dispositivos como impresoras, sensores, entre otros.
- Además, sugerimos que se cubran otros modelos de servicios web específicos para la Computación Pervasiva y el IoT, que no son tan populares como los tres anteriores, como es el caso de los servicios web DOHA (*Dynamic Open Home-Automation*) [222]—un modelo de servicio gestiona la adquisición y fusión de datos basándose en un modelo de composición de servicios en cada dispositivo de IoT en función de su capacidad de cálculo.

3.3.6. Información de los artefactos de agentes

Debido a que la funcionalidad de los agentes está determinada por los artefactos que integra, hemos considerado relevante describir los elementos que constituyen parte de la arquitectura del propio agente, sin importar el modelo que se emplee para su implementación (i.e. reactivo,

deliberativo o híbrido). Para ello, hemos integrado un aspecto adicional denominado *AgentModel*. Los metadatos que este aspecto incluye es dependiente de la arquitectura del agente modelado. Sin embargo, nos hemos enfocado únicamente en el modelo reactivo por las razones descritas en la sección “Agentes orientados a IoA” del Anexo A.

En el caso específico de los agentes reactivos, éstos deben tener sensibilidad a los estímulos detectados en el entorno en el que operan [227]. Por tanto, deben describirse elementos como los descritos a continuación:

- Comportamientos como un conjunto de acciones (i.e., invocaciones de servicios web para acceder a objetos de IoT, operaciones de comunicación con otros agentes contrapartes) asociadas con un agente reactivo que se ejecuta cuando se produce un evento en su entorno.
- Un conjunto de parámetros temporales que determinan la frecuencia de ejecución del comportamiento del agente así como su tiempo máximo de ejecución.
- Flujos de trabajo como mecanismos que abordan las acciones de control del propio agente sobre el ecosistema de IoT.

3.3.7. Información de los recursos de IoT

Por último, considerando que cada objeto de IoT tiene un agente vinculado a él, hemos considerado necesario describir también sus recursos asociados. Por eso hemos incluido un nuevo aspecto llamado *AgentSmartThing*. En términos generales, se recomienda que para este componente se prepare la información de las siguientes propiedades:

- Aspectos técnicos de la red a la que está conectado el objeto de IoT, como el host, el puerto, el nombre lógico del objeto, entre otros.
- Lista de funcionalidades/operaciones soportadas por el objeto de IoT, así como los datos y unidades de medida proporcionados por el propio objeto.
- Recursos de cómputo asociados al objeto de IoT tales como: interfaces de comunicación compatibles, nivel de batería, capacidad de memoria, ancho de banda y procesamiento, entre otros.
- Finalmente, la información contextual (similar a la información contextual del agente) en la que se encuentra operando el objeto de IoT.

3.4. Potenciales aplicaciones

Existe una diversidad de áreas potenciales en los que el IoT es factible de ser aplicado—ciudad inteligente, industria conectada, edificios conectados, coches conectados, energía inteligente, salud inteligente, cadena de suministro inteligente, agricultura inteligente, tienda inteligente y otros que engloban aplicaciones en dominios complementarios. El número de aplicaciones en los dominios especificados tiene en su mayoría proyecciones de tendencia de crecimiento según el Portal IoT *Analytic*. No obstante, nuestro análisis cubre únicamente dos de estas áreas—ciudades inteligentes y cuidado de la salud. Dichas áreas han sido consideradas y analizadas también en el trabajo de Borgia [38], y en el estudio de Pico-Valencia et al. [205]. El último trabajo incluye un análisis de casos de aplicación del IoA en las dos áreas de estudio visto desde la arista de la autonomía e inteligencia.

3.4.1. Ciudades inteligentes

Un sistema orientado a una ciudad inteligente tiene como objetivo mejorar la calidad de vida de los ciudadanos a través de la optimización de los servicios públicos (i.e., transporte, iluminación, vigilancia) y de las infraestructuras físicas (i.e., accesibilidad, saneamiento, parques) [61, 280]. Las principales sub-áreas generales cubiertas por este campo incluyen: seguridad pública y vigilancia, hogares y edificios inteligentes, redes eléctricas inteligentes y movilidad y turismo inteligentes [38]. Para estos sub-ámbitos, se ha optimizado una amplia gama de servicios públicos tradicionales utilizando tecnologías de IoT. Algunos de estos servicios, determinados por Zanella et al. [280], son los siguientes: sistemas de iluminación inteligentes, control de calidad ambiental, reciclaje inteligente y gestión de residuos, gestión del tráfico y aparcamientos inteligentes, gestión del consumo energético urbano, automatización de edificios públicos, etc.

A continuación, se presentan dos posibles comportamientos de sistemas IoT junto con su equivalente comportamiento en IoA:

- *Sistema de gestión de servicios de guía turístico basado en IoT.*
Una aplicación de turismo electrónico orientado a recomendar las

atracciones en las cuales un turista está interesado basado en las preferencias y la situación actual del lugar visitado [247].

- *Sistema equivalente basado en IoA.* Una aplicación inteligente de guía turística especializada en proporcionar al turista una lista de atractivos recuperados a partir de sus preferencias, la duración de la visita y la geolocalización actual del visitante. Además, a través de agentes especializados, es posible calcular la ruta más corta y establecer comunicación con agentes externos que ofrecen reservar/comprar las entradas para los lugares de pago seleccionados por el visitante, como museos, zoológicos, teatros, conciertos, reservas de albergues, transporte, tour y estadios [205].
- *Sistema de monitorización de parámetros medioambientales basado en IoT.* Un sistema de monitorización de parámetros ambientales importantes como la temperatura, la humedad y el CO₂ dentro de una ciudad inteligente [239].
 - *Sistema equivalente basado en IoA.* Un sistema de monitorización inteligente basado en sensores heterogéneos conectados a diferentes redes que recogen datos meteorológicos (i.e., humedad, temperatura ambiental, CO₂) con el fin de optimizar el informe de monitorización de la calidad del aire midiendo los niveles de contaminación en el exterior mediante el uso de drones. Dichos drones, pueden generar informes de predicción en función de la información histórica almacenada para constituir un árbol de decisión centrado en la evaluación de riesgos en el área.

3.4.2. Cuidado de la salud

Los sistemas de asistencia médica permiten la gestión de la información médica electrónica asociada a los pacientes para proporcionar un diagnóstico oportuno y monitorizar, en tiempo real y en situaciones seguras, las condiciones fisiológicas (i.e., latidos cardíacos, presión arterial, temperatura) de los pacientes en diferentes contextos (i.e., dentro del hospital, residencias de ancianos y en el hogar) con el fin de mejorar su calidad de vida [9, 129, 153].

El estudio de Borgia [38] hace hincapié en dos sub-áreas generales de aplicación en las que las tecnologías de IoT se están empleando en la asistencia de salud, como la medicina y la asistencia sanitaria, y la vida independiente [38]. En estas sub-áreas se han desarrollado servicios centrados en la atención especializada del paciente. Algunos de ellos se resumen a continuación: monitorización de condiciones fisiológicas, asistencia para pacientes con enfermedades crónicas, cuidado de ancianos, programas de acondicionamiento físico, etc.

A continuación, se presentan dos posibles comportamientos de sistemas IoT en esta área junto con su equivalente comportamiento en IoA:

- *Sistema para el cuidado de personas mayores basado en IoT.* Un sistema para la prestación de servicios básicos a las personas mayores enfermas que viven solas y que deben tomar sus medicamentos según el horario determinado por el médico [8].
 - *Sistema equivalente basado en IoA.* Un sistema para proveer confort en el hogar de ancianos enfermos en función de su estado de salud. La inclusión de agentes inteligentes proporciona temperatura y confort lumínico en función de las condiciones meteorológicas. Además, el agente está capacitado para reportar fallas ocurridas en los componentes de la casa, pagar cuentas de servicios públicos en línea, comprar alimentos y medicinas en línea, controlar los horarios de los medicamentos y notificar a la familia o al centro médico si los signos vitales de la persona no son normales.
- *Sistema de monitorización de parámetros médicos basado en IoT.* Medición del nivel de glucosa del paciente a través de un glucómetro para controlar y analizar el estado de la enfermedad [152].
 - *Sistema equivalente basado en IoA.* Medición inteligente del nivel de glucosa del paciente, que informa de la condición al paciente, a su familia y a su médico para actuar preventivamente antes de que su estado se agrave. Además, en caso de emergencia, los agentes pueden analizar la geolocalización del paciente, recomendar el hospital más cercano, solicitar y pagar un taxi Uber para moverse lo más rápido posible (paciente consiente) o realizar una llamada a la ambulancia (paciente inconsciente).

3.5. Discusión

El IoT surgió a finales de los noventa. Sin embargo, los modelos para la integración de los agentes y las tecnologías de IoT han sido recientemente presentados. No obstante, el IoA sigue siendo un enfoque en proceso de consolidación que requiere la formulación de las directrices generales y una arquitectura de referencia estándar centrada en la gestión de componentes como la semántica, la adaptación orientada a usuarios finales, la inteligencia basada en diferentes técnicas (i.e., inteligencia artificial) y la interoperabilidad entre agentes heterogéneos. En este capítulo se ha definido el alcance del IoA y se ha formalizado las implicaciones de un ecosistema de IoA basado en tecnologías semánticas.

Aunque se han plantado algunos mecanismos y herramientas que permiten habilitar la interoperabilidad funcional en los sistemas multiagentes; la interoperabilidad semántica es aún un reto. Por ello, creemos que el IoA debe integrar un componente semántico que permita describir de manera global los ecosistemas de IoA mediante descriptores semánticos estándares. Así, los agentes que se ejecuten en sistemas multiagentes heterogéneos podrán actuar de forma más precisa en ecosistemas sociales y colaborativos a nivel global y no únicamente en plataformas específicas de agentes. Esto habilitaría el desarrollo de sistemas inteligentes de IoT basados en agentes interoperables entre sistemas complementarios como ciudades inteligentes, industria, cuidado de la salud, entre otros.

3.6. Contribuciones relacionadas

Las ideas y planteamientos presentados en este capítulo han sido evaluadas y publicadas en revistas científicas. Los detalles de estas contribuciones se detallan a continuación:

P. Pico-Valencia, J.A. Holgado-Terriza. Semantic Agent Contracts for Internet of Agents. In: *2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, Omaha, USA, 2016, pp. 76-79.

P. Pico-Valencia, J. A. Holgado-Terriza, D. Herrera-Sánchez, José Sampietro. *Towards the internet of agents: an analysis of the internet of things from the intelligence and autonomy perspective*, Ingeniería e Investigación, 38(1), 121-129 (2018) ISSN 2248-8723.

Capítulo 4

IoA-OWL, una ontología para describir el IoA

“La ciencia es el cementerio de las hipótesis”—**Lemait.**

El Internet de los Agentes (IoA) constituye una alternativa enfocada en mitigar las limitaciones inherentes al IoT concernientes a la autonomía, razonamiento y capacidades socio-colaborativas [186, 209]. Con el propósito de gestionar los principales componentes vinculados al enfoque del IoA de manera integrada, se considera oportuno el uso de un modelo semántico compacto para describir entidades que habiliten a los agentes para que lleven a cabo procesos más precisos basados en el significado de las cosas.

El modelo semántico propuesto, denominado IoA-OWL, sigue las directrices establecidas en una ontología que integra seis componentes entre los que figuran los siguientes: (i) contexto donde el agente opera (*AgentContext*), (ii) círculos sociales en los que el agente coopera (*AgentSocial*), (iii) recursos ofrecidos por el agente basados en ecosistemas heterogéneos de servicios (*AgentService*), (iv) recursos complementarios asociados a objetos de IoT (*AgentSmartThing*), (v) atributos descriptivos no funcionales e identificativos del agente (*AgentProfile*) y finalmente, (vi) descripciones de los artefactos que integra cada agente, acorde al modelo utilizado (*AgentModel*) [211]. En función de estos seis componentes, es posible describir entidades que operan en ecosistemas de IoA mediante el uso de contratos semánticos. Así, los agentes que operan en IoA emplean el mismo vocabulario y por tanto los ecosistemas donde trabajan podrán mejorar su nivel de interoperabilidad semántica. El modelo ontológico y los vocabularios reutilizados para su construcción son descritos con detalle en este capítulo.

4.1. Modelos semánticos para IoT

Las tecnologías relacionadas con las redes de sensores e IoT están tendiendo hacia la Web Semántica [242]. Varios modelos ontológicos [65] y propuestas enfocadas en la publicación de datos de sensores como datos enlazados [141] han sido planteados [45, 133]. Su objetivo se centra básicamente en la publicación de datos interconectados mediante la definición de relaciones entre ontologías especializadas y esquemas existentes de forma pública en la Web para que la descripción de los objetos de IoT se realice de forma estándar. De esta manera, los datos son comprensibles por humanos y las máquinas que los procesan, y consecuentemente, el proceso automático de descubrimiento de recursos (e.g. entidades, objetos, información) es más preciso debido a que dicho proceso se realiza en base al significado de las cosas. En este sentido, los agentes software constituyen entidades compatibles [121].

En vista de que los actuales objetos de IoT no poseen un suficiente grado de autonomía, los procesos de gestión de datos aplicando el enfoque de datos abiertos enlazados (*Linked Open Data*, LOD por sus siglas en inglés) se ve limitado. De hecho, algunos de los objetos de IoT comercializados por grandes empresas como Google o Microsoft les habilitan para que los datos sean gestionados solo por su propio servicio de Computación en la Nube. No obstante, los objetos de IoT deben brindar al usuario la posibilidad de definir los datos capturados como públicos o privados acorde a la situación y al tipo de aplicación desarrollada. En este sentido, la propuesta de vincular agentes software con los objetos o embeberlos en su arquitectura interna adquiere importancia y justifica esta propuesta ya que de esta manera el objeto no solo puede personalizar aspectos de gestión de los datos capturados (i.e., servicios de la nube a emplear, carácter de los datos); sino que también puede procesar y dar significado a dichos datos. Sin embargo, como se afirma en el estudio de Vandenbussche et al. [263], existen barreras para el despliegue de datos enlazados tales como la dificultad para determinar qué vocabularios específicos usar para describir la semántica de los datos.

4.1.1. Ontologías relacionadas con el IoA

La evolución reciente en el campo del IoT ha dado lugar a un renovado interés por las tecnologías orientadas a los agentes [7, 99, 135, 158,

187, 193, 276]. Sin embargo, en lo que respecta al dominio del IoA, pocos estudios se refieren directamente a este enfoque. Nos hemos basado en la concepción propuesta en Pico-Valencia et al. [209], donde el IoA es visto como una interacción inteligente de agente, definida en un nivel superior que gobierna los recursos del IoT, que incluye no sólo la estructura del agente y sus artefactos; sino también aspectos adicionales como su contexto de ejecución, naturaleza social, restricciones en tiempo real, ecosistema de servicios y un mecanismo para la planificación de tareas. En este contexto, las ontologías relativas a la tecnología de los agentes, los conceptos de IoT y los aspectos relacionados con los sistemas sensibles al contexto son fundamentales para describir con precisión los principales aspectos del IoA.

En la Tabla 4.1 se resumen las 15 ontologías—disponibles en Internet—más relevantes relacionadas con el IoA. La disponibilidad de estos modelos fue comprobada empleando herramientas especializadas de la Web Semántica como Linked Data Portal (<http://linkeddata.org/>) y Swoogle (<http://swoogle.umbc.edu/2006/>). Como muestra la columna 4 de la tabla especificada, cinco de estos modelos son ontologías publicadas como Vocabularios Abiertos Enlazados (1-5); mientras que el resto constituyen ontologías específicas (6-16) que cubren cuestiones individuales del IoT o de agentes.

En la columna 3 de la Tabla 4.1 se define el formato empleado para almacenar cada modelo semántico. Diez de estos modelos han sido expresados y publicados en formatos OWL (*Web Ontology Language*) y RDF (*Resource Description Framework*); mientras que los seis restantes fueron sólo propuestas extraídas de publicaciones científicas. Tanto el formato OWL como el RDF son lenguajes de etiquetado ampliamente utilizados para publicar y compartir datos en la web mediante ontologías. Por otra parte, la columna 5 etiquetada como *Disponible?* muestra que no todos los modelos listados estuvieron accesibles en Internet. De hecho, seis de los dieciséis modelos no fueron accesibles. Sin embargo, las ontologías conceptuales pudieron ser analizadas para incluir las principales clases, propiedades de objetos y propiedades de datos que previamente ya habían sido ideadas por otros grupos.

TABLA 4.1: Ontologías que describen aspectos relacionados con el IoA.

#	Modelos semánticos	Formato	Tipo	Disponibilidad	Reutilizado?	Documentación	C	O-P	D-P
1	FOAF [44]	RDF	Vocabulario	No	Sí	++	Sí	Sí	Sí
2	VCard [126]	RDF	Vocabulario	Sí	Sí	++	Sí	Sí	Sí
3	Computer [224]	OWL	Vocabulario	Sí	Sí	++	Sí	Sí	Sí
4	Dublin Core [79]	RDF	Vocabulario	Sí	Sí	++	Sí	Sí	Sí
5	M3-lite [116]	OWL	Vocabulario	Sí	Sí	++	Sí	Sí	Sí
6	Time [70]	OWL	Ontología	Sí	Sí	++	Sí	Sí	Sí
7	Cobra [56]	RDF	Ontología	Sí	Sí	++	Sí	Sí	Sí
8	CoDAMoS [63]	OWL	Ontología	Sí	No	+/-	Sí	No	No
9	COforAmI [216]	Teórico	Ontología	No	No	++	Sí	Sí	Sí
10	INoRIA SNoA [85]	RDF	Ontología	Sí	Sí	++	Sí	Sí	Sí
11	AgentOWL [150]	Teórico	Ontología	No	No	++	Sí	Sí	Sí
12	SOUPA [57]	Teórico	Ontología	No	No	--	Sí	No	No
13	BDI-workflow [165]	Teórico	Ontología	No	No	+/-	Sí	Sí	No
14	OWL-S [174]	OWL	Ontología	Sí	Sí	++	Sí	Sí	Sí
15	GOforIoT [117]	Teórico	Ontología	No	No	+/-	Sí	Sí	No

*C clases, O-P propiedad de objeto, D-P propiedad de datos reutilizada

Siguiendo con la explicación del contenido de la Tabla 4.1, la columna seis etiquetada como *Reutilizado?* muestra qué modelos fueron importados en la ontología propuesta (Sí), y qué modelos no fueron empleados de forma directa (No). La séptima columna etiquetada como *Documentación* muestra si cada modelo semántico tiene documentación completa (++), parcial (+/-) o limitada (- -). Finalmente, las últimas tres columnas muestran si nuestra propuesta reutilizó algunas de las clases (*C*), propiedades de objeto (*O-P*) y propiedades de datos (*D-P*) de cada modelo semántico listado.

4.1.2. Vocabularios enlazados y reutilizados

A continuación, se explica brevemente cada uno de los vocabularios reutilizados en la creación del modelo semántico para describir el IoA. Los vocabularios descritos corresponden a los elementos (1-5) listados en la Tabla 4.1.

- FOAF [44], *Friend Of A Friend*, es un vocabulario para vincular personas e información a través de la Web mediante el uso de redes sociales de colaboración humana, amistad y asociación. Algunas de las propiedades empleadas fueron *name* y *topic_interest*.
- VCard [126], es un vocabulario compuesto de términos útiles para describir personas y organizaciones. *Latitude*, *longitudo* y *country-name* fueron algunas de las propiedades que se reutilizaron.
- Computer [224], es un vocabulario que permite describir ordenadores personales, portátiles y netbooks. Las principales propiedades empleadas fueron: *battery type*, *processor model* y *memory*.
- Dublin Core [79], es un vocabulario especializado para describir recursos en la Web Semántica. *Creator*, *date* y *title* fueron algunas de las propiedades que se reutilizaron.
- M3-lite [116], es una taxonomía diseñada para permitir anotaciones semánticas para objetos de IoT heterogéneos. También incluye una clasificación bien definida de dispositivos basada en sensores y actuadores en diferentes dominios.

4.1.3. Ontologías reutilizadas

De manera similar a las descripciones de los vocabularios reutilizados que se explicaron previamente, a continuación, se describe cada una de las ontologías reutilizadas para construir el modelo semántico propuesto para describir el IoA con un mayor nivel de detalle. Dichas explicaciones corresponden a los elementos 6-15 incluidos en la Tabla 4.1.

- Time [70], es una ontología que proporciona un vocabulario para describir propiedades temporales de recursos. Propiedades como *time zone* y *unit type* fueron reutilizadas.
- Cobra [56], es una ontología de localización que detalla las propiedades espaciales y temporales de objetos físicos. Las principales propiedades reutilizadas fueron: *locatedIn*, *locatedFar*, *awayfrom*, *locatedNearBy*, *inRegion* y *notLocatedIn*.
- CoDAMoS [63], *Context-Driven Adaptation of Mobile Services*, es una ontología para resolver desafíos en el área de AmI, donde los dispositivos personales son considerados una extensión del entorno de cada usuario, ejecutando servicios móviles adaptados al usuario y su contexto. Algunas clases importantes integradas por este modelo fueron las siguientes: *profile*, *service* y *platform*.
- COforAmI [216], *Context Ontology for Ambient Intelligence*, es una extensión de la ontología CoDAMoS para la creación de aplicaciones informáticas sensibles al contexto. Algunas de las clases generales empleadas por este modelo fueron las siguientes: *service*, *platform* y *environment*.
- INRIA SNA [85], *Social Network Analysis*, es una ontología que incluye los conceptos y medidas de centralidad asociados al Análisis de Redes Sociales (SNA) [55], tales como *degree*, *betweenness*, *closeness*, *relationship* y *path*. Todos ellos fueron integrados en la ontología propuesta.
- AgentOWL [150], es una ontología que presenta un modelo que permite describir el entorno del agente JADE [28] (*collaborators*), su contexto (*resource*, *event*) y los resultados generados (*answer agents*).

- SOUPA [57], *Standard Ontology for Ubiquitous and Pervasive Applications*, es una ontología estándar para aplicaciones ubicuas y omnipresentes que describe agentes deliberativos basados en creencias, deseos e intenciones. Las principales clases reutilizadas fueron las siguientes: *belief*, *desire* and *intention*.
- BDI-workflow [165], es una ontología compatible con agentes BDI para describir el proceso de vinculación de servicios web semánticos según flujos de trabajo. Conceptos como: *goal*, *belief*, *plan* y *workflow* fueron reutilizados.
- OWL-S [174], *Web Ontology Language for Services*, es una ontología para la gestión de servicios semánticos mediante un perfil, un modelo de procesos y una base *grounding*. Las propiedades asociadas con el perfil y el modelo fueron reutilizadas por la ontología propuesta.
- GOforIoT [117], *General Ontology for IoT*, es una ontología global para IoT que emplea una abstracción de alto nivel para describir las funcionalidades técnicas de los dispositivos hardware. *Sensor*, *actuator* y *unit of measurement* fueron algunos de los conceptos reutilizados por la ontología propuesta.

4.2. Ontología IoA-OWL

El IoA no sólo exige la gestión de los aspectos no funcionales asociados a los agentes, sino también de los mecanismos de comunicación y de los artefactos de los agentes, como propone la mayoría de los estudios publicados en la Literatura. Además de estas cuestiones, también es necesario diseñar los mecanismos adecuados para proporcionar a las máquinas y a los usuarios la capacidad de descubrir y colaborar fácilmente con otros agentes asociados en redes heterogéneas de IoT.

Esta sección presenta la visión general de una ontología completa cuyas bases fueron propuestas por Pico-Valencia et al. [209]—compatible con los principales aspectos relacionados con la IoA detallados en el Capítulo 4— que está dirigida a describir los agentes que tienen la capacidad de descubrir los agentes contrapartes más adecuados para ejecutar las tareas en ecosistemas de IoT. Este proceso de descubrimiento es conducido

por un razonador semántico y un conjunto de descriptores o etiquetas semánticas descritos en una ontología a la que hemos denominado IoA-OWL. Esto significa, que el modelo ontológico que se describe a continuación, es la clave para integrar la búsqueda semántica de recursos en los ecosistemas de IoT.

4.2.1. Ontología de nivel superior para IoA

IoA-OWL [209] es un modelo ontológico para describir los ecosistemas de IoA. Este modelo integra seis componentes ontológicos que guardan suficiente información para ejecutar procesos de razonamiento semántico orientados a la recuperación de recursos en redes de IoT gestionadas por agentes. De este modo, los agentes pueden actuar de forma autónoma y social sobre diferentes ecosistemas dinámicos de IoT, así como sobre los MAS que gestionen redes de IoT. Formalmente, como se muestra en (4.1), este modelo está representado por seis tuplas, donde P_a es el *AgentProfile*, P_c el *AgentContext*, P_s el *AgentSocial*, P_m el *AgentModel*, P_v el *AgentService* y P_t el *AgentSmartThing*. Sus correspondientes implicaciones fueron detalladas en el Capítulo 4.

$$O_{IoA} < P_a, P_c, P_s, P_m, P_v, P_t > \quad (4.1)$$

La implementación de cada uno de los componentes especificados se ha logrado a través de una sub-ontología OWL. Utilizando estas piezas ontológicas desacopladas, es factible ampliar nuestro modelo semántico en función de necesidades futuras como la inclusión de un componente para describir la movilidad del agente, el modelo normativo del agente, entre otros. A continuación, la Figura 4.1 muestra el modelo detallado a nivel de implementación. El diagrama ilustrado corresponde al que se muestra al gráfico ontológico (*ontograph*) generado a partir del editor de ontologías Protegé (<https://protege.stanford.edu/>).

Los seis componentes de IoA-OWL se incluyeron mediante el uso de seis propiedades de objetos (*object-properties*). La propiedad *presents* y su correspondiente propiedad inversa *isPresentedBy* integran el *AgentProfile*. La propiedad *acts* y su propiedad inversa *isActingIn* permiten la integración del componente *AgentContext*. Del mismo modo, las propiedades *leans* y *isLeanedBy* nos permiten describir la información relativa al componente *AgentSocial*.

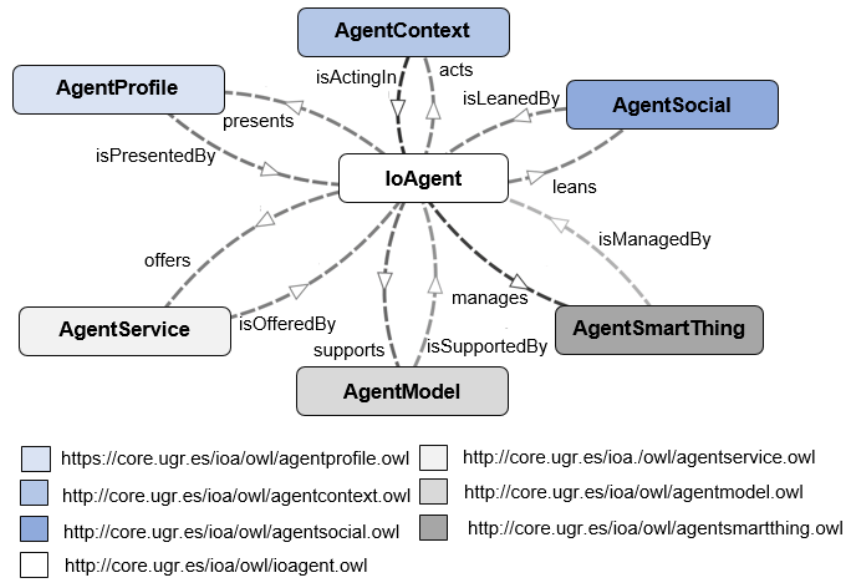


FIGURA 4.1: Ontología superior y componentes ontológicos de IoA-OWL

El *AgentModel* se enlazó a través de la propiedad *supports* y su propiedad inversa *isSupportedBy*. A continuación, *AgentService* se integró en el modelo ontológico a través de las propiedades *offers* y *isOfferedBy*. Finalmente, el componente *AgentSmartThing* se integró mediante la propiedad *manages* y su propiedad inversa *isManagedBy*. Estos componentes, acoplados de forma conjunta, constituyen la ontología IoA-OWL a partir de la cual se describen semánticamente los objetos de IoT y su agente vinculado para que puedan ser descubiertos fácilmente por contrapartes externos.

Hemos hecho grandes esfuerzos para identificar las ontologías más apropiadas en relación con la descripción del IoT y los agentes de software. No es intención de este trabajo empezar a crear una ontología completa desde el principio; sino más bien identificar las clases, propiedades de datos y propiedades de objetos más relevantes para componer un nuevo modelo capaz de soportar los aspectos más comunes manejados por el dominio IoA. Una vez aclarado esto, se procedió a formalizar una especificación de cada componente ontológico mediante el uso de tuplas.

4.2.2. Componente *AgentProfile*

El perfil del agente describe las propiedades no funcionales del agente. Formalmente, y de forma obligatoria, el *AgentProfile* está representado por ocho tuplas, como detalla la ecuación (4.2).

$$Pa_i < pid_i, pn_i, pd_i, [optionalElements], pB_i, pO_i, pC_i, pP_i, pRT_i > \quad (4.2)$$

Los tres primeros elementos corresponden a propiedades de datos que describen los aspectos básicos del agente: *identifier* (pid_i), *name* (pn_i) y *description* (pd_i). Sin embargo, es posible incluir propiedades de datos opcionales como *status* (ps_i), *version* (pv_i), *date of submission* (pds_i), *date of publishing* (pdp_i), *requests* (pr_i) y *accomplished requests* (par_i) si es necesario. Por otra parte, los cuatro elementos restantes constituyen propiedades sobre objetos que vinculan al agente con otras entidades. Estas propiedades, mostradas en la Figura 4.2, son descritas a continuación:

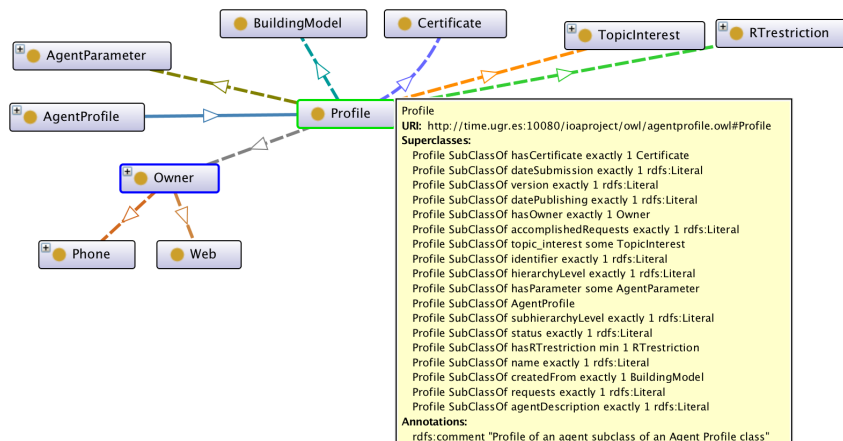


FIGURA 4.2: Gráfico ontológico del componente *AgentProfile*

- La propiedad *createdFrom* determina el modelo *Building-Model* (pB_i) usado para crear el agente. Un modelo de construcción es un conjunto de reglas que muestran detalles sobre la creación de agentes. Dicho modelo es útil para validar cada agente antes de su publicación.

- La propiedad *hasOwner* permite describir al propietario *Owner* (pO_i) del agente. Un propietario es el creador o proveedor del agente, y él/ella es descrito por datos como: *identifier*, *name*, *website* y *phone*.
- La propiedad *hasCertificate* vincula la propiedad *Digital Certificate* (pC_i) con el agente. Un certificado digital guarda información sobre las credenciales del agente, lo que garantiza la seguridad de los procesos de comunicación entre agentes y la confianza en terceros.
- La propiedad *hasParameter* permite especificar los parámetros temporales *Parameters* (pP_i) que el agente utiliza para ejecutar la tarea de gestionar los recursos de IoT y la tarea de descubrir contrapartes en el ecosistema de IoA.
- La propiedad *hasRestriction* enlaza con la propiedad *Real-Time Constraints* (pRT_I) que el agente satisface para el logro de sus objetivos. Cada restricción en tiempo real se describe mediante un parámetro en tiempo real, un operador y un valor numérico expresado en milisegundos.

Esta sub-ontología se definió mediante la reutilización de términos FOAF. Sin embargo, pB_i , pO_i y pC_i fueron introducidos usando un nuevo vocabulario. La información guardada en este componente permite a los agentes externos identificar los agentes más relevantes para su uso en un dominio específico basado en los metadatos de los aspectos de los agentes no funcionales. Por ejemplo, para determinar si hay agentes capaces de medir una condición específica en un tiempo determinado.

4.2.3. Componente *AgentContext*

El contexto del agente *AgentContext* caracteriza la situación actual respecto al entorno físico real en el que se ejecuta el agente [279]. Sobre este trasfondo, el contexto está formalmente representado por siete tuplas mandatarias como se muestra en la ecuación (4.3).

$$Pc_i < cid_i, cn_i, cd_i, cL_i, cT_i, cN_i, cP_i, [optionalElements] > \quad (4.3)$$

Los tres primeros elementos constituyen propiedades de datos y los restantes son propiedades de objetos. Las propiedades de datos como *identificar* (cid_i), *nombre* (cn_i) y *descripción* (cd_i) describen la información sobre el contexto de agente específico. Por otro lado, los cinco elementos restantes permiten vincular cuatro entidades en relación con los sistemas sensibles al contexto y una entidad adicional relacionada con el propio contexto del agente. Todos ellos, ilustrados en la Figura 4.3, fueron introducidos utilizando la propiedad *isDescribedBy*:

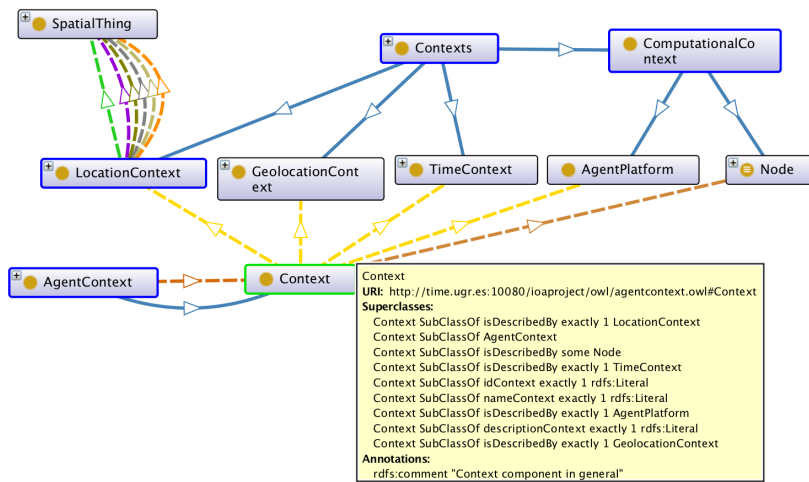


FIGURA 4.3: Gráfico ontológico del componente *Agent-Context*

- El contexto de localización (cL_i) se definió reutilizando la ontología de Cobra 4.0 para incluir propiedades espaciales como: *locatedIn*, *locatedFarAwayFrom*, *nextTo*, *insideOf*, *outsideOf*, *locatedNearBy* y *behind*.
- Para definir el contexto relacionado con el tiempo (cT_i) se reutilizaron las propiedades de datos de contexto soportadas por las ontologías *Time* y *VCard* como son: *time zone*, *unit time* y *local time*.
- El contexto del nodo computacional (cN_i) es el hábitat donde residen los agentes. Este contexto está formado por un ordenador

personal o dispositivo, que se describe semánticamente utilizando la ontología *Computer* y *m3-lite*.

- La plataforma de agente (cP_i) se describe utilizando propiedades de datos como: *identifier*, *name*, *description*, *host* y *port* de la plataforma de agente.
- Finalmente, el contexto de geolocalización (cG_i) fue definido usando términos de la ontología *VCard*. Así, la descripción de datos geográficos como: *country*, *locality*, *latitude* y *longitude* fue posible. Es importante destacar que este elemento es opcional ya que no todos los dispositivos soportan la geolocalización por sí mismos.

La información guardada por esta sub-ontología proporciona a los agentes y objetos de IoT de datos contextuales que el razonador del agente puede emplear para actuar coherentemente en entornos dinámicos donde el contexto del mundo real es crucial para comprender la situación en la que se proporciona una funcionalidad [13]. Por lo tanto, los agentes de IoA tienen conocimientos sobre el entorno actual y los cambios que se producen para gestionar los recursos de IoT de manera coherente. Por ejemplo, una tarea en la que esta información es relevante podría ser la adquisición de la predicción meteorológica de la zona en la que están instalados los sensores de una red de IoT.

4.2.4. Componente *AgentService*

Este componente fue diseñado teniendo en cuenta la integración de servicios heterogéneos como acciones del agente [202, 212]. Por lo tanto, las acciones ejecutadas por el agente encapsulan un conjunto de servicios web—SOAP, RESTful, DPWS, DOHA—organizados lógicamente para lograr sus objetivos. Cada servicio se encuentra formalmente representado por un tupla de cinco elementos como muestra la ecuación 4.4.

$$Pv_i < su_i, sPr_i, sIP_i, sOP_i, sO_i > \quad (4.4)$$

La única propiedad de datos de este componente es la URI del contrato (su_i) la cual define la ruta del documento que describe el servicio web que ofrece la funcionalidad requerida por el agente modelado. Los elementos restantes son propiedades de objeto. Dichas propiedades se ilustran en la Figura 4.4 y se describen a continuación:

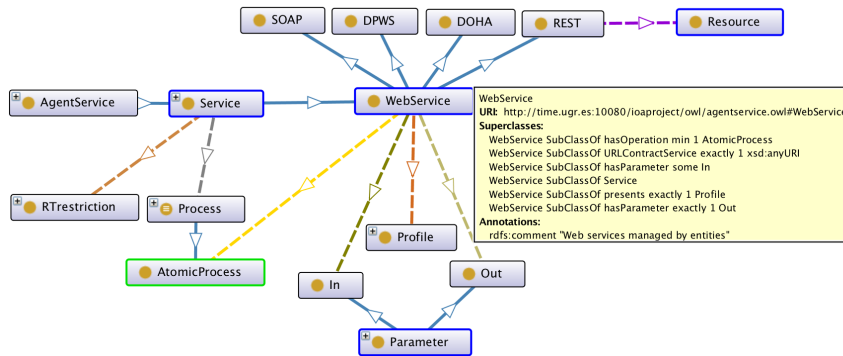


FIGURA 4.4: Gráfico ontológico del componente *AgentService*

- La información no funcional sobre el servicio incluida como perfil de servicio de la ontología OWL-S [174] (sPr_i) fue enlazada mediante la propiedad *present*.
- Se introdujo un conjunto de parámetros de entrada (sIP_i) requeridos por el servicio web para realizar sus operaciones utilizando la propiedad *hasParameter*.
- Se incluyó un parámetro de salida (sOP_i) para representar el resultado del servicio web empleando también la propiedad *hasParameter*.
- Finalmente, se introdujo una operación (sO_i) mediante la propiedad *hasOperation*. En función de esta operación, el servicio devuelve un valor específico.

Hemos ampliado el modelo OWL-S [174] porque está estandarizado y es compatible con servicios web SOAP, DPWS y DOHA. La extensión del modelo ontológico consistió básicamente en la adición de descriptores para soportar los recursos gestionados por los servicios RESTful. De este modo, proporcionamos al enfoque IoA la capacidad de aplicar el enfoque IoS para componer procesos complejos a partir de una serie de microservicios heterogéneos implementados como servicios web. Esto se implementó a través de una API que maneja uniformemente los cuatro tipos de servicios utilizados en esta propuesta [212]. Así, por ejemplo,

un agente puede combinar un servicio REST para obtener el pronóstico del tiempo con un servicio DPWS para controlar el sistema de aire acondicionado basado en los datos pronosticados.

4.2.5. Componente *AgentModel*

Este componente se introduce para describir la arquitectura de los agentes reactivos y deliberativos. El modelo de agente *AgentModel* permite al creador del agente describir los artefactos estructurales de software que este tipo de entidades emplean para definir su estructura. En este trabajo sólo hemos considerado los agentes reactivos debido a sus beneficios en ecosistemas de IoA (justificado en el Capítulo 1).

Un agente de JADE es una entidad reactiva autónoma cuya funcionalidad se modela utilizando comportamientos. Un comportamiento es una unidad computacional que se activa automáticamente cuando ocurre un evento para resolver una tarea específica [29]. Formalmente, un agente reactivo puede ser representado por una dupla como se define en la ecuación 4.5:

$$Pmr_i < rP_i, rBehav_i > \quad (4.5)$$

Las dos propiedades que describen este componente ontológico son ilustrados en la Figura 4.5. La propiedad de objeto *hasParameter* nos permite describir los argumentos (rP_i) requeridos por el agente para ejecutar sus tareas. Además, los comportamientos ($rBehav_i$) que coordinan la conducta del agente se describen a través de la propiedad de objeto *hasBehavior*. Este tipo de comportamientos están orientados a controlar el objeto de IoT vinculado al agente.

Los comportamientos de agentes ($rBehav_i$) son unidades de software que forman parte de la estructura de agentes JADE. Se introducen en tiempo de diseño para dar al agente funciones específicas. Este tipo de comportamiento está representado por una tupla formada de cuatro elementos $rBehav_i < n_i, d_i, P_i, W_i >$. Las dos primeras propiedades de datos describen información básica, como el comportamiento *name* (n_i) y *description* del agente. Los dos elementos restantes son propiedades de objeto que nos permiten vincular los argumentos (P_i) y el flujo de trabajo (W_i) definido por el comportamiento.

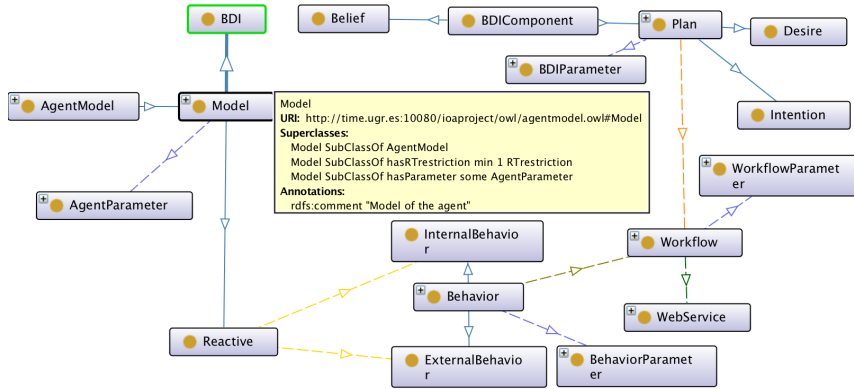


FIGURA 4.5: Gráfico ontológico del componente *Agent-Model*

Un flujo de trabajo puede incluir acciones de control específicas, invocaciones de operaciones ejecutadas sobre objetos de IoT a través de servicios web e incluso peticiones realizadas a agentes asociados. Formalmente, un flujo de trabajo está compuesto por dos elementos $W_i < wn_i, wsect_i >$, donde wn_i es el nombre lógico del flujo de trabajo y $wsect_i$ es la URI del documento donde se almacena.

4.2.6. Componente *AgentSmartThing*

Este componente nos proporciona información esencial sobre los objetos de IoT y sus recursos. Esto permite que los objetos de IoT se conviertan en objetos activos; es decir, que los objetos de IoT estén controlados por agentes y, por consiguiente, puedan actuar de forma proactiva e inteligente. Dichos agentes pueden utilizar la información asociada a estos objetos para establecer el control sobre ellos. Formalmente, este componente está representado por una tupla de siete elementos, como se detalla la ecuación 4.6.

$$Pt_i < tlog_i, tnet_i, tF_i, tG_i, tL_i, tT_i, tR_i > \quad (4.6)$$

Las dos primeras propiedades son de tipo propiedad de datos: *logical name* ($tlog_i$) y *host* ($tnet_i$) del objeto de IoT. El tercer elemento es una propiedad de objeto denominada funcionalidades u operaciones (tF_i). Dichas funcionalidades especifican los datos y unidades de medida

generados por los objetos de IoT que son obtenidos por un agente a través de la invocación de servicios web. Las propiedades restantes también constituyen propiedades de objeto que integran información relativa a la geolocalización (tG_i), localización (tL_i) y tiempo (tT_i) en el que opera el objeto de IoT. Estas propiedades, ilustradas en la Figura 4.6, se gestionan de forma similar al componente de contexto del agente descrito en el componente *AgentContext*.

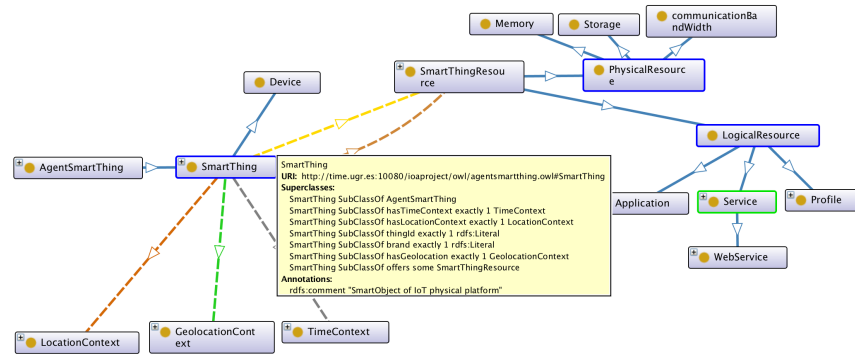


FIGURA 4.6: Gráfico ontológico del componente *AgentSmartThing*

La última propiedad, denominada *offered*, nos permite describir los recursos de IoT (tR_i) del tipo recursos computacionales. A manera de ejemplificación, estos metadatos son útiles para monitorizar la ubicación de un sensor móvil y su nivel de batería para que el administrador del sistema pueda actuar de manera oportuna y así evitar que el objeto interrumpa su funcionamiento.

4.2.7. Componente *AgentSocial*

El componente *AgentSocial* es un componente integrado para permitir la colaboración en el dominio del IoA a través de entidades y círculos sociales establecidos entre los agentes implicados. Con este componente se puede modelar el círculo social de un agente guiado por los temas de interés del agente y las unidades de medida requeridas para ejecutar acciones específicas. El proceso de búsqueda se apoya en la teoría del análisis de redes sociales.

Formalmente, este componente está representado por una tupla de seis elementos como se establece en la ecuación 4.7, donde la primera propiedad es una propiedad de datos llamada grado de colaboración (soc_i). Esta propiedad especifica si el agente es una unidad colaborativa o no. Únicamente los agentes colaborativos construyen un círculo social. Los elementos restantes incluidos en este componente corresponden a propiedades de objeto. Dichas propiedades son ilustradas en la Figura 4.7 y se describen a continuación:

$$Ps_i < soc_i, soUMr_i, soUMp_i, soT_i, soSN_i, soR_i > \quad (4.7)$$

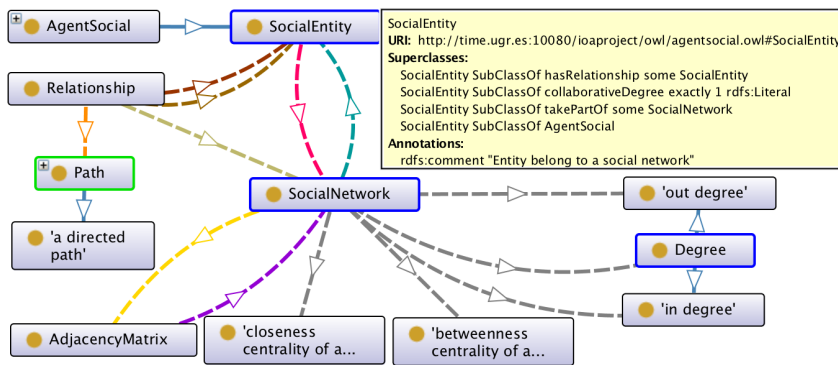


FIGURA 4.7: Gráfico ontológico del componente *AgentSocial*

- La propiedad tema de interés especifica los temas de interés particulares (soT_i) que tiene el agente. Cada tema de interés se describe siguiendo el esquema compuesto de campo, disciplina, sub-disciplina y tema.
- Unidades de medida en las que el agente está interesado en recibir ($soUMr_i$) para lograr su objetivo y las que puede proporcionar ($soUMp_i$) entre sus colaboradores.
- $soSN_i$ detalla en qué redes sociales coopera el agente. Una red social $soSN_i$ está representada por una tupla de ocho elementos $soSN_i < nn_i, nM_i, nSSE_i, nI_i, nO_i, nD_i, nB_i, nC_i >$, donde el nombre de la red social (nn_i) es un atributo expresado mediante

una propiedad de datos; mientras que el resto de elementos son introducidos mediante propiedades de objetos. Cada red social está representada con una matriz de adyacencia (nM_i) de orden N, donde N es el total de contrapartes del agente en la red social. Además, se requiere un conjunto de un mínimo de tres agentes ($nSSE_i$) para crear una red social. Finalmente, las cinco medidas de centralidad más utilizadas fueron: *InDegree* (nI_i), *OutDegree* (nO_i), *Degree* (nD_i), *Betweenness* (nB_i), and *Closeness* (nC_i) [55, 137]. Estas medidas de centralidad se emplearon para identificar los nodos prominentes en un gráfico social [274].

- soR_i integra relaciones que un agente guarda con agentes externos conocidos en una red social. Una relación es representada formalmente usando una tríada $rR_i < rn_i, rTo_i, rFrom_i >$, donde el nombre de la relación (rn_i) es un dato introducido a través de una propiedad de datos. Contrariamente, el origen ($rFrom_i$) y destino (rTo_i) de la relación fueron introducidos mediante propiedades de objeto.

Algunas de estas propiedades fueron integradas por esta sub-ontología y fueron reutilizadas a partir de la ontología SNA[85]. Con esta información, los agentes colaborativos pueden descubrir contrapartes externos. A partir de los agentes contrapartes recuperados, el círculo social y las tareas de colaboración son realizadas por el agente. Un ejemplo del uso de esta información puede ser un agente colaborativo que busca contrapartes con mayor poder social y que estén interesados en ejecutar acciones de control de confort inteligente en hogares. Así, el agente evita la búsqueda en grandes catálogos de agentes y se dirige sólo a sus contrapartes. Sin embargo, el agente está obligado a mantener su círculo social actualizado a intervalos regulares. Este aspecto dependerá de cuán dinámico es el ecosistema de IoA donde opera el agente.

4.3. Discusión

Ecosistemas actuales de agentes fusionados con IoT son cada vez más comunes. Se ha presentado un modelo semántico que describe seis componentes—*AgentProfile*, *AgentContext*, *AgentService*, *AgentModel*, *AgentSmartThing* y *AgentSocial*—que ayudan a describir los recursos

asociados a los dispositivos de IoT para que los agentes pueden gestionarlos de manera más precisa. En función a nuevas necesidades, el modelo ontológico propuesto puede ser extendido para que incorpore componentes adicionales y así describir nuevas entidades y elementos que formen parte de ecosistemas de IoA. No obstante se recomienda que los nuevos componentes consideren la reutilización de conceptos ya descritos por otros modelos semánticos y que de preferencias hayan sido evaluados por la comunidad científica y organismos responsables de estandarización.

El uso de descripciones formales y estándares para la descripción de servicios web ha permitido que dichos componentes lleguen a convertirse en piezas de software semánticamente interoperables. Sobre estas bases, esta propuesta busca también mejorar la interoperabilidad semántica de los agentes de software en ecosistemas de IoA. Se ha planteado un modelo semántico a partir del cual los agentes pueden modelar un contrato descriptivo de agente (similar al contrato manejado por los servicios) a fin de que los agentes de ecosistemas de IoA puedan ser descritos utilizando un mismo vocabulario y con un mayor nivel de detalle (incluyendo la mayor cantidad de descriptores) y que así, puedan llevar a cabo procesos de razonamiento semántico a partir de los cuales se puedan recuperar los datos y contrapartes más adecuados para alcanzar objetivos colaborativos

4.4. Contribuciones relacionadas

El modelo planteado en este capítulo ha sido evaluado por pares especialistas en el área. Un artículo de revista científica fue obtenido como resultado. Los datos de la contribución es especificada a continuación:

P. Pico-Valencia, J.A. Holgado-Terriza, J. Senso. Towards an Internet of Agents Model Based on Linked Open Data Approach. *Journal of Autonomous Agents and Multi-Agent Systems* [En Revisión].

Capítulo 5

Framework-IoA: Un marco de trabajo para agentificar el IoT

“Cada día sabemos más y entendemos menos.”— **Albert Einstein**.

El Internet de los Agents (IoA) se basa en el principio del concepto *Agent of Thing* [185] para integrar un componente de inteligencia y colaboración que habilita a los objetos de Internet de las Cosas (IoT) interactuar de forma activa con otros objetos conectados a redes heterogéneas [99]. El concepto *Agent of Thing* es una de las primeras aproximaciones hacia el enfoque IoA en términos de implementación. Sin embargo, debido a que la web actual tiene una tendencia creciente hacia la Web Semántica, este trabajo propone para el IoA el uso de agentes capaces de manejar datos enlazados [33] y un componente semántico.

El uso de datos enlazados proporciona un enfoque de publicación flexible que facilita a los consumidores de datos descubrir e integrar las fuentes de datos distribuidas en la web [33, 120]. Esto es posible gracias a características del enfoque tales como: el uso de un modelo de datos uniforme, el uso de un mecanismo estandarizado de acceso a los datos, el uso de un descubrimiento de datos distribuidos basado en enlaces, y el uso de datos auto-descriptivos para gestionar fuentes de datos heterogéneas. En este contexto, el IoA puede beneficiarse del uso de los datos enlazados, contribuyendo a que los agentes tengan la capacidad de gestionar los datos de forma transparente y proporcionar datos uniformes y globales a partir de los cuales se pueden optimizar el trabajo colaborativo, la toma de decisiones y la interoperabilidad semántica en ecosistemas de IoT. En este capítulo se presentan las directrices generales del marco de trabajo Framework-IoA el cual direcciona el proceso de agentificación del IoT usando sistemas multiagentes.

5.1. IoA basado en datos enlazados

Uno de los principales objetivos de esta investigación es modelar un ecosistema de IoA introduciendo tecnologías semánticas y datos enlazados. Para llevarlo a cabo se ha planteado un modelo orientado a agentificar el IoT mediante el uso de sistemas multiagentes. Como ilustra la Figura 5.1, dicho modelo cuenta en términos generales de un conjunto de Agentes Abiertos Enlazados (Linked Open Agents) o LOAs—como unidad de procesamiento—, sus LACs correspondientes—como unidad de descripción semántica—y sus respectivos WACs—como unidad de control de las acciones ejecutadas por los agentes. Tanto la unidad descriptiva (LAC) como la unidad funcional (WAC) vinculadas a cada LOA son indispensables para proporcionar capacidades de colaboración con otros LOA homólogos recuperados mediante un proceso de descubrimiento de agentes contrapartes, así como para controlar directamente los objetos de IoT que tiene asociados.

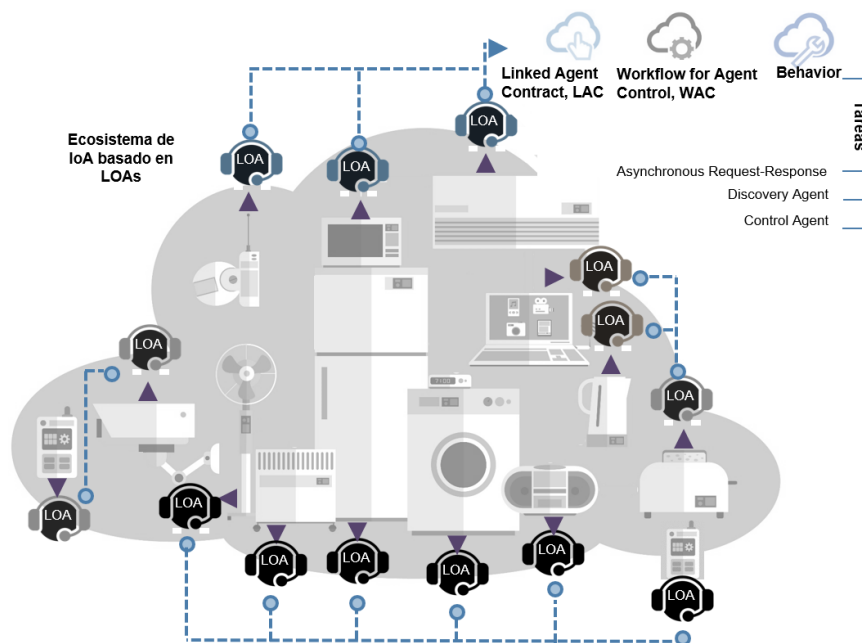


FIGURA 5.1: Ecosistema de IoA dirigido por entidades LOAs.

La Figura 5.2 detalla brevemente los principales niveles de abstracción y herramientas introducidas en esta propuesta para modelar el IoA basado en el enfoque Datos Abiertos Enlazados (*Linked Open Data*, por sus siglas en inglés LOD). Por un lado, un WAC describe una secuencia de acciones que el agente debe ejecutar para lograr sus objetivos. Por otro lado, un LAC incluye descripciones semánticas de las propiedades del agente que soportan un modelo contractual basado en un meta-modelo conocido como IoA-Contract. Este meta-modelo ha sido formalizado a través de un diagrama UML compuesto por las clases que representan los conceptos asociados a los principales aspectos que debe tratar el IoA (Anexo E), y que han sido definidos siguiendo las directrices de la ontología IoA-OWL [209]. Esta ontología descrita en el Capítulo 4 fue creada siguiendo las buenas prácticas de la Web Semántica, que propone reutilizar conceptos para describir entidades en la web siempre que sea posible.

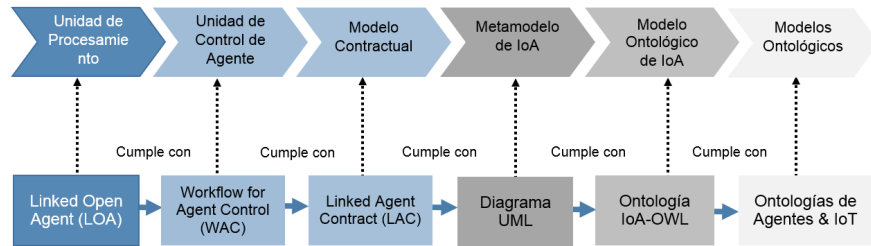


FIGURA 5.2: Proceso, y principales mecanismos y herramientas para modelar el IoA basado en datos enlazados.

5.2. *Linked Open Agent*

Un agente abierto enlazado (*Linked Open Agent*, LOA por sus siglas en inglés) una entidad de software autónoma descrita empleando el enfoque LOD. Formalmente un LOA puede ser representado mediante una tupla formada de 4 elementos (ecuación 5.1), donde id_i constituye su identificador que le sirve para distinguirse en el ecosistema de IoA, LAC_i es su contrato semántico, WAC_i es su flujo de control, y $Tasks_i$ corresponde al conjunto finito de tareas a partir de las cuales puede controlar uno o más objetos de IoT de forma independiente y colaborativa. Dichas

tareas son descritas con mayor detalle en el aparatado que refiere a la arquitectura LOA.

$$LOA_i < id_i, LAC_i, WAC_i, TASKS_i > \quad (5.1)$$

De manera genérica, un LOA maneja un conjunto de descriptores semánticos asociados con los principales aspectos modelados por el IoA (descritos en el Capítulo 4), tales como *AgentProfile*, *AgentContext*, *AgentService*, *AgentModel*, *AgentSmartThing* y *AgentSocial*. Esta información, almacenada de forma integrada, habilita a los agentes para que puedan ejecutar procesos de descubrimiento basados en descriptores comunes relacionados con las aplicaciones de IoT. En consecuencia, la probabilidad de recuperación de contrapartes especializados aumenta en comparación con las plataformas en las que se utilizan ciertos descriptores de forma desacoplada. Así, la gestión adecuada de estos descriptores contribuye significativamente para que los agentes que se ejecutan en los ecosistemas de IoA adquieran capacidades mejoradas y adicionales que los modelos de agentes actuales no ofrecen. Entre las principales características adquiridas por los LOAs se describen las siguientes:

- *Autónomo*. Esta característica permite a los LOAs actuar en base al conocimiento modelado almacenado mediante el uso de contratos para que los agentes puedan ser capaces de desempeñarse con la mínima intervención humana.
- *Sociable y colaborativo*. Esta propiedad le da a los LOAs la capacidad de comunicarse con otros LOAs, que residen en el ecosistema de IoA, para solicitar o proporcionar colaboración de manera proactiva. Aunque los LOAs se basan en el modelo de agente reactivo para definir comportamientos que les permiten actuar sobre los mensajes que reciben y ejecutar su flujo de trabajo de acuerdo a un planificador, estos agentes también incorporan ciertos comportamientos pro-activos. Estos comportamientos pro-activos son llevados a cabo por los propios LOAs para aprovechar los nuevos agentes contrapartes publicados en el ecosistema de IoT. Algunas de las acciones pro-activas que los LOAs pueden ejecutar son las siguientes acciones:
 - Recuperación ante fallos reportados por los objetos de IoT y los agentes contrapartes.

- Optimización del tiempo de procesamiento de su flujo de trabajo (WAC) buscando y usando los agentes con mejor tiempo de respuesta.
 - Actualización de su círculo social cada vez que se integren mejores agentes contrapartes especializados.
-
- *Semántico*. Esta característica permite describir semánticamente los LOAs. Así, los agentes tienen la capacidad de descubrir a otros homólogos basándose en la información semántica almacenada en sus contratos. Esto implica que los procesos de comparación se realizan usando el significado de las cosas en lugar de solo establecer comparaciones sintácticas.
 - *Abierto*. La reutilización de los LOAs existentes requiere que los agentes publiquen sus descripciones como datos abiertos mediante la adopción de licencias *Creative Commons* (CC) [145]. De esta manera, los desarrolladores de aplicaciones basadas en IoA pueden reutilizar descriptores integrados en los LACs de LOAs que fueron previamente publicados en repositorios manejados en IoA.
 - *Sensible al contexto*. La información contextual combinada con descriptores pertenecientes a otros componentes semánticos permite a los agentes la ejecución de acciones de control más exactas y consistentes respecto su contexto de operación.
 - *Inteligente*. La consistencia de las acciones implementadas por los LOAs depende en parte de cómo se utilizan los datos del contrato. En este sentido, los agentes adquieren un nivel avanzado de inteligencia porque realizan procesos basados en razonamientos semánticos sobre el ecosistema IoA.
 - *SopORTE de servicios*. El uso de servicios web para modelar acciones de agentes permite a los LOAs ejecutar tareas complejas sin convertirse en entidades de software demasiado pesadas que posean todo el código de sus algoritmos y procesos embebidos en su estructura.
 - *Adaptable*. Los entornos dinámicos implican un cambio constante. Los LOAs adaptan su comportamiento de acuerdo a la información

contenida en su LAC y flujo de trabajo. Ambos elementos se actualizan con la información del entorno en el que se está ejecutando el agente.

- *Basado en estándares.* Los LOAs son compatibles con el lenguaje estándar ACL. Además, con el fin de mantener la interoperabilidad funcional, los contratos son compatibles con el estándar de intercambio de datos en la web, JSON-LD (*JavaScript Object Notation for Linked Data*) [251] y las tripletas RDF.
- *Testable.* Por razones de seguridad se requiere que los LOAs puedan ser auditados. Aspectos como las credenciales y las redes de colaboración en las que participa cada agente se pueden verificar gracias a la información almacenada en su contrato.

5.2.1. Arquitectura de agente

En IoA es posible integrar LOAs colaborativos y no colaborativos. La decisión del tipo de agente que debe modelarse depende en términos generales de las restricciones de control del objeto de IoT y del nivel de privacidad de los datos tratados por el LOA. A continuación se describen las principales características que definen a los LOAs colaborativos y no colaborativos:

- Un LOA no colaborativo es un agente capaz de controlar su objeto de IoT vinculado de forma autónoma; es decir, valiéndose de sus propios recursos y sin necesidad de interacción con terceros. Este tipo de LOAs se justifican cuando la información que manejan es propiedad privada y no puede ser compartida. Además, un LOA no colaborativo es útil cuando se requieren entidades activas capaces de controlar múltiples objetos de IoT que operan de forma pasiva y con un cierto nivel de inteligencia.
- Un LOA colaborativo se define como un agente con la capacidad de controlar su objeto IoT vinculado aprovechando los datos especializados proporcionados por contrapartes externos, los cuales son descubiertos a través de la ejecución de procesos de razonamiento que utilizan los metadatos descritos por los LOAs que forman parte del ecosistema global de IoA. Este tipo de agentes son entidades

flexibles y son capaces de adaptar su comportamiento en función de los recursos que pueden descubrir en su entorno. Es importante notar que cuando se crea un LOA colaborativo, su información se hace pública para el ecosistema de IoA. Sin embargo, es posible restringir qué datos específicos no deben ser compartidos públicamente para evitar problemas de privacidad (i.e., acceso indiscriminado a datos confidenciales).

Los componentes de un LOA, ilustrados en la Figura 5.3, muestran sus cinco componentes básicos. En primer lugar, el Linked Agent Contract (LAC) es la unidad básica de descripción semántica del LOA para describir las propiedades y capacidades de este agente así como las características de los agentes con los que colabora. Este contrato sigue las especificaciones de la ontología IoA-OWL y puede ser actualizado regularmente para reflejar el estado del LOA. En segundo lugar, el flujo de control es la unidad de control del LOA que define el comportamiento del agente sobre el ecosistema de IoA. Contiene una descripción de la secuencia de acciones que el agente debe ejecutar para lograr sus objetivos. En tercer lugar, las tareas del agente son responsables de mantener actualizado el estado del LOA y ayudar a la ejecución regular del flujo de trabajo del agente.

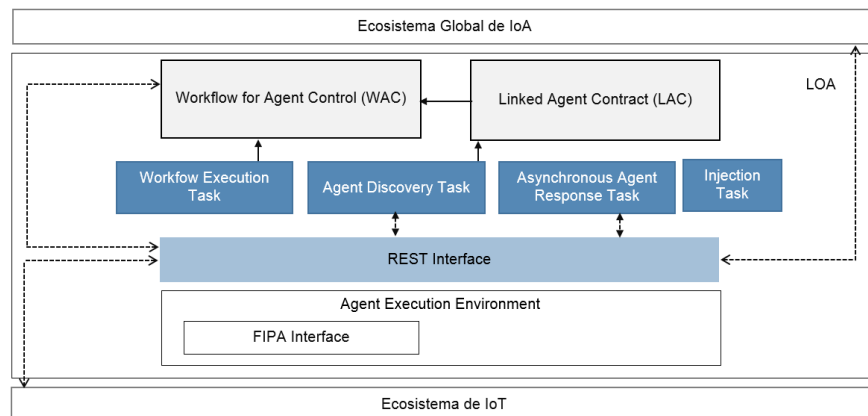


FIGURA 5.3: Arquitectura de un LOA.

A continuación, se especifican y describen cada una de las tareas genéricas que modela un LOA.

- Tarea de Descubrimiento de Agentes (*Agent Discovery Task*). Esta tarea es la encargada de ejecutar el proceso de descubrimiento de LOAs contrapartes en el ecosistema de IoA. Como resultado, un LOA puede conocer a otros candidatos con los cuales el agente puede colaborar.
- Tarea de ejecución de flujo de trabajo (*Workflow Execution Task*). Esta tarea guía la ejecución del flujo de trabajo siguiendo un modelo de programación basado en un bucle compartido sincronizado. La naturaleza reactiva del agente depende de la frecuencia de la ejecución de esta tarea y de la secuencia de acciones que contiene el flujo de control.
- Tarea de respuesta del agente asíncrono (*Asynchronous Agent Response Task*). Intercepta y procesa cualquier mensaje escrito en FIPA-ACL cuyo destinatario sea el propio LOA. Esta tarea espera las posibles solicitudes de los mensajes FIPA-ACL, ejecuta la solicitud correspondiente de forma asíncrona sin bloquear al remitente y, a continuación, envía una respuesta al remitente del agente.
- Tarea de Inyección (*Injection Task*). Inyecta una nueva tarea extra al agente en caso de que las tareas genéricas definidas por un LOA no sean suficientes para ejecutar las acciones en el ecosistema de IoA modelado. Este proceso se lleva a cabo adoptando la filosofía propuesta por la herramienta ADELE (*Agent Dynamic Evolutionary at runtime*), el cual se basa en técnicas de Metaprogramación [208].

En cuarto lugar, la interfaz *REST* permite que el LOA se comunique con el mundo exterior (i.e., LOAs, agentes que soportan el estándar de comunicación de agentes ACL y objetos de IoT) utilizando servicios REST y de una manera más general empleando URIs.

Por último, el entorno de ejecución del agente (*Agent Execution Environment*) es importante ya que establece cómo se inicia el modelo de agente y gestiona la ejecución de su comportamiento, su ciclo de vida y la capacidad de establecer comunicación mediante el lenguaje FIPA-ACL.

5.2.2. Particiones de agentes

Los LOAs han sido agrupados en particiones específicas a fin de organizar mejor los agentes que pertenecen a los ecosistemas de IoA. Estas particiones se describen como sigue:

- Partición 1 - *Agents of Things* (Agentes de Objeto/Cosa). Esta partición sólo incluye agentes de tipo LOAs que tienen asociado un objeto IoT y cuya misión es manejar los recursos que cada objeto integra. Algunos ejemplos de este tipo agentes pueden ser: agente de temperatura, agente vinculado a un sensor de humedad y agente vinculado a una bombilla.
- Partición 2 – *Agents of Coordination* (Agentes de Coordinación). Esta partición incluye agentes cuyo propósito es realizar actos de comunicación con agentes de tipo *Agent of Things*, con el fin de coordinar los procesos necesarios para ejecutar tareas que estén alineadas con el logro de los objetivos de control propuestos por la aplicación. Entidades como: agente de gestión de confort, agente de control de iluminación y agente de gestión de seguridad pueden ser algunos ejemplos de agentes pertenecientes a esta categoría.
- Partición 3 - *Agents of Data* (Agentes de Datos). Este tipo de agente está orientado a procesar cualquier tipo de dato capturado por las entidades *Agent of Things* con el fin de generar información y conocimiento para la toma de decisiones relacionadas con el uso de los recursos IoT. Algunos agentes de este tipo pueden ser los siguientes: agente de fusión de datos, agente de predicción y agente de historización de datos.
- Partición 4 - *Agents of People* (Agentes de personas/usuarios). Este tipo de agente proporciona una interfaz (i.e., usando URIs) que habilita la comunicación entre el usuario final y la aplicación IoA. De este modo, los usuarios pueden interactuar y personalizar sus necesidades a través de este tipo de agente. Dos ejemplos de agentes pertenecientes a esta partición pueden ser: agente de configuración de usuarios y agente de monitorización.

5.3. Linked Agent Contract

Normalmente se han utilizado contratos de software (contratos digitales) para formalizar las relaciones sociales y establecer el proceso de negociación y resolución de conflictos en el dominio de agentes [17, 43, 103]. Sin embargo, en este trabajo se ha planteado la re-definición de esta unidad orientada hacia un contexto semántico. En este contexto, proponemos el uso de un contrato de software como una unidad compacta de descripción orientada a la representación de entidades de agentes de forma estándar [88, 112]. Así, los agentes en el dominio IoA pueden tener suficiente información sobre el ecosistema en el que están operando y, en consecuencia, pueden realizar procesos de descubrimiento semántico basados en el significado de las cosas. A partir de los resultados recuperados por el proceso de descubrimiento, los agentes pueden tomar decisiones autónomas y más inteligentes [154].

Una vez especificado el uso del contrato en el campo de las tecnologías orientadas a agentes, proponemos el uso de un contrato de tipo *Linked Agent Contract* (LAC) como unidad básica de descripción semántica para el IoA. Un LAC es un documento dinámico enlazado que almacena una instancia de la ontología IoA-OWL para proveer a terceros de información semántica del agente y de su objeto IoT vinculado.

Todo LAC sigue la formalización de la ontología IoA-OWL. Por lo tanto, como detalla la ecuación 5.2, un LAC se representa de manera formal a partir de una tupla formada por siete elementos, donde id_i corresponde al identificador único del contrato (es el mismo identificador del LOA), P_i la sección contractual *AgentProfile*, C_i la sección contractual *AgentContext*, V_i la sección contractual *AgentService*, M_i la sección contractual *AgentModel*, T_i la sección contractual *AgentSmartThing* y finalmente S_i la sección contractual *AgentSocial* del contrato. Una sección contractual equivale a un conjunto finito de descriptores asociados a un componente de la ontología IoA-OWL.

$$LAC_i < id_i, P_i, C_i, V_i, M_i, T_i, S_i > \quad (5.2)$$

Tomando como base la representación formal de la ecuación 5.2, podemos definir un LAC como un documento contractual que se descompone en siete documentos, seis de ellos son empleados para almacenar los metadatos de cada uno de los componentes de la ontología IoA-OWL

y el restante para organizar las URIs que enlazan los seis documentos anteriores.

El LAC sigue las especificaciones de un documento abierto enlazado. Esto significa que un LAC cumple con los cuatro principios del enfoque LOD tales como: **(i)** el uso de URIs como nombres de cosas, **(ii)** uso de URIs basados en el protocolo https, **(iii)** uso de estándares y **(iv)** la inclusión de enlaces a URIs externas [33]. Por otra parte, los LACs fueron desarrollados cumpliendo con las mejores prácticas de JSON-LD [251], el cual es actualmente un estándar para expresar datos enlazados en formato JSON. Entre estos principios se destacan los siguientes: **(i)** publicación de datos utilizando JSON, **(ii)** uso de identificadores bien conocidos para describir datos, **(iii)** almacenamiento en caché de contextos JSON-LD, **(iv)** identificación de objetos con un identificador único, **(v)** vinculación de entidades mediante URIs, y **(vi)** uso de términos escritos para referencias externas. La aplicación de estos principios garantiza la interoperabilidad entre los contratos y también proporciona los siguientes beneficios:

- Formato fácil y ligero, legible tanto por humanos como por máquinas. Esto permite que tanto los usuarios como los agentes pueden saber qué y cómo funciona un agente específico. En el caso de los usuarios, éstos pueden entender los metadatos que describen el comportamiento de los agentes; mientras que en el caso de las máquinas, éstas pueden procesar los archivos de contratos de manera más eficiente que si se emplearan otros formatos para la descripción de recursos en la Web como XML.
- Descripción de los agentes compatibles con las directrices relacionadas con el enfoque LOD. La adopción de JSON-LD como formato para el almacenamiento persistente de los contratos de agente facilita enormemente el proceso de enriquecimiento semántico dinámico (asignación de valores a descriptores que definen un componente ontológico) de cualquier entidad relacionada con el IoA. Los agentes de IoA se describen reutilizando el mismo vocabulario. En consecuencia, el reaprovechamiento de los datos publicados en la Web se maximiza.
- Compatibilidad con el formato de datos de la Web Semántica. El uso del formato JSON-LD no sólo incluye beneficios con respecto a

la reutilización de datos vinculados disponibles en la web global. A partir de este formato es posible que la información sea representada en tripletas RDF—formato que permite la interoperabilidad entre aplicaciones que intercambian información en la Web Semántica. Esto es una ventaja porque los procesos de razonamiento se llevan a cabo en función de los descriptores almacenados en el contrato del agente.

- Generación automática de agentes. A partir de los descriptores semánticos almacenados en los contratos es posible generar los agentes de forma automática, ahorrando esfuerzos de programación. En este sentido, el enfoque del diseño por contratos (*Design by Contracts*, DbC por sus siglas en inglés) se afianza en el IoA [176, 196]. Esto ahorra tiempo de desarrollo de los sistemas de IoA y los usuarios sin conocimientos técnicos en el área de agentes podrían también crear sus propios sistemas.

5.3.1. Estructura general del contrato

Los LACs son generados automáticamente por una herramienta especializada llamada *IoA-Building Model*. Esta herramienta —desarrollada a partir de la especificación UML ilustrada en el Anexo E— construye los contratos de tipo LAC siguiendo la sintaxis establecida por JSON-LD. En el fragmento de código 5.1, se ilustra el esquema y estructura general de un LAC. Las tres partes básicas de un contrato se describen a continuación:

En primer lugar figura el contexto del contrato. Las líneas 2-13 incluyen el contexto del documento enlazado, en el que se definen los prefijos utilizados como abreviaturas en el cuerpo del contrato. Esto simplifica considerablemente la escritura del documento y facilita a los usuarios la comprensión del mismo si necesitan trabajar directamente con su contenido. Además, el contexto también define el idioma por defecto @ para describir cada uno de los componentes del contrato. En este caso particular, se definió el lenguaje inglés para describir los atributos JSON-LD que componen el LAC. Sin embargo, es posible describir cada recurso descrito en el contrato adoptando otros lenguajes naturales como español, chino, japonés, entre otros.

FRAGMENTO DE CÓDIGO 5.1: Representación de un LAC

```

1. {
2.   "@context":
3.     {
4.       "ioa": "http://core.ugr.es/ioa/owl/ioagent.owl#",
5.       "aprof": "http://core.ugr.es/ioa/owl/agentprofile.owl#",
6.       "acon": "http://core.ugr.es/ioa/owl/agentcontext.owl#",
7.       "asoci": "http://core.ugr.es/ioa/owl/agentsocial.owl#",
8.       "aserv": "http://core.ugr.es/ioa/owl/agentservice.owl#",
9.       "athing": "http://core.ugr.es/ioa/owl/agentosmartthing.owl#",
10.      "amodl": "http://core.ugr.es/ioa/owl/agentmodel.owl#",
11.      "repo": "http://rt.ugr.es:5984/ioahost/",
12.      "@language": "en"
13.    },
14.   "_id": "ioa_27ad68d2b2eb00fcfd8b82263403209",
15.   "aprof:AgentProfile": {
16.     "@type": ["aprof:AgentProfile"],
17.     "@id": "repo:profile_27ad68d2b2eb00fcfd8b82263403209"
18.   },
19.   "acon:AgentContext": {
20.     "@type": ["acon:AgentContext"],
21.     "@id": "repo:context_27ad68d2b2eb00fcfd8b82263403209"
22.   },
23.   "athing:AgentSmartThing": {
24.     "@type": ["athing:AgentSmartThing"],
25.     "@id": "repo:smartthing_27ad68d2b2eb00fcfd8b82263403209"
26.   },
27.   "amodl:AgentModel": {
28.     "@type": ["amodl:AgentModel"],
29.     "@id": "repo:model_27ad68d2b2eb00fcfd8b82263403209"
30.   },
31.   "aserv:AgentService": {
32.     "@type": ["aserv:AgentService"],
33.     "@id": "repo:service_27ad68d2b2eb00fcfd8b82263403209"
34.   },
35.   "asoci:AgentSocial": {
36.     "@type": ["asoci:AgentSocial"],
37.     "@id": "repo:social_27ad68d2b2eb00fcfd8b82263403209"
38.   }
39. }

```

En segundo lugar se incluye los datos de identificación del contrato. Esta parte del LAC incluye dos IRIs (Internationalized Resource Identifiers) específicos para definir el identificador (@id) del contrato y el tipo (@type) de entidad (líneas 14-15). El identificador fue definido a través de un Identificador Universal Único (UUID) generado por el motor de base de datos CouchDB (<http://couchdb.apache.org/>) que es único para cada contrato. A partir de este identificador, los contratos son accesibles a través de URIs.

En último lugar se encuentran los enlaces a las secciones contractuales que definen un LAC. Las líneas 15-18 enlazan con la sección del perfil del

agente, las líneas 23-26 se enlazan con la sección de contexto del agente, las líneas 27-30 con la sección modelo del agente, y así sucesivamente con las secciones contractuales restantes.

5.3.2. Estructura de las secciones contractuales

Las seis secciones contractuales que forman parte del contrato general previamente detallado adoptan la misma sintaxis y esquema. Sin embargo, en lugar de enlazar con las secciones contractuales como hace el contrato general, una sección contractual se conecta a los datos enlazados que describen cada componente integrado por la ontología IoA-OWL. Para ilustrar estos enlaces a algunos de los *datasets* de IoA, en el fragmento de código 5.2 presentamos un extracto de la sección del perfil del LOA (*AgentProfile*).

FRAGMENTO DE CÓDIGO 5.2: Representación de un fragmento del componente AgentProfile

```

1. {
2.   "@context":
3.     {
4.       "rtre": "http://core.ugr.es/ioa/owl/RTrestriction.owl#",
5.       "aprof": "http://core.ugr.es/ioa/owl/agentprofile.owl#",
6.       "ioa": "http://core.ugr.es/ioa/owl/ioagent.owl#",
7.       "foaf": "http://xmlns.com/foaf/0.1/",
8.       "dset": "http://core.ugr.es/ioa/datasets/",
9.       "@language": "en"
10.    },
11.    "_id": "profile_27ad68d2b2eb00fcfd8b82263403209",
12.    "_rev": "1-78d96946030c9634e7827668da620f16",
13.    "ioa:presents": {
14.      "@type": ["aprof:AgentProfile"],
15.      "foaf:name": "AComfortTemperature",
16.      "foaf:status": "Active",
17.      "aprof:description": "Comfort in building A1 Floor A2",
18.      "aprof:hasOwner": {
19.        "@type": ["aprof:Owner"],
20.        "@id": "dset:Owner/SmartTech"
21.      },
22.      "rtre:hasRTrestriction": [
23.        {
24.          "@id": "dset:RTparameter/execution-time",
25.          "@type": ["rtre:RTrestriction"],
26.          "rtre:notation": "less than",
27.          "rtre:paramvalueRT": 8
28.        }
29.      ]
30.    }

```

La parte contextual de la sección *AgentProfile* se muestra en las líneas 2-10, los identificadores de sección se definen en las líneas 11-12 y, por último, los descriptores de las secciones se introducen en las líneas 13-29. Específicamente, las líneas 18-21 muestran un enlace al proveedor del agente a través de su identificador correspondiente. Desde este identificador, es posible acceder a toda la información complementaria del proveedor como su nombre, contacto, entre otros, si es necesario. Esto significa que un LAC reutiliza descriptores y sólo almacena la información básica de la que es posible enlazar con los conjuntos de datos a los que se debe acceder para obtener datos más detallados de las entidades incluidas en los contratos. Asimismo, las líneas 22-28 se refieren a un parámetro en tiempo real llamado tiempo de ejecución (*execution-time*). Este es un parámetro previamente almacenado en un conjunto de datos público de IoA que incluye detalles adicionales de la entidad del parámetro como su descripción y unidad de tiempo. Las secciones restantes de un LAC se describen de forma similar pero enlazando los datasets correspondientes.

5.4. Workflow for Agent Control

El concepto de flujo de trabajo se introdujo en la formalización del componente ontológico *AgentModel*. El flujo de trabajo, denominado *Workflow for Agent Control* (WAC), representa una entidad importante de los LOAs que modela su comportamiento. Aunque este concepto puede diferir entre los desarrolladores, un WAC es una descripción secuencial de las acciones que un agente debe ejecutar para lograr sus objetivos. De manera formal, un WAC es representado mediante una tupla formada de 3 elementos como se detalla en la ecuación 5.3, donde id_i es el identificador del flujo de trabajo, $name_i$ su nombre y finalmente, $IFTTRecipe_i$ un conjunto finito de bloques IFTTT (If-This Then-That) [124] que el LOA ejecuta para cumplir sus objetivos.

$$WAC_i < id_i, name_i, IFTTRecipe_i > \quad (5.3)$$

5.4.1. Recetas IFTTT orientadas a objetos de IoT

La ejecución de acciones en el ecosistema de IoA se ha desarrollado un módulo especializado en gestionar este mecanismo de automatización empleando micro-recetas y macro-recetas siguiendo las directrices de la tecnología IFTTT (<https://ifttt.com/>).

La estructura genérica de una receta IFTTT (micro-receta), descrita en la ecuación 5.4, se compone de los dos componentes básicos de un proceso condicional simple tales como antecedente (*trigger*) y consecuente (*action*). Así, cuando ocurre el evento (*trigger_i*), automáticamente se ejecuta la acción predeterminada (*action_i*). La acción de ambos componentes de la receta son usualmente ejecutadas a través de canales (*channel*) compatibles y el tipo de canal a usar depende del componente de la receta que lo ejecute. En este sentido, el antecedente de la receta usa un canal orientado a sensor (*lectura de magnitud*); mientras que en el caso del consecuente, éste emplea un canal orientado a actuador (*modificación de una condición*).

$$SimpleReceta_{IFTTT} < Trigger_i, Action_i > \quad (5.4)$$

Los canales IFTTT manejados por el módulo desarrollado están orientados a gestionar las operaciones asociadas a objetos de IoT mediante una interfaz REST. De manera general, los elementos que forman un canal orientado a objeto (ecuación 5.5) se constituye de 2 elementos mandatorios (nombre del objeto y la propiedad del objeto a leer o modificar) y 1 optativo (valor de la propiedad a modificar, solo para canales que controlan actuadores).

$$Canal_i < Object_name_i, Propiedad_i, NewValue(optional) > \quad (5.5)$$

A modo de ejemplificación, un canal orientado a sensor especializado en leer el valor de la temperatura de un sensor estaría definido por el nombre del objeto *TemperatureSensor* y la propiedad *value*; mientras que en el caso de un canal orientado a actuador especializado en apagar una bombilla, éste estaría definido por el nombre del objeto *LightBulb* y la propiedad *state*. En este último caso, el argumento optativo *NewValue* es necesario de que sea incluido con valor “OFF” para que el estado tome dicho valor.

La especificación UML en la Figura 5.4 no sólo muestra los componentes de una receta IFTTT genérica (*SimpleMicroRecipe*); sino que también el diagrama incluye las entidades que permiten el modelado de micro-recetas anidadas (*NestedMicroRecipe*). Este tipo de recetas permite que más de un canal pueda ser ejecutado en el antecedente y consecuente de la receta, los cuales están unidos mediante operadores lógicos de tipo AND y OR. De esta manera, es posible ejecutar un conjunto de n acciones si se cumplen una conjunto de m condiciones gestionadas a través de canales orientados a objetos. Para el caso de los operadores OR y AND en el lado del antecedente, éstos funcionan de igual forma a como operan en una estructura condicional anidada; mientras que los mismos operadores en el lado del consecuente de la receta, el operador AND implica que se deben ejecutar todas las acciones y si es el caso del operador OR, se ejecutará únicamente una de las acciones que es elegida de forma aleatoria.

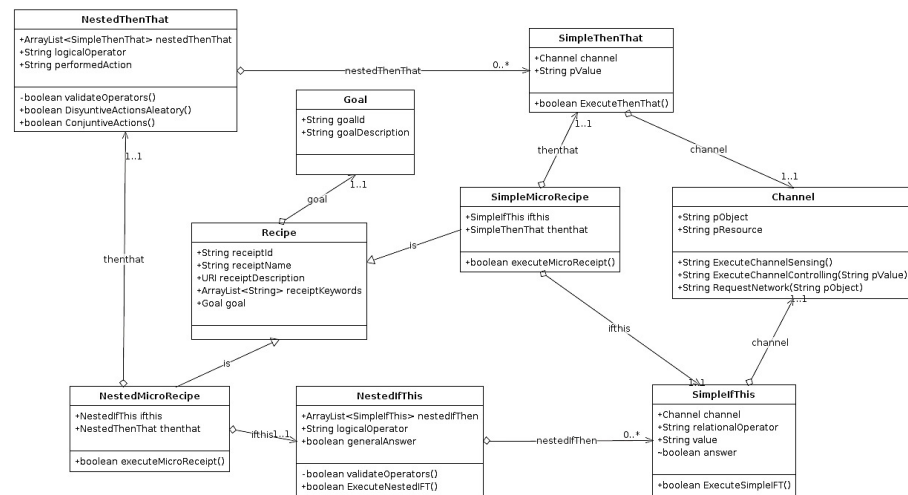


FIGURA 5.4: Especificación UML de una micro-receta IFTTT simple y anidada orientada a objetos de IoT.

Como complemento a las micro-recetas IFTTT simples y anidadas, el módulo de adaptación del agente propuesto permite la definición de macro-recetas IFTTT (Figura 5.5) las cuales organizan un conjunto de micro-recetas en un único componente. Esto facilita la planificación de las tareas que el objeto debe desempeñar a través del agente que tiene

embebido. Esta planificación se lleva a cabo mediante el modelado de recetas especiales (*PlannedSingleMicroRecipe* y *PlannedNestedMicroRecipe*) que gestionan un nivel de prioridad y que un algoritmo específico emplea para controlar la macro-receta. Hemos aprovechado este concepto para definir el comportamiento del agente ya que cada micro-receta puede ser usado para modelar uno de sus objetivos específicos y la macro-receta para definir su objetivo general—a nivel de control de los sensores del objeto de IoT.

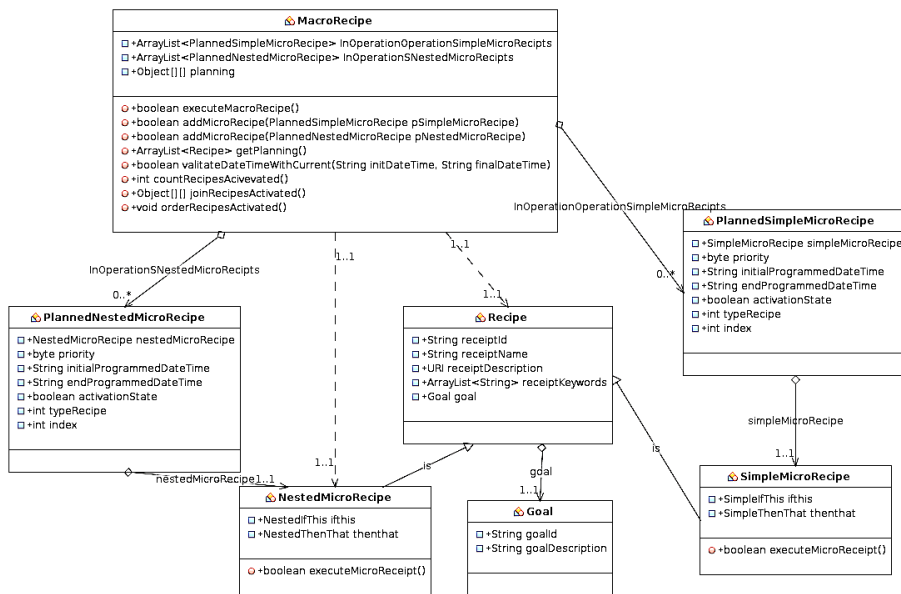


FIGURA 5.5: Especificación UML de una macro-receta IFTTT orientada a objeto de IoT.

5.4.2. Estructura de un flujo de control

En términos generales, un WAC puede incluir acciones de control específicas, invocaciones de operaciones a objetos IoT o peticiones a otros agentes contrapartes. Así, un flujo de control estructura sus acciones organizadas en un conjunto de bloques de control. Cada bloque de control especifica las acciones de control según los patrones *If-This Then-That*. La porción de un flujo de trabajo, mostrado en el fragmento de código 5.3, describe el contenido de un bloque de control que consiste

en una estructura If-This (líneas 5-15) y Then-That (líneas 16-32). Esta estructura permite establecer una regla que debe garantizar el LOA; es decir, si se cumple la condición, se aplica una acción.

FRAGMENTO DE CÓDIGO 5.3: Esquema general de un WAC siguiendo el formato JSON

```

1. {
2.   "workflow": {
3.     "workflow name": "control-HVAC",
4.     "workflow description": "collaborative control HVAC",
5.     "if-this": {
6.       "channel": {
7.         "channel-data-needed": "temperature",
8.         "channel-data-unit-measurement": "C",
9.         "channel host": "https://xx.com",
10.        "channel object": "temperature_c2",
11.        "channel resource": "state"
12.      },
13.      "if-then relational operator": "GREATHER THAN",
14.      "if-then value": 21
15.    },
16.    "then-that": [
17.      {
18.        "then-that logical operator": "AND"
19.        "channel": {
20.          "channel host": "https://yy.smarthome.com",
21.          "channel object": "ext_central_hvac_c2",
22.          "channel resource": "state"
23.        },
24.        "then-that value": "ON"
25.      },
26.      {
27.        "channel": {
28.          "channel host": "https://yy.smarthome.com",
29.          "channel object": "shutters_c2_f3",
30.          "channel resource": "state"
31.        },
32.        "then-that value": "0"
33.      }
34.    ]
35.  }
36. }

```

Las condiciones y acciones pueden especificarse en base a entidades de tipo agente u objeto IoT, y utilizando la unidad de canal para expresar, por ejemplo, una lectura de datos de sensor o una acción de control del actuador. De hecho, el canal describe metadatos de objetos que se encuentran fuera del contexto del LOA, pero que son accesibles desde el LOA. Por ejemplo, en el canal ilustrado en las líneas (18-22), la información del canal está completa; es decir, el HVAC es gestionado por un objeto IoT llamado *ext_central_hvac_c2* accesible en *https://yy.smarthome.com*.

Sin embargo, un canal podría tener alguna información incompleta como ocurre en el canal representado en las líneas (9-11). En este último caso tenemos información sobre los datos requeridos (temperatura) pero no sobre cómo se consigue. Entonces, la información restante sobre el agente puede ser desconocida inicialmente. Los metadatos desconocidos, como el nombre del agente, el *host* y el puerto de la plataforma del agente, pueden obtenerse durante el proceso de descubrimiento del agente, o directamente del LAC en el caso del objeto IoT, que es gestionado por el propio LOA.

5.5. Arquitectura de referencia para IoA

El esquema mostrado en la Figura 5.6 constituye nuestra propuesta de un Modelo de Referencia para IoA. Se trata de una arquitectura de 5 capas formada por: una capa de IoT (*IoT layer*), una capa de gestión de las comunicaciones a nivel de objetos de IoT (*IoT-communication layer*), una capa de servicios (*IoS layer*), una capa de agentes (*IoA layer*) y una capa semántica (*semantic layer*) al más alto nivel de abstracción. Este modelo se ha creado siguiendo una filosofía en la que cada objeto de IoT tiene asociado un LOA para aumentar el nivel de inteligencia e interoperabilidad de IoT. Este agente asociado se ejecuta en el servidor donde se implementa el sistema multiagente.

En el nivel de abstracción más bajo de la arquitectura, se encuentra la capa *IoT Layer*, también denominada como capa de sensor/actuador (*Sensor/Actuator Layer*). Esta capa gestiona principalmente los sensores y actuadores integrados en los dispositivos/objetos/cosas conectados a una infraestructura física de IoT [186]. Es importante destacar que los dispositivos conectados deben ser compatibles con las interfaces y estándares de comunicación cableadas o inalámbricas del IoT para que su correspondiente agente enlazado pueda establecer procesos de comunicación y, de este modo, el agente pueda ejecutar las acciones de control adecuadas cuando el objeto específico asociado.

La capa *IoT-Communication Layer* establece comunicación entre los objetos de IoT y la capa siguiente por medio de protocolos de comunicación inalámbrica como *ZigBee*, *WiFi*, *WiMax*, *3G* y *4G* celular, entre otros [106, 163]. Tanto la capa *IoT Layer* y la subsecuente capa *IoT-Communication Layer* se han tomado como referencia de la arquitectura

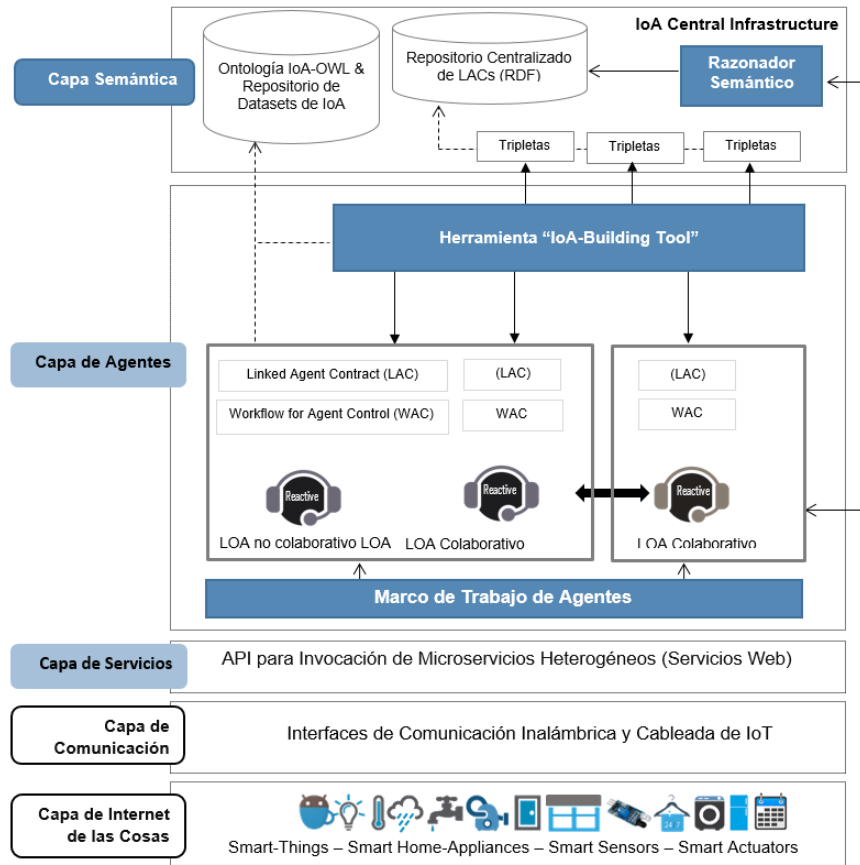


FIGURA 5.6: Arquitectura de referencia propuesta para el IoA

propuesta para el IoT. Esto significa que no hemos trabajado a este nivel. Las cuestiones de este tipo se pueden resolver utilizando cualquier middleware para crear sistemas de IoT existente en la actualidad, como es el caso de: *UbiSOAP*, *Ubiware* y *OpenHAB*, entre otros [220].

Los servicios web son, sin duda, los recursos computacionales más importantes que los agentes pueden invocar para controlar directamente los objetos de IoT, acceder a los datos adquiridos de los sensores o ejecutar procesos de gestión en caso de ser necesario. Por lo tanto, la capa servicios (*IoS Layer*) incorpora mecanismos para consumir servicios web heterogéneos pertenecientes a diferentes ecosistemas de servicios distribuidos incluyendo el suyo propio. La implementación de los servicios web

por cada objeto de IoT suele estar alojada en la nube o desplegada en la propia arquitectura. Aún así, esta capa implementa un mecanismo especializado en la invocación de microservicios heterogéneos mediante un mecanismo de invocación uniforme. Para cubrir este problema específico, recomendamos utilizar una interfaz de programación de aplicaciones (API) capaz de invocar servicios web heterogéneos que forma parte de este trabajo [212]. A partir de ésta, se pueden componer flujos de trabajo mediante invocaciones de servicios *SOAP*, *RESTful*, *DPWS* y *DOHA* de manera sencilla, uniforme y amigable.

La siguiente capa es la capa de agentes (*IoA Layer*). Esta capa abarca un conjunto de LOAs. Cada LOA tiene un LAC donde se describen sus propiedades y un WAC para definir su comportamiento. Tanto el LAC como el WAC se publican de manera local en un entorno de ejecución de agentes. Esta infraestructura puede almacenar la información de varios LOAs al mismo tiempo. Además, esta capa permite que los LOAs también puedan almacenar y actualizar su LAC como datos globales en el ecosistema de IoA para permitir a los agentes ejecutar procesos de razonamiento y así descubrir eventuales agentes contrapartes.

En cuanto a la selección de la herramienta de software para el desarrollo de LOAs, se recomienda emplear los marcos de trabajo de agentes más extendidos basados en bases teóricas validadas (i.e, Jack, JADE, Jadex y Jason, entre otros) para tener compatibilidad con aplicaciones ya desarrolladas [147]. Por eso sugerimos la selección de JADE [28] para agentes reactivos y Jadex [214] para agentes deliberativos. Sus documentación, descritas respectivamente en [108] y [42], han sido validadas por la comunidad científica y guardan compatibilidad con el estándar de comunicación ACL propuesto por FIPA.

Finalmente, la capa semántica (*Semantic Layer*) se encarga de gestionar la coexistencia del conocimiento semántico y los datos enlazados, con el fin de ejecutar el razonamiento semántico solicitado por los LOAs desde la capa *IoA Layer* en un formato compacto. Esta capa incluye dos repositorios. El primero es el *Centralized-Repository of LACs* que almacena los LACs para en función de su contenido llevar a cabo las peticiones de razonamiento semántico sobre el ecosistema de IoA. Por otro lado, el segundo repositorio, *IoA Datasets Repository*, almacena la ontología IoA-OWL y los conjuntos de datos propios del ecosistema IoA (i.e., parámetros, propietarios, certificados) que pueden ser reutilizados al

momento de que se requiere enriquecer semánticamente nuevas entidades LOAs.

5.6. Procesos y mecanismos del modelo

5.6.1. Proceso de construcción de LOAs

El proceso de creación de LOAs es llevado a cabo mediante la herramienta *IoA-Building Tool*. La creación de un LOA requiere la preparación de la plataforma de agentes para ser utilizada como hábitat antes del despliegue de las instancias LOAs. Posteriormente, la creación de los LOAs es técnicamente factible siguiendo los pasos ilustrados en el Diagrama de Secuencia UML de la Figura 5.7.

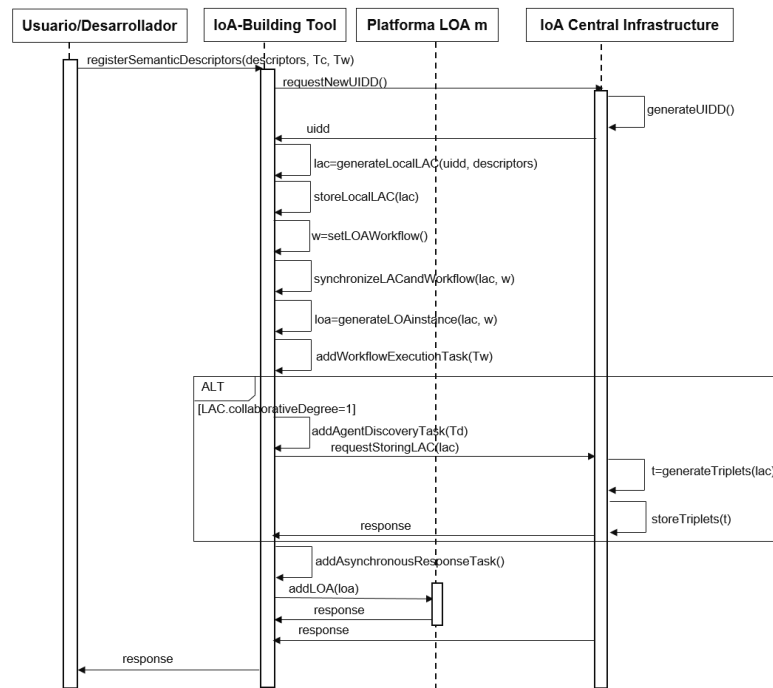


FIGURA 5.7: Diagrama de secuencia UML para la creación de un LOA.

El proceso de creación del LOA requiere la participación del desarrollador/usuario en una etapa preliminar. Su tarea principal es recolectar

los datos asociados a los descriptores semánticos que definen la instancia del LOA modelado usando una interfaz GUI. Una vez especificados los descriptores, se crean los elementos arquitectónicos básicos para construir el LOA.

La primera etapa del proceso de creación del LOA es la identificación del LOA. En esta etapa se solicita un UUID a la infraestructura de IoA, *IoA Central Infrastructure*, para generar el LAC local el cual será almacenado en la plataforma de agentes.

En la siguiente etapa, el LAC se genera teniendo en cuenta los descriptores semánticos proporcionados por el usuario. El LAC se valida comprobando que los descriptores que enlazan con los datasets de IoA que estén publicados de forma pública. Algunos de descriptores son completados asignando valores predeterminados según el caso (i.e., medidas de centralidad).

Tanto los documentos de descripción (LAC) como el flujo de trabajo (WAC), deben sincronizarse antes de crear el LOA que los contiene. De hecho, el flujo de trabajo puede tener datos incompletos sobre sus canales en relación con objetos de IoT. Estos datos se completan con los metadatos del componente *AgentSmartthing* integrado en el LAC. Un ejemplo de un canal de un flujo de trabajo que se puede completar con estos metadatos se ilustra en las líneas 31-37 del fragmento de código 5.3. Asimismo, los datos de interés de la sección *AgentSocial* del LAC se actualizan a partir de la información de datos de interés disponible en los canales de flujo de trabajo.

Al final, el conjunto de tareas colaborativas del LOA se configura seleccionando el tiempo de control (ejecución de la tarea *Workflow Execution Task*), y el tiempo de descubrimiento (ejecución de la tarea *Agent Discovery Task*) y, a continuación, ambas se planifican. Luego, se agrega una tarea para que el agente se comunique con entidades externas (*Agent Asynchronous Response Task*). Una vez que se han completado todas las etapas, el LOA es añadido a su plataforma correspondiente y se activa para cumplir sus objetivos.

5.6.2. Proceso de descubrimiento semántico

Los metadatos del LAC almacenados a escala global (centralizada) permiten que los LOAs puedan ser descubiertos por entidades externas del ecosistema de IoA. El formato utilizado para el almacenamiento de

estos metadatos puede variar en función de los recursos tecnológicos disponibles. Sin embargo, es obligatorio que los metadatos asociados al LAC mantengan la compatibilidad con los formatos utilizados por los programas orientados al soporte de datos enlazados tales como: SPARQL, D2RQ y Virtuoso.

Dos formatos alternativos que podrían ser utilizados son los formatos JSON-LD y las tripletas RDF. El primero corresponde al formato JSON para publicar datos enlazados (fragmento de código 5.2); mientras que el segundo corresponde a un formato ampliamente difundido en la Web Semántica cuyo esquema está formado por tres elementos—sujeto, predicado y objeto— $T_i < subject_i, predicate_i, object_i >$. Para ilustrar el uso de este último formato, en el fragmento de código 5.4, presentamos el contenido equivalente del fragmento de código 5.2 en formato tripletas RDF.

FRAGMENTO DE CÓDIGO 5.4: Representación del LAC
en formato RDF.

1. Triplet#1 `<ioa#ioa_27ad68d2b2eb00fcfd8b82263403209>`
`<ioa#presents>`
`<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
2. Triplet#3 `<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
`<aprof:description>`
`<Provide comfort in building A1 Floor A2>`
3. Triplet#2 `<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
`<foaf:name>`
`<AComfortTemperature>`
4. Triplet#4 `<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
`<aprof:status>`
`<active>`
5. Triplet#5 `<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
`<aprof:hasOwner>`
`<http://core.ugr.es/ioadatasets/Owner/SmartTech>`
6. Triplet#6 `<aprof:profile_27ad68d2b2eb00fcfd8b82263403209>`
`<aprof:hasRTrestriction>`
`<dsetrt:27ad68d2b2eb00fcfd8b82263403209_worts-case>`
7. Triplet#7 `<27ad68d2b2eb00fcfd8b82263403209_worts-case>`
`<rt:notation>`
`<rt:less than>`
8. Triplet#8 `<ad84abb7b0fc0b697b8febb02502506_worts-case>`
`<rt:paramValueRT>`
`<8.0>`

El nombre del agente, su estado, descripción y proveedor, definidos en las líneas 15, 16, 17 y 20 del fragmento de código 5.2 son respectivamente equivalentes a las tripletas RDF 2, 3, 4 y 5 del fragmento de código 5.4. El total de tripletas RDF generadas a partir de un LAC depende del número de valores descriptivos introducidos por el desarrollador/usuario

durante el proceso de enriquecimiento semántico de un LOA. Este número de tripleta no se mantiene constante porque se pueden agregar o actualizar nuevos valores de descriptores después de la creación de LAC. Por ejemplo, algunos descriptores del componente *AgentSocial* como los datos de interés del agente y la lista de sus contrapartes se introducen dinámicamente en función de los cambios del entorno del agente.

La publicación del LAC en formato tripletas empodera a los LOAs para que puedan ejecutar tareas para lograr objetivos colaborativos. Un objetivo colaborativo es un objetivo alcanzado mediante la interacción de un conjunto finito de participantes (i.e., objetos IoT, agentes) en el que cada uno de ellos realiza tareas más sencillas que juntos pueden resolver una tarea compleja.

Cuando a un LOA se le asigna una tarea colaborativa, se debe identificar el conjunto de tareas que puede realizar con sus propios recursos y también debe identificarse qué tareas están fuera de su alcance. En este último caso, el LOA debe explorar el ecosistema de IoA de manera proactiva de acuerdo con un planificación a fin de identificar los agentes contrapartes que lo apoyan para lograr dicho objetivo. Este proceso de exploración también llamado proceso de descubrimiento—dirigido por el algoritmo de la Figura 5.8—es ejecutado por el agente a través de la tarea *Agent Discovery Task*. Gracias a la ejecución de esta tarea, se actualizan el propio LAC y WAC del agente. Por lo tanto, el LOA mantiene un estado actualizado y consistente con respecto al ecosistema de IoA.

En términos generales, el algoritmo de la Figura 5.8 requiere sólo un argumento de entrada. Este argumento es el nombre del LOA (α) que lanza el proceso de descubrimiento. El valor de retorno es una lista de agentes asociados (ρ) que coinciden con los requisitos actuales del agente α . Para crear esta lista, el agente α ejecuta un conjunto de operaciones basadas en el contenido de su LAC (ϕ) y su flujo de trabajo (ω). Por lo tanto, antes de comenzar el proceso de exploración del ecosistema IoA, el agente α debería asegurarse de que su flujo de trabajo todavía requiere actualizar algunos canales (línea 2); de lo contrario, no toma ninguna acción. De esta manera, el agente α puede verificar si su flujo de trabajo ya está completo o no. Un ejemplo de canales completos e incompletos se ilustra respectivamente en las líneas 20-26 y 7-14 del fragmento de código 5.3.

Algorithm 1 LOA discovery algorithm

```

Require: LOA_name ( $\alpha$ );
Ensure: partnership_list ( $\rho$ );
1:  $\omega \leftarrow \text{get\_workflow}(\alpha)$ ;
2: if ( $\text{has\_all\_metadata}(\omega) = \text{true}$ ) then
3:    $\rho \leftarrow \text{null}$ ;
4: else
5:    $\phi \leftarrow \text{get\_lac}(\alpha)$ ;
6:    $\mu \leftarrow \text{reasoning\_over\_ioa\_ecosystem}(\phi, \text{semantic\_arguments})$ ;
7:   if ( $\mu \neq \text{null}$ ) then
8:      $\rho \leftarrow \text{apply\_selection\_method}(\mu, \text{selection\_operator})$ ;
9:     for ( $\text{individual\_partner} \in \{1, \dots, \rho.\text{size}()\}$ ) do
10:       $\phi' \leftarrow \text{get\_parther\_lac}(\rho[\text{individual\_partner}])$ ;
11:       $\omega \leftarrow \text{update\_workflow\_channel}(\phi')$ ;
12:       $\phi \leftarrow \text{update\_lac\_social\_circle}(\phi')$ ;
13:     end for
14:   end if
15:    $\text{register\_discovery\_execution}()$ ;
16: end if
17: return  $\rho$ 

```

FIGURA 5.8: Algoritmo de descubrimiento ejecutado por los LOAs.

Luego, en la segunda etapa se ejecuta el núcleo del proceso de descubrimiento sobre el ecosistema IoA a partir de un descubrimiento semántico (línea 6). El descubrimiento semántico aplica un motor de razonamiento en una lista estratégica de descriptores semánticos como argumentos de entrada. Estos descriptores semánticos pueden ser seleccionados de cualquiera de las seis secciones de LAC del agente α . Dependiendo del operador de razonamiento (i.e., similar a, contextual, poder social, calidad de servicio) establecido como criterio de descubrimiento, se recupera un conjunto de agentes que satisfacen plenamente la solicitud. Por ejemplo, si el agente α necesita recuperar colaboradores mediante los datos de interés y el contexto de objeto de IoT, se debe seleccionar una combinación de descriptores semánticos de las secciones *AgentProfile*, *AgentSocial* y *AgentContext* que incluyan magnitud, unidad de medida, restricciones en tiempo real y datos referente a la localización. La formalización de estos descriptores fue descrita en el Capítulo 4.

Una vez que los agentes son recuperados a partir del descubrimiento

semántico, se realiza un proceso de selección de agentes contrapartes en la tercera etapa para descartar los peores LOAs recuperados (línea 8). Para llevarlo a cabo es necesario especificar el operador de selección a aplicarlo en la lista de agentes recuperados por la tarea de razonamiento. Algunos ejemplos de operadores de selección podrían ser mejores tiempos de respuesta, mayor poder social, entre otros. La aplicación de un operador específico también depende de las restricciones del agente y de la naturaleza del sistema modelado.

Finalmente, utilizando los metadatos de cada uno de los agentes que superaron el proceso de selección aplicado, el algoritmo debe actualizar tanto el LAC como el flujo de trabajo del agente α ; es decir, actualizar los metadatos de los canales de su flujo de trabajo (línea 11) y el círculo social del agente α (línea 12). Al final, el agente α incrementará el contador de descubrimiento exitoso (línea 15).

5.7. Calidad de servicio del modelo

Los aspectos relacionados con la satisfacción de la calidad de servicio (*Quality of Service, QoS*) y los requisitos de tolerancia a fallos son intrínsecos a enfoques tales como el IoT y el IoA debido a que ambos enfoques proponen modelar redes dinámicas en las que el uso de los recursos puede variar según el contexto [162]. Frente a esta afirmación, a continuación se detallan las principales implicaciones y métricas consideradas para el modelado de los ecosistemas de IoA encaminadas en la optimización del rendimiento de estos sistemas.

5.7.1. Mecanismos de calidad de servicio

La QoS en las redes de comunicación se define como la “*capacidad de la red para lograr un comportamiento más determinístico, de modo que los datos pueden ser transportados con una pérdida mínima de paquetes, un retardo y un ancho de banda máximo*” [16]. Sin embargo, Li et al. señalan que para los sistemas de IoT—y en nuestra opinión también para los sistemas de IoA—la definición de QoS no es completamente clara [162]. Esto se debe principalmente a que la definición de servicio en este tipo de redes difiere de las redes convencionales (i.e., redes de sensores inalámbricos). Por ejemplo, un servicio en IoT e IoA puede definirse

como una entidad de software autónoma que proporciona una forma de acceder a algunos recursos específicos (i.e., datos de sensores, actuadores) utilizando una interfaz uniforme, así como su procesamiento, la ejecución de acciones de control y la comunicación entre los objetos implicados.

Las métricas tradicionales de QoS como el rendimiento, el retardo o el *jitter* no son las más apropiadas a nivel de aplicación a pesar de que tienen dependencia [162]. Estas métricas son específicas para el nivel de la capa de red y el nivel físico de la red de IoT [56]; es decir, para esquemas de enrutamiento o de control de acceso medio (MAC), para enrutamiento consciente, enrutamiento de eficiencia energética y para flujos de tráfico. No obstante, el IoA está en un nivel más alto y, por lo tanto, debe manejar otras métricas. Estas métricas son generalmente similares a las de la QoS existente en los servicios web [162]—escalabilidad, rendimiento, fiabilidad, disponibilidad, flexibilidad, precisión, integridad, seguridad y privacidad [219]—; es decir, para mejorar la precisión de la información recuperada y la vida útil de los objetos de IoT vinculados a LOAs.

El IoA se centra principalmente en proporcionar un proceso de control más preciso mediante el uso de los datos más adecuados recuperados, cuidando de que los datos se transmitan y traten con un cierto nivel de seguridad para evitar el acceso discriminatorio a la información. Para ello, se han implementado algunas métricas de QoS en el modelo IoA las cuales tienen los siguientes alcances:

- *Exactitud en la recuperación de datos.* Los LOAs han asociado un mecanismo para descubrir contrapartes externos basado en un razonamiento semántico y en un proceso de selección. El primero recupera a los candidatos de contraparte a partir de un conjunto de descriptores semánticos recibidos como argumentos (descriptores LAC). El segundo aplica un criterio de selección (i.e., mejor tiempo de respuesta, contexto específico, poder social de las contrapartes) para proporcionar los agentes más adecuados dentro de un proceso colaborativo. Ambos procesos permiten la recuperación de contrapartes en IoA de manera más precisa que los sistemas actuales que operan basados en el descubrimiento de recursos de acuerdo a la sintaxis de los datos [34].
- *Disponibilidad.* Dado que el modelo IoA utiliza una capa de servicio (*Internet of Service Layer*) como parte de su arquitectura de

referencia, la propiedad de disponibilidad del servicio es heredada por el modelo IoA. Como resultado, los LOAs pueden utilizar componentes sin estado, como los servicios web, para llevar a cabo sus tareas de control sobre objetos de IoT. Además, todas las métricas asociadas a los servicios web especificados anteriormente también son heredadas por el IoA.

- *Confiabilidad y rendimiento.* Los LOAs tienen la capacidad de soportar la comunicación a nivel de agente sin pérdida de mensajes y de manera eficiente. Esto se debe a que la arquitectura LOA ha sido construida adoptando el framework JADE, el cual constituye un entorno de agente bastante eficiente y fiable para desarrollar agentes [59]. Además, debido a que los LOAs definen un WAC, el cual determina el comportamiento del agente, los LOAs tienen la capacidad de descubrir y seleccionar los agentes contrapartes que proveen un mejor tiempo de respuesta. En consecuencia, las acciones que se ejecutan sobre el ecosistema de IoA son abordadas por los agentes más eficientes.
- *Seguridad y privacidad.* Los LOAs pueden establecer procesos de comunicación de agente seguros basados en certificados digitales manejados mediante canales seguros. Adicionalmente, en cuanto a la privacidad de los datos, los LOAs manejan un contrato que almacena descriptores semánticos. Estos descriptores se almacenan según dos niveles de privacidad, privado y público. Esto evita que los datos confidenciales asociados con la operación del agente puedan ser accedidos por usuarios de agentes que no están autorizados a hacerlo. Es importante destacar que nuestro modelo propuesto no está dirigido a cubrir completamente los aspectos de seguridad y privacidad, sino a mejorar la precisión de la información recuperada. Por lo tanto, esta métrica puede mejorarse aplicando mecanismos de seguridad específicos para agentes o IoT como, por ejemplo, métodos de autenticación más complejos, consentimiento informado del usuario, mecanismos de encriptación y comunicación de datos a nivel de agente, nuevos niveles de privacidad de datos, autorización y control de acceso, y entre otros [10, 72].

Además de las métricas descritas anteriormente, el modelo de IoA también cumple otras métricas adicionales inherentes a las tecnologías

de IoT, ya que la arquitectura de referencia incluye una capa específica para manejar los objetos de IoT, como la capa *Internet of Things Layer*. Las principales métricas cubiertas por la capa de IoT dependen de las tecnologías empleadas para aplicarla. Sin embargo, las métricas de QoS que este enfoque hereda de IoT cubren temas tales como: interoperabilidad, escalabilidad, flexibilidad, adaptabilidad y seguridad [269].

5.7.2. Mecanismos de tolerancia a fallos

El aspecto de la tolerancia a fallos, así como los aspectos relacionados con la calidad de servicio, pueden gestionarse a diferentes niveles en IoT, por ejemplo, el nivel de red físico y lógico, el nivel de servicio y el nivel de aplicación. Para cubrir el aspecto relacionado con la tolerancia a fallos a nivel de servicio y de aplicación, se pueden emplear algunos mecanismos centrados en controlar los fallos de los procedimientos de software tales como: redundancia y diversidad, bloque de recuperación y manejo de excepciones [125]. Con el fin de cubrir este aspecto, los siguientes mecanismos han sido implementados en el IoA:

- *Manejo de excepciones.* Todos los procesos soportados por una plataforma IoA son gestionados por la herramienta *IoA-Building Tool*. Esta herramienta es un software que permite a los desarrolladores y usuarios, gestionar los elementos (LOAs, LACs y WACs) y las tareas adicionales que los agentes de apoyo ejecutan por sí mismos durante su ciclo de vida.

En cuanto a los elementos de IoA, la herramienta *IoA-Building Tool* ha sido desarrollada para poder interceptar los errores y advertencias, así como manejar las excepciones que pueden ser activadas durante su gestión. Estos procesos se clasifican de acuerdo a cada elemento de la siguiente manera: **(i)** procedimientos relacionados con LOAs—creación, borrado, comunicación, inyección de nuevas tareas de agente, restauración—, **(ii)** procedimientos asociados con LACs—creación, almacenamiento, consulta y alteración—, y **(iii)** procedimientos que manejan WACs—creación, enlace a LOA, invocación de canales, agentes y servicios web.

Por otro lado, respecto a los procesos extra de IoA, la herramienta *IoA-Building Tool* también realiza un control exhaustivo de las

excepciones sobre los siguientes módulos: **(i)** solicitud y procesamiento de datos proporcionados por el razonamiento semántico, **(ii)** conversión de los LACs JSON-LD a tripletes RDF, y **(iii)** ejecución de cada una de las etapas de la generación automática de los LACs y LOAs (i.e., comprobación de datos y URIs, validación de Identificadores Únicos Universales (UUIDs) y verificación de la integridad de plataformas de agentes).

- *Recuperación ante fallos.* La recuperación de los fallos se produce en IoA a dos niveles: a nivel de LOA y a nivel de objeto de IoT. En el primer caso, cuando un LOA solicita a un agente contraparte que no está disponible en el ecosistema IoA, éste intercepta una excepción. Luego, el LOA revisa su WAC, y ejecuta un proceso de descubrimiento para reemplazar el agente de falla con otro LOA funcional. Por otra parte, el segundo caso se produce cuando no se dispone de un objeto de IoT. Nuestro modelo propone vincular un LOA a cada objeto de IoT. Por lo tanto, si un objeto específico no está disponible, su agente gestor puede interceptar el error y notificarlo a cualquier otro agente de contraparte interesado. Este agente puede seguir ejecutando proporcionando valores o acciones anteriores del objeto IoT correspondiente o puede esperar para reconectar con el objeto IoT fallido.

5.8. Beneficios potenciales del modelo

En general, una de las principales contribuciones que el IoA introduce en el IoT es la adición de nuevas entidades llamadas LOAs. Los LOAs permiten que los objetos del IoT se transformen de objetos pasivos tradicionales en objetos activos. Por consiguiente, los LOA habilitan a los objetos para que sean capaces de actuar de forma autónoma, proactiva, social, colaborativa e inteligente. A continuación, se describen los principales beneficios del modelo propuesto:

- *Aplicaciones de IoT sociales y colaborativas (B_1).* La inclusión de un LAC junto con la capacidad de los LOAs para descubrir colaboradores permite que los objetos de IoT sean tratados de manera social y colaborativa. Por consiguiente, la colaboración entre objetos situados en ecosistemas heterogéneos de IoT que comparten

sus recursos es ahora posible bajo el control del propio LOA. Los LOAs adoptan un modelo social basado en el descubrimiento dinámico de agentes como lo hacen los objetos inteligentes propuestos por Fortino, Guerrieri y Russo [97].

- *Aplicaciones de IoT inteligentes (B_2)*. El proceso de descubrimiento ayuda a los LOAs a encontrar los recursos más adecuados mediante un proceso de *matching* inteligente que se basa en el significado de los descriptores del LAC y no sólo en la comparación de valores nominales (i.e., cadenas y números). Por lo tanto, la información obtenida a partir de los LOAs recuperados es más precisa que la mayoría de las actuales aplicaciones de IoT. De hecho, la aplicación de búsquedas semánticas permite recuperar datos dentro de un contexto que una búsqueda sintáctica no es capaz de encontrar. Además, los LOAs llevan a cabo procesos de razonamiento integrando un conjunto de descriptores de diferentes aspectos como el incluido por la ontología IoA-OWL y no sólo basado en el contexto como lo hace el modelo *Agent of Thing* [185].
- *Aplicaciones de IoT autónomas y proactivas (B_3)*. La agentificación de objetos de IoT a través de LOAs, junto con sus correspondientes LAC y WAC, permite al agente controlar de forma autónoma los objetos de IoT sin necesidad de supervisión humana. Por lo tanto, los LOAs también pueden ejecutar acciones de detección y control sobre el IoT de manera proactiva y a lo largo de todo su ciclo de vida. En resumen, los LOAs aprovechan las características de proactividad y autonomía inherentes a los agentes de software para poder alcanzar sus objetivos por sí mismos, como otros ecosistemas de IoT agentificados lo han realizado [97].
- *Aplicaciones de IoT más adaptables (B_4)*. Los LOAs aplican un mecanismo adaptativo en dos niveles para ajustar su objetivo de control. El primero está relacionado con la capacidad de cambiar o sustituir el contenido del WAC de un LOA en cualquier momento y en tiempo de ejecución, sin penalizar la ejecución del agente. Este proceso es abordado básicamente por los usuarios. El segundo depende del contenido actual del WAC. Basándose en este contenido, el LOA puede buscar los recursos más adecuados (i.e., LOAs externos, objetos de IoT) para satisfacer los objetivos del agente. En

ambos casos, el LOA debe actualizar su comportamiento a partir de los recursos recuperados que se hayan seleccionado. A diferencia de la mayoría de los otros modelos de agentes, el modelo LOA proporciona a los agentes un mecanismo que los usuarios pueden personalizar en tiempo de ejecución, tal y como Yu et al. [278] sugieren para adaptar el IoT.

- *Aplicaciones de IoT más flexibles y distribuidas (B_5)*. El soporte de SOA y ROA por parte de los LOAs permite desacoplar las funcionalidades del agente y distribuirlas a través de los servidores web. Así, la modularidad de los artefactos del agente puede ser explotada para proporcionar un mecanismo flexible que facilite la escalabilidad del comportamiento del LOA hacia una mejor calidad de servicio sin cambiar la estructura interna de los LOAs [255]. De esta manera, los LOAs se convierten en un componente más flexible, ya que distribuye las acciones de procesamiento (i.e., razonamiento, acceso a sensores y control de actuadores, procesamiento de datos y cualquier otro tipo de acciones que involucren algoritmos) a través de la web. Además, los nuevos sistemas pueden reutilizar estos componentes para su desarrollo.
- *Uniformidad para acceder y controlar aplicaciones de IoT (B_6)*. La aplicación de normas para el desarrollo de los ecosistemas del IoA es importante para contribuir a la mejora de la interoperabilidad. En este sentido, los LOAs soportan estándares orientados a manejar mecanismos uniformes de acceso (interfaz REST), comunicación (FIPA-ACL) y gestión de datos (conjuntos de datos JSON-LD y RDF) sobre ecosistemas de IoT. Esto facilita la interoperabilidad entre agentes heterogéneos y, en consecuencia, la comunicación con objetos conectados a ecosistemas heterogéneos de IoT. Aunque el uso de todos los estándares es importante, las comunicaciones FIPA son esenciales para interactuar con sistemas manejados por modelos heterogéneos como *IoT-a* [52], *agent of things* [185], *smart object* [97], e incluso con otros modelos de agentes genéricos como *JADE*.
- *Sistemas IoT con soporte de recuperación de fallos (B_7)*. Las características sociales, cognitivas, autonómicas y adaptativas en conjunto proporcionan un nuevo beneficio sobre los LOAs. Esta nueva ventaja se centra en las entidades capaces de recuperarse de los

fallos que se producen en entornos de IoT, sin necesidad de que el usuario supervise los procesos adaptativos que se producen. Este es un beneficio adquirido por los sistemas dinámicos como los modelos *smart objects* [97] o *agent of things* [185]—ambos significativos en el proceso de agentificación del IoT.

- *Mecanismos adicionales de seguridad y privacidad que los genéricos de IoT (B_8)*. Los aspectos asociados a la seguridad y privacidad de los datos en IoA son primordiales. Para cubrir estos temas, el modelo LOA gestiona ecosistemas IoA seguros en los que las comunicaciones a nivel de agente se llevan a cabo en base a credenciales de agente (seguridad), y LACs con descriptores públicos y privados (privacidad de datos). A partir de estos mecanismos, los LOAs crean sus círculos sociales y ejecutan procesos colaborativos seguros con la ayuda de agentes contrapartes. Sin embargo, el modelo LOA podría optimizarse aún más en términos de seguridad y gestión de la privacidad de los datos con métodos más complejos.

5.9. Limitaciones del modelo

A continuación se detallan algunas de las limitaciones inherentes del modelo de IoA propuesto:

- *Modelo basado en elementos específicos*. Aunque los agentes en un dominio de IoA pueden ser implementados por cualquier marco de trabajo de agente para crear los LOAs, deben estar estrechamente relacionados con la existencia de los documentos LAC y WAC. Esto significa que para que las LOAs operen consistentemente sobre el ecosistema de IoA, se deben hacer esfuerzos adicionales para definir el WAC de cada agente. En este sentido, se requiere el desarrollo de una herramienta especializada en la construcción de plantillas de WACs para objetos específicos que soporten diferentes niveles de control sobre los ecosistemas de IoT. De esta manera, los desarrolladores pueden utilizar WACs simples o complejos de acuerdo a sus necesidades.
- *Requerimiento de una interfaz más usable y accesible*. El éxito del enfoque IoA propuesto depende en gran medida de los descriptores

semánticos que los usuarios ingresan para llevar a cabo la generación de agentes. En esta línea, el prototipo de la herramienta *IoA-Building Tool* proporciona una interfaz gráfica simple basada en *widgets GUI* para realizar el proceso de enriquecimiento semántico de LOAs. Sin embargo, dicha interfaz podría mejorarse para que los aspectos dependientes del usuario y la interacción con la aplicación sea más sencilla y divertida. La nueva interfaz de interacción que se diseñe debe cumplir con los estándares actuales de accesibilidad web y usabilidad propuestos por el Consorcio W3C.

- *Federación del conocimiento semántico.* Los LOAs y sus respectivos LACs son almacenados de forma distribuida basados en plataformas de agentes. Sin embargo, el conocimiento semántico—usado por la infraestructura *IoA Central Infrastructure* para proveer razonamiento semántico—está actualmente almacenado en formato grafo RDF centralizado. Con el fin de extender también el concepto de distribución al campo de la persistencia del conocimiento semántico, se podrían utilizar grafos RDF federados que permitan la distribución del conocimiento y minimicen la complejidad de los procesos de razonamiento semántico solicitados por los LOAs.

La evaluación del modelo de agente LOA, propuesto en este capítulo es presentado en la Sección 7.

5.10. Discusión

Se han establecido las principales bases conceptuales y aspectos relacionados con el enfoque de Internet de Agentes (IoA). Con el fin de proporcionar un alto nivel de interoperabilidad dentro del IoT, mediante la integración de agentes de software, hemos propuesto un mecanismo novedoso basado, por un lado, en el uso de LOA como la unidad de proceso para el IoA y, por otro lado, en el empleo de LAC para describir semánticamente los LOAs y publicarlos como datos abiertos. Sugerimos estos mecanismos para mejorar el nivel de interoperabilidad semántica en el ámbito de los agentes y, en consecuencia, en las redes de IoT gobernadas por agentes.

Actualmente, los datos están tendiendo hacia los datos enlazados, y las tecnologías Web relacionadas con los agentes y el IoT se dirigen hacia la web Semántica. Por lo tanto, el IoA requiere de nuevos modelos, métodos y mecanismos para apoyar los nuevos avances tecnológicos emergentes de los sistemas de información. Con el fin de contribuir al proceso de integración de los agentes y el IoT, hemos propuesto un modelo novedoso y un marco de trabajo (Framework-IoA), útiles para aumentar la interoperabilidad semántica y la inteligencia en objetos y redes de objetos IoT inteligentes.

A nivel de descripción semántica de agente hemos re-definido el concepto de contrato LACs en el dominio de agente para describir y publicar información como datos abiertos a nivel global. De esta manera, es posible normalizar los conceptos empleados por los desarrolladores de aplicaciones IoA y lograr un alto nivel de interoperabilidad semántica. Además, a nivel funcional de agente hemos introducido el uso de flujo de trabajo asociado a los LOAs, a partir de los cuales se define la lógica para controlar los objetos IoT por medio de micro-recetas y macro-recetas de tipo IFTTT. Así, es posible crear agentes fácilmente adaptables en tiempo de ejecución.

5.11. Contribuciones relacionadas

El modelo y mecanismos descritos en este capítulo han sido planteados en los siguientes medios científicos:

P. Pico-Valencia, J.A. Holgado-Terriza and J. Senso. Towards an Internet of Agents Model Based on Linked Open Data Approach. *Journal of Autonomous Agents and Multi-Agent Systems* [En Revisión].

P. Pico-Valencia and J.A. Holgado-Terriza. Integration of MultiAgent Systems with Resource-Oriented Architecture for Management of IoT-objects. In: *7th Workshop on the Reliability of Intelligent Environments (WORIE 2018)*, Rome, Italy [En Edición].

P. Pico-Valencia and J.A. Holgado-Terriza. An Agent of Thing Model for IoT-objects Driven by If-This Then-That Recipes. *IEEE Internet Computing Journal*, 2018. [En Revisión, Primera ronda].

Pablo A. Pico-Valencia and Juan A. Holgado-Terriza, *ADELE: A Middleware for Supporting the Evolution of Multi-agents Systems Based on a Metaprogramming Approach*, In: Trends in Practical Applications of Scalable Multi-Agent Systems, Seville, Spain, 2016, pp. 297-310.

Capítulo 6

Método sistemático para agentificar el IoT

“La ciencia se compone de errores, que a su vez son los pasos hacia la verdad”—**Jules Verne**.

En el campo de la Ingeniería de Software Orientada a Agentes se han propuesto varias herramientas enfocadas en llevar a cabo tareas de análisis, diseño, desarrollo, implementación, prueba y documentación de los sistemas multiagentes. Existen actualmente más de una decena de Metodología Orientadas a Agentes entre las que sobresalen: *MAS-CommonKads*, *Gaia*, *Tropos*, *Prometheus*, *Passi*, *Adelfe*, *RAP*, *Message*, *Ingenias*, *Desire* y *MaSE* [5, 240, 261]. No obstante, la mayoría de ellas, únicamente soportan dominios específicos y consecuentemente, ninguna toma en cuenta el modelado de aspectos importantes que gestionan comúnmente los ecosistemas de IoA (e.g. contexto, círculos sociales y colaboración, inteligencia, soporte de ecosistemas de servicios, artefactos de agentes, y objetos y recursos asociados a redes de IoT). Basado en estas limitaciones, y haciendo hincapié en la aserción de Tran y Low [261] respecto a que la mejor metodología tiene dependencia directa con el tipo de aplicación a desarrollar, es una necesidad contar con herramientas de Ingeniería de Software que guíe el proceso de análisis, diseño y desarrollo de aplicaciones que adopten el paradigma del IoA.

En este capítulo se describe un método sistemático basado en las directrices de las metodologías ágiles para el desarrollo de software. Este método permite modelar ecosistemas de IoA dirigidos por entidades del tipo LOA (*Linked Open Agent*) desde una perspectiva microscópica (inteligencia a nivel de agente individual) y macroscópica (inteligencia a nivel de ecosistema de agentes) [210].

6.1. Visión general del método

Las metodologías ágiles de desarrollo de software proponen básicamente un modelo basado en un proceso iterativo e incremental; es decir, que dicho proceso inicia con la especificación de requerimientos, su formalización, desarrollo e integración, hasta alcanzar el objetivo planteado. No obstante, es posible introducir nuevos requerimientos si estos son demandados por los usuarios en un momento específico del proceso de desarrollo, posterior a la implementación y puesta en marcha del sistema [236].

En torno a escenarios complejos y altamente dinámicos, característicos de escenarios de IoT e IoA, la demanda de procesos de adaptación, evolución, desempeño y calidad de servicio son bastante comunes debido a que los procesos de desarrollo de software inician aunque no se tengan objetivos claros al 100%. De allí, la necesidad de que se considere el proceso de modelado a nivel de componente específico (i.e. agente software, objeto de IoT) y a nivel de interacción (i.e. comunicación, colaboración). Por estos motivos, es recomendable usar componentes de software atómicos (i.e. microservicios) y metodologías de desarrollo ágil que permitan llevar a cabo su integración cuando se lo requiera. Sobre este trasfondo, se propone un método formal para desarrollar aplicaciones de IoA.

6.1.1. Método sistemático

El método propuesto constituye un método ascendente (*bottom-up*) que inicia definiendo una colección de requisitos globales que complementan una red de IoT a fin de escalar dicha red a una de red de IoA. Así, dicha red puede alcanzar un grado superior de interoperabilidad e inteligencia, no inherente en redes genéricas de IoT.

Los requerimientos son definidos en una fase preliminar de licitación de requerimientos del sistema por parte de un equipo especializado de trabajo a partir de las necesidades de los diferentes implicados (i.e., usuarios finales, clientes, administradores) en el sistema a modelar. Luego de definir los requerimientos, un conjunto de tareas son llevadas a cabo para modelar los aspectos asociados a cada agente de forma individual (fase microscópica) y por otro lado, modelar también aspectos relacionados con la infraestructura o red (fase macroscópica) [210], como detalla la Figura 6.1.

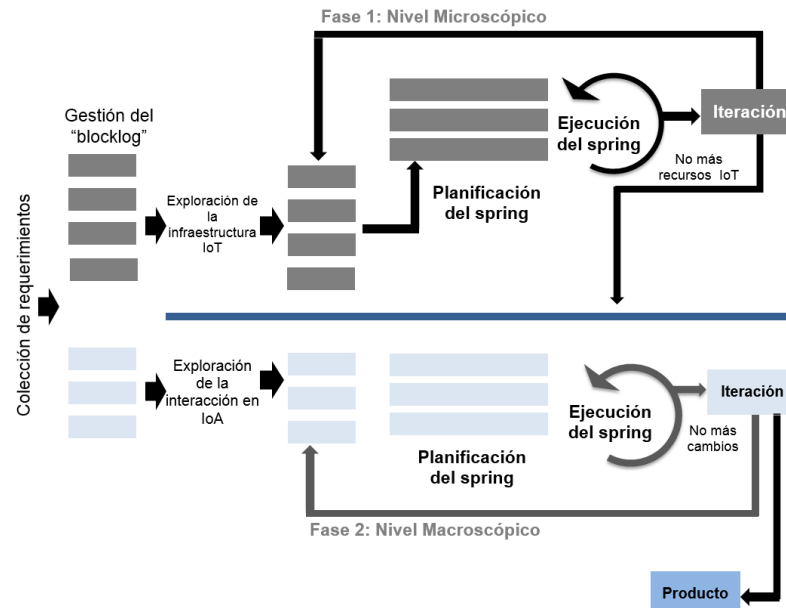


FIGURA 6.1: Esquema general del método propuesto.

De manera general, la fase microscópica, ilustrada en la parte superior de la Figura 6.1, lleva a cabo un proceso de exploración de la red de IoT; es decir, identificar cada uno de los objetos compatibles con las interfaces de IoT conectados a la red que se desea escalar hacia el IoA. Por cada uno de los objetos identificados, se ejecuta una iteración que permite convertir un objeto genérico de IoT a entidad de tipo LOA. No obstante, es importante señalar que no todos los objetos requieren una entidad de este tipo. Por ejemplo, en el caso de simples sensores o actuadores, éstos pueden seguir operando como objetos tontos que operan como simple recolectores de datos. Mientras que en el caso de objetos que pueden operar de manera semi-inteligente (i.e., persianas, ventanas) o inteligente (i.e., sistema de calefacción e iluminación basado en política ambiental y de ahorro energético), es recomendable vincular un agente a dicho objeto (*Agent of Thing*).

Por otro lado, la fase macroscópica, ilustrada en la parte inferior de la Figura 6.1, se ejecuta por todos los agentes que formen parte del ecosistema de IoA, y no solo por los objetos de IoT modelados en la fase anterior. Esta fase, parte de las entidades de tipo LOA modeladas a nivel

microscópico para integrarlos en una única red. Por tanto, esta fase se enfoca principalmente en ejecutar las tareas almacenadas en el bloque de registro, generadas a partir de la exploración de requisitos a nivel de infraestructura tales como: comunicación de agentes, círculos sociales, acciones colaborativas y acciones de inteligencia colectiva. Consecuentemente, agentes del tipo *Agent of Coordination*, *Agent of Data* y *Agent of People* son también modelados.

6.1.2. Proceso

El proceso lógico del método propuesto, ilustrado en la Figura 6.2, se describe en término de fases, etapas y pasos que se ejecutan de manera iterativa por cada objeto de IoT y por cada agente software que forma parte de un ecosistema de IoT escalado a IoA.

El proceso definido inicia con el mapa general de la red de IoT, que constituye el principal insumo para ejecutar la primera fase del proceso. En este mapa, se identifica esencialmente los diferentes componentes físicos tales como objetos inteligentes fijos y móviles, sensores y actuadores fijos y móviles, nodos pasarelas, servidores especializados y enrutadores, que se encuentran conectados físicamente a la plataforma física de IoT. De esta manera, se dispone un panorama claro respecto a las necesidades de procesamiento, coordinación, comunicación, inteligencia y colaboración que han sido introducidos previamente o requieren ser añadidos en la red para que su comportamiento alcance un mayor nivel de autonomía, interoperabilidad e inteligencia. Además, se dispone de características técnicas tales como: direcciones IP, protocolos de comunicación utilizados, nombre de estaciones, puertos, tipo de conexión (i.e., cableada, inalámbrica), servicios y aplicaciones soportadas, recursos de computación (i.e., capacidad de procesamiento, memoria, almacenamiento) y otros adicionales (i.e., batería).

Una vez que se disponga de la información descriptiva de la red física de IoT, es factible iniciar el proceso. La fase 1, Modelado a Nivel Microscópico (MI), se compone de seis etapas (desde MI-1 hasta MI-6); mientras que la fase 2, Modelado a Nivel Macroscópico (MA) la conforman siete etapas (desde MA-7 hasta MA-11). En ambas fases, cada una de las etapas son caracterizadas como: semiautomáticas (MI-2, MI-3 y MI-5), manuales (MI-1, MI-4), automáticas (MA-6, MA-8, MA-9, MA-10 y MA-11) e híbridas (MA-7).

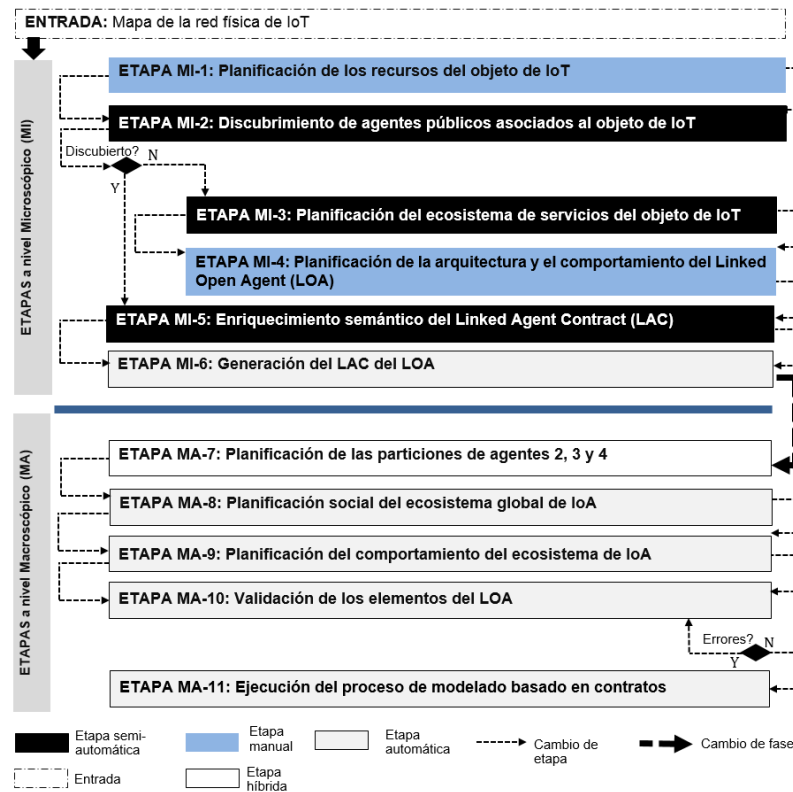


FIGURA 6.2: Visión general del proceso seguido por el método.

En el caso de las etapas semiautomáticas, se sugiere que el equipo de desarrollo emplee herramientas de software que permitan llevar a cabo el proceso de descubrimiento de servicios y agentes públicos disponible en la web. Por otro lado, respecto a las fases de naturaleza manual, el equipo de desarrollo es responsable de identificar los descriptores semánticos relevantes para describir los ecosistemas de servicios (IoS), de las cosas (IoT) y de los agentes (IoA) implicados en el modelado de la aplicación en cuestión. De igual manera, en el caso de las etapas automáticas, éstas son soportadas por herramientas de software especializadas en llevar a cabo la creación, gestión y validación de LACs, y la creación automática de LOAs. Finalmente, en lo que respecta a las etapas de tipo híbrido, éstas se ejecutan a partir de la combinación de dos o más etapas de las anteriormente descritas.

Es importante señalar que en caso de presentarse nuevos requerimientos no definidos al momento de realizar la especificación de requisitos en el registro de tareas, dichos requerimientos pueden introducirse cuidando que no se presente posibles conflictos con las entidades ya modeladas. Su introducción se debe llevar a cabo de forma iterativa, como se realizó previamente.

6.2. Visión microscópica

Como se ha detallado previamente, esta fase se lleva a cabo de manera iterativa por cada uno de los objetos que forma parte del ecosistema de IoT que lo requieren. Esto implica que se debe dar tratamiento a objetos semi-inteligente e inteligentes. No siendo relevante dar tratamiento a objetos que operan de manera no inteligente que solo capturan datos (i.e. sensores de temperatura, alarmas).

En resumen, en esta fase se convierte a los objetos que deben operar de forma semi-inteligente e inteligente en una entidad del tipo *Agent of Thing* [186] que vricula directamente un agente software al objeto. El proceso de modelado de dicha entidad se limita en esta fase únicamente en definir los artefactos de agentes—tareas y comportamientos—que dependen exclusivamente del agente y su correspondiente contrato semántico, sin tomar en cuenta los aspectos de interacción en ecosistemas globales.

6.2.1. Planificación de los recursos del objeto de IoT (MI-1)

La etapa MI-1 Planificación de recursos asociados al objeto de IoT, ilustrada en la Figura 6.3, constituye un proceso manual que el equipo de desarrollo debe llevar a cabo por cada uno de los objetos conectados a la red de IoT que será modelada como una infraestructura de IoA. Esta etapa consta de tres pasos y su principal objetivo está orientado a recopilar información relacionada con las capacidades y recursos asociados con el objeto de IoT sujeto a estudio en la iteración actual. Entre los datos más relevantes a rescatar en esta fase figuran los siguientes: **(i)** tipo de nodo según su movilidad (móvil o fija), **(ii)** tipo de nodo según su nivel de privacidad (público o privado), **(iii)** contexto donde opera, **(iv)** protocolo de comunicación empleado y **(v)** redes externas con las que el nodo se

comunica. Con esta información, se lleva a cabo el proceso de conversión de objetos IoT a objetos que tienen vinculado un agente tipo LOA; es decir, una entidad del tipo *Agent of Thing*. Así, dichos objetos son habilitados para modelar comportamientos inteligentes y cooperativos, los cuales son aspectos fundamentales en ecosistemas de IoA.

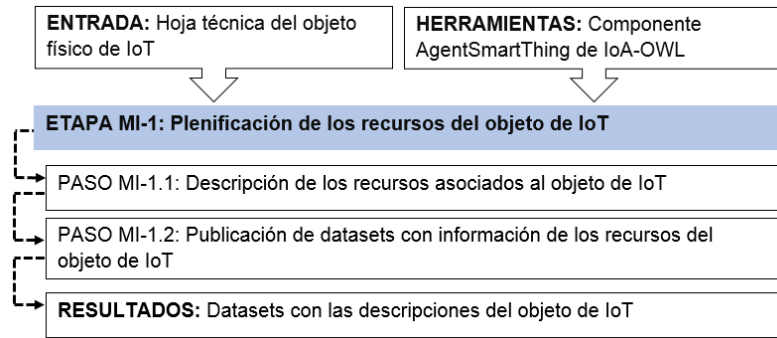


FIGURA 6.3: Flujo, insumos y resultados de la etapa MI-1.

6.2.1.1. Descripción de los recursos asociados al objeto de IoT (MI-1.1)

En este paso, se lleva a cabo un proceso de recopilación de información que satisfaga los conceptos integrados en el componente que forma parte de la ontología IoA-OWL, denominado *AgentSmartThing*. Entre los principales descriptores que deben ser definidos por el objeto modelado en esta iteración, figuran los siguientes: *identificador* (id_i), *nombre lógico* (ln_i), *dato generado* (dp_i) y *marca* (b_i) del objeto. Adicionalmente, deben ser definidas algunas referencias a entidades externas del tipo: *localización geográfica* (G_i), *ubicación con referencia a otros objetos* (L_i), *tiempo* (T_i), y un conjunto de recursos (R_i) lógicos (i.e., aplicaciones, agentes, servicios) y físicos (i.e., memoria, batería).

6.2.1.2. Publicación de datasets con información de los recursos de objetos de IoT (MI-1.2)

En este paso se realiza la publicación de la información relativa a los recursos (públicos y privados) del objeto de IoT sujeto a estudio en esta

iteración y que fueron definidos en el paso anterior (MI-1.1). Una vez concluido este paso se provee de una colección de datos que describen semánticamente un conjunto de objetos de IoT junto con sus recursos. Dichos datos (descriptores) podrán ser reutilizados al momento de la elaboración del contrato de los agentes. La persistencia de estos datos, así como los datos generados en etapas posteriores, deben ser gestionados por la infraestructura central de IoA (*IoA Central Infrastructure*) a través del repositorio de datos de IoA (*Repository of IoA Datasets*). Este repositorio debe implementarse utilizando una base de datos documental basada en los principios REST [259] para acceder a los elementos a través de invocaciones REST, al igual que se accede a los datos vinculados.

6.2.2. Descubrimiento de agentes públicos especializados asociados al objeto de IoT (MI-2)

Esta etapa, ilustrada en la Figura 6.4, consta de tres pasos y se centra en descubrir uno o más agentes que satisfagan parcial o totalmente las tareas que el objeto específico de IoT debe ejecutar para lograr sus objetivos. El proceso de descubrimiento puede ser realizado manualmente por los desarrolladores cuando tienen la necesidad de definir agentes específicos. Entonces, si existe tal restricción, esta etapa debe continuar paso a paso hasta que se complete la etapa y todos los agentes recuperados se definen como contrapartes del LOA modelado a través de su WAC. Por el contrario, si el sistema modelado no incluye ninguna restricción con respecto a los agentes contrapartes con los que el LOA puede establecer cooperación, se continúa directamente a la siguiente etapa (MI-4). En este caso, el proceso de descubrimiento de agentes es realizado por el propio LOA en tiempo de ejecución y el WAC del agente se completa gradualmente a medida que se descubran agentes contrapartes que lo satisfagan.

Esta etapa comienza con los requisitos funcionales asociados con el objeto de IoT en cuestión y debe abordar dos situaciones. La primera de ellas tiene por objeto descubrir a los agentes públicos que adoptan el modelo LOA para cumplir uno o más requisitos funcionales asociados con el objeto de IoT. Esto implica que los agentes descubiertos tendrán los descriptores semánticos empleados por el modelo IoA-OWL, lo que les

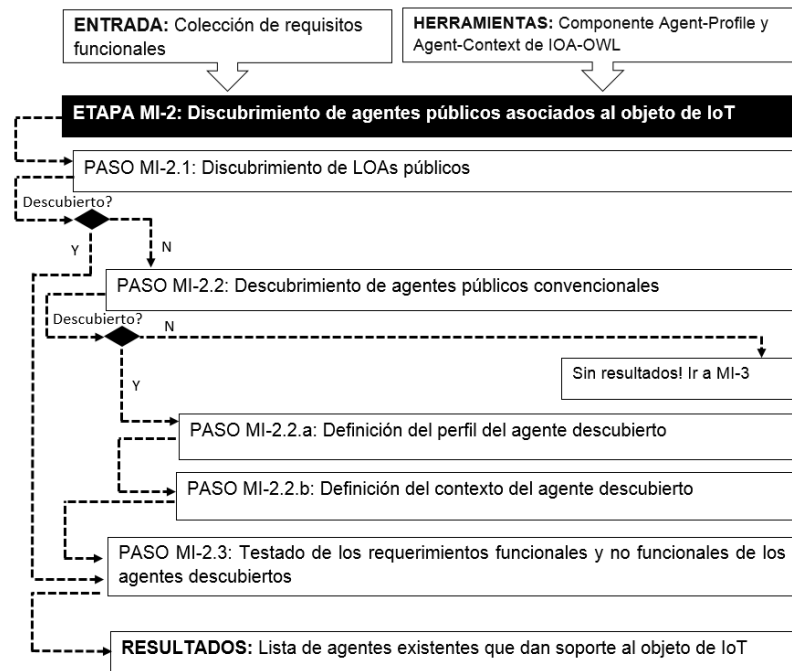


FIGURA 6.4: Flujo, insumos y resultados de la etapa MI-2.

da la capacidad de descubrir otros agentes y también de realizar procesos automatizados basados en sus descriptores semánticos. Sin embargo, también se pueden descubrir agentes externos convencionales y públicos en la segunda situación. En este caso, es necesario añadir los descriptores semánticos tanto del *AgentProfile* como del *AgentContext* al agente externo. Es importante tener en cuenta que ambos procesos pueden no tener éxito; en este caso particular, se debe seguir directamente al paso MI-3.

6.2.2.1. Descubrimiento de LOAs públicos (MI-2.1)

Este paso se enfoca en descubrir agentes software que adopten el modelo LOA para integrarlos a la aplicación modelada, coadyuvando así a resolver total o parcialmente los requisitos funcionales que el objeto de IoT debe satisfacer. Para cada agente descubierto, se debe identificar la URI (*Uniform Resource Identifier*) del contrato general asociado al

agente ya que éste guarda tanto la URI del componente *AgentProfile* como la URI del componente *AgentContext*. Es importante resaltar que si se descubrieron LOAs en este paso, el proceso debe continuar con el paso MI-2.3; de lo contrario se debe continuar secuencialmente con el paso MI-2.2.

6.2.2.2. Descubrimiento de agentes públicos convencionales (MI-2.2)

Este paso se recomienda aplicarlo únicamente cuando se descubren agentes convencionales especializados en proveer datos que resultarían muy caros de desarrollar por nuestra cuenta (i.e. búsqueda de vuelos, bolsa, información de climatología, entre otros). No obstante, hoy en día, gracias a la Computación en la Nube, muchas de estas funcionalidades son proveídas a través de servicios web. Si este es el caso, se recomienda agregar esta función al ecosistema IoS y luego procesarlos como se detalla en el paso MI-3.

- **Definición del perfil del agente descubierto (MI-2.2.a).** Para describir la información asociada al perfil de un agente, se recomienda emplear los conceptos integrados en el componente *AgentProfile*. Los principales descriptores necesarios para determinar este componente son los siguientes: *identificador* (id_i), *nombre* (n_i), *descripción* (d_i), *estado* (ds_i), *versión* (v_i), *fecha de presentación* (ds_i), *fecha de publicación* (dp_i), *solicitudes* (ds_i) r_i), y *solicitudes cumplidas* (ar_i). Además, es necesario incluir referencias a entidades externas, como el modelo para crear el agente (*createdFrom* (B_i)), propietario (*hasOwner* (O_i)), credenciales (*hasCertificate* (C_i)), y las restricciones en tiempo real (*hasRTrestriction* (RT_i)) que el agente debe satisfacer.
- **Definición del contexto del agente descubierto (MI-2.2.b).** Para describir la información contextual donde el agente opera, se recomienda emplear los conceptos del componente *AgentContext*. Los datos contextuales necesarios se centran en la definición de los siguientes aspectos: *identificador* (id_i), *nombre* (n_i) y breve *descripción* (d_i) del contexto. Además, deben incluirse referencias a entidades, tales como: localización geográfica (G_i), ubicación respecto a objetos (L_i), tiempo (T_i), nodo (N_i) y plataforma (P_i) donde

se ejecuta el agente. Esta última información permite a cualquier LOA conocer los parámetros técnicos para establecer comunicación con sus agentes homólogos y adicionalmente, les permite completar su WAC.

6.2.2.3. Testado de los requerimientos funcionales y no funcionales de los agentes descubiertos (MI-2.3)

En el área de la Ingeniería de Software, el proceso de testear componentes de software permite detectar errores o discrepancias entre el resultado proporcionado y el resultado esperado que un artefacto debe proporcionar [250]. Con el fin de evitar la reutilización de agentes que incorporen errores en términos de resultados inconsistentes o comportamientos inadecuados, este paso recomienda aplicar una prueba unitaria a cada agente descubierto en esta etapa y examinarlos individualmente y de forma aislada [131, 250]. Se recomienda evaluar aspectos como la capacidad de respuesta, capacidad de manejo de excepciones, comunicación FIPA, calidad de servicio y calidad de datos. Sin embargo, los problemas no funcionales como la fiabilidad, usabilidad y la robustez del agente también pueden ser inspeccionados. Además, se sugiere que únicamente los agentes que superen este proceso de prueba de manera integral; es decir, agentes sin fallos del tipo muy graves, graves y moderados, deben ser considerados como parte de la aplicación modelada.

Al finalizar, este paso proporcionará un conjunto de agentes fiables que siguen el modelo LOA, útiles en el modelado del comportamiento del agente vinculado directamente al objeto de IoT procesado en esta iteración (MI-4). Sin embargo, si todas las tareas requeridas no son resueltas por los agentes recuperados, entonces se debe seguir a la siguiente etapa (MI-3).

6.2.3. Planificación de ecosistema de servicios del objeto de IoT (MI-3)

Esta etapa, mostrada en la Figura 6.5, se lleva a cabo de forma semi-automática. Su misión está enfocada en intentar descubrir los microservicios [260] que han sido empleados o son requeridos por el objeto para ejecutar sus operaciones. Así, se llega a definir una red de microservicios heterogéneos que satisfagan cada tarea que el objeto modelado en esta

iteración debe ejecutar. Para llevar a cabo este proceso de descubrimiento se recomienda utilizar herramientas especializadas en la exploración de servicios públicos tanto en ecosistemas de IoS implementados mediante SOA [87] y/o ROA [115].

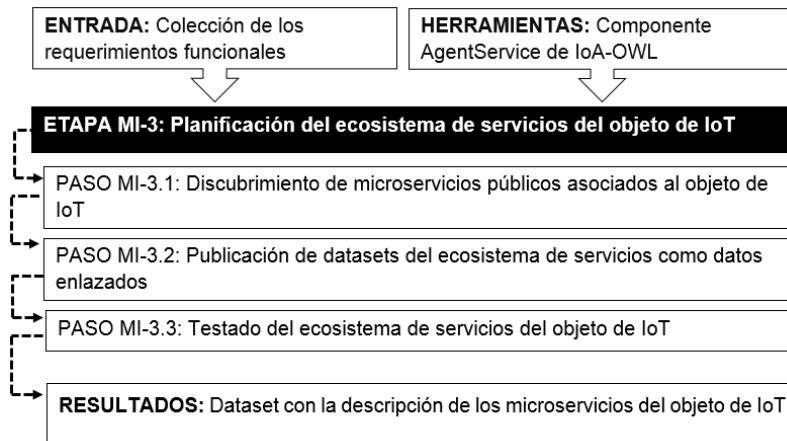


FIGURA 6.5: Flujo, insumos y resultados de la etapa MI-3.

El soporte de SOA y ROA nos asegura de que la filosofía de la Computación Orientada a Servicios apoye a la reutilización de componentes distribuidos. De esta manera, algunos de los componentes públicos que son útiles para la aplicación modelada pueden ser integrados sin la necesidad de hacer esfuerzos adicionales o incurrir en costos extras.

6.2.3.1. Descubrimiento de microservicios públicos asociados al objeto de IoT (MI-3.1)

En este paso, se deben emplear herramientas de software para descubrir microservicios públicos relacionados con el dominio de la aplicación modelada tanto en ecosistemas IoS pertenecientes al dominio IoA como en ecosistemas IoS convencionales. Las herramientas empleadas dependerán del dominio de la aplicación modelada. No obstante, se recomienda el uso de herramientas que garanticen invocaciones seguras de servicios y que éstos provean calidad de servicio en los procesos, para así, garantizar la confiabilidad y consistencia de los datos ofrecidos por la aplicación modelada.

Los servicios recuperados deben ser enriquecidos semánticamente empleando el componente *AgentService* debido a que no guardan compatibilidad con la descripción semántica usada en el modelo de servicios gestionado en los ecosistemas de servicios de IoA. Entre los principales descriptores a añadir deben figurar los siguientes: URI asociada con el documento de descripción del servicio (*URI del contrato* (u_i)), restricciones en tiempo real (*hasRTrestriction* (RT_i)), perfil del servicio (Pr_i), parámetros de entrada (IP_i), parámetro de salida (OP_i), operación del servicio (O_i) y recursos (R_i) si se trata de un servicio RESTful.

Es importante señalar que partimos de la premisa de que todos los servicios están operativos, pero si no es así, entonces los servicios que faltan deben ser construidos y enriquecidos semánticamente, como se detalla en este paso. Posterior a ello, se debe continuar al paso MI-3.2.

6.2.3.2. Publicación de datasets del ecosistema de servicios como datos enlazados (MI-3.2)

Este paso toma las instancias de la descripción semántica de cada servicio llevada a cabo en el paso MI-3.1 para publicar los descriptores como datos abiertos enlazados. Cuando se concluye este paso, la etapa proporciona una colección de *datasets* públicos y privados que describen el ecosistema de servicios del objeto modelado. Sólo posterior a la publicación de estos descriptores, se puede realizar la composición del contrato del agente.

6.2.3.3. Testado del ecosistema de servicios del objeto de IoT (MI-3.3)

De manera similar al proceso detallado en el paso MI-2.3 donde los agentes fueron testados, en este paso se debe aplicar la misma mecánica, pero ahora enfocado a probar las acciones del tipo petición/respuesta de cada uno de los microservicios que el objeto de IoT y su agente vinculado gestionan.

6.2.4. Planificación de la arquitectura y el comportamiento del LOA (MI-4)

Esta etapa se enfoca en definir los componentes del LOA de acuerdo al rol que desempeña en la red de IoA, y organizarlos de una manera lógica para modelar un comportamiento más inteligente, a nivel de entidad individual. Para ello, es necesario contar con el mapa de la red de IoT y el mapa del ecosistema de IoS asociado al objeto de IoT para el que se está modelando el agente.

Los cuatro pasos MI-4.1, MI-4.2, MI-4.3, MI-4.4 y MI-4.7, ilustrados en la Figura 6.6, representan pasos independientes del modelo de agente adoptado por el LOA; mientras que los pasos MI-4.5 y MI-4.6 tienen dependencia con respecto al modelo de agente empleado para construir el LOA. Sin embargo, este método sólo enfatiza en LOAs reactivos.

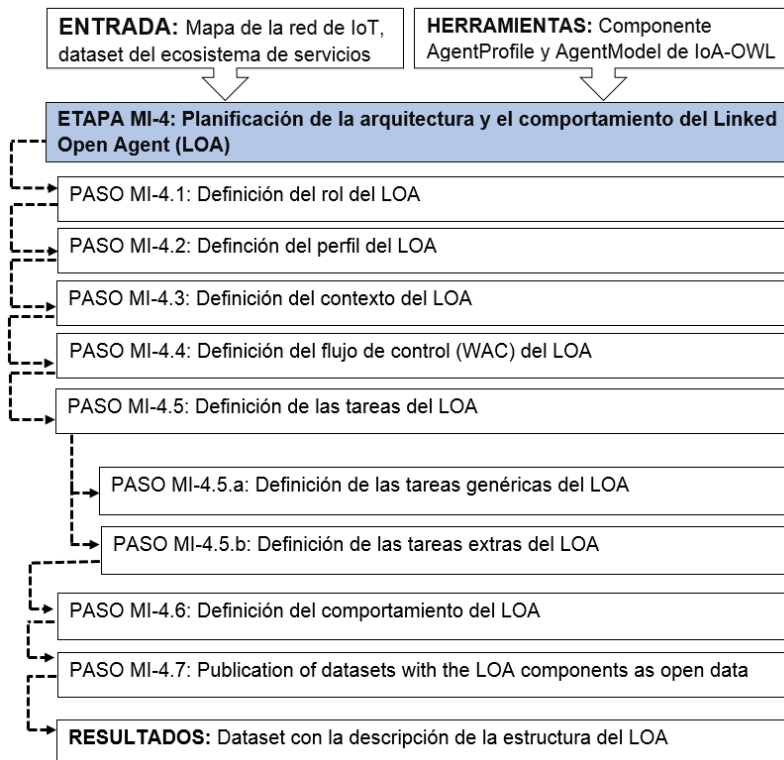


FIGURA 6.6: Flujo, insumos y resultados de la etapa MI-4.

6.2.4.1. Definición del rol del LOA (MI-4.1)

Antes de decidir qué modelo utilizar para construir el LOA, es importante determinar su función con el fin de definir los principales objetivos y tareas que debe realizar en tiempo de ejecución. En este punto, todos los agentes modelados tienen un objeto IoT vinculado y, por lo tanto, el rol de todos los agentes a nivel microscópico es *Agent of Thing*.

6.2.4.2. Definición del perfil del LOA (MI-4.2)

En este punto, el agente a modelar será un nuevo LOA, y por lo tanto se requiere añadir los descriptores semánticos correspondientes para completar el perfil del agente (*AgentProfile*). Para añadir esta información es necesario seguir el mismo proceso descrito anteriormente en el paso MI-2.2.a.

6.2.4.3. Definición del contexto del LOA (MI-4.3)

En este paso, también se requiere agregar los descriptores semánticos correspondientes para completar el contexto de operación del agente (*AgentContext*). Para hacer esto, es necesario seguir el mismo proceso descrito anteriormente en el paso MI-2.2.b.

6.2.4.4. Definición del flujo de control (WAC) del LOA (MI-4.4)

En este paso, se evalúa la naturaleza del objeto de IoT que se va a agentificar para determinar las acciones de control adecuadas que éste debe realizar sobre el ecosistema de IoT.

Las acciones que el LOA debe realizar en el ecosistema IoA se especifican usando un WAC. Este elemento sigue el esquema y la sintaxis descritos en el fragmento de código 5.3. En términos generales, cada WAC está compuesto por un conjunto de canales que, cuando se ejecutan, permiten la comunicación con los objetos y agentes externos de IoT correspondientes y, en consecuencia, se alcanzan los objetivos del LOA de forma colaborativa.

Para la definición del contenido de un WAC no es necesario especificar los metadatos de red que permiten la comunicación con objetos específicos de IoT (i.e. líneas 11-13 y líneas 23-25 en el fragmento de código 5.3). Estos metadatos son descubiertos automáticamente por el LOA mediante

procesos de descubrimiento semántico. Los únicos metadatos obligatorios a definir son los relativos a los datos de interés para cada canal (línea 8.9 del fragmento de código 5.3). El descubrimiento de los objetos de IoT y agentes se lleva a cabo en función de ellos. Por ello, es importante que la lógica de control (reglas) de cada WAC se defina correctamente. De la lógica modelada por los WACs depende qué agentes contrapartes serán descubiertos para controlar el ecosistema de IoA. Un componente de la herramienta *IoA-Building Tool* enfocada a crear la lógica de este elemento ha sido creada. En el Anexo C se muestra la interfaz principal de dicha herramienta.

6.2.4.5. Definición de las tareas del LOA (MI-4.5)

En la introducción de esta etapa indicamos que dos pasos (MI-4.4 y MI-4.5) tenían dependencia con el modelo de agente seleccionado para construir el LOA. Este es el primero de estos pasos y se enfoca en determinar un conjunto de tareas de agente que abordan el comportamiento del LOA.

En términos generales, este paso propone el uso de tareas de agente para estructurar el comportamiento de un LOA reactivo. Siguiendo las especificaciones descritas en el marco de trabajo Framework-IoA (Capítulo 5) se recomienda realizar lo siguiente:

- **Definición de las tareas genéricas del LOA (MI-4.5-a).** Un LOA define estratégicamente cuatro tareas de agentes tales como: *Asynchronous Agent Response Task*, *Injection Task*, *Workflow Execution Task* y *Agent Discovery Task*. Estas tareas son genéricas para cualquier LOA y han sido definidas usando *InternalBehaviors*. Estos comportamientos se integran dentro del agente en la etapa de desarrollo mediante la definición de descriptores semánticos, tales como nombre (in_i), descripción (id_i), restricciones en tiempo real (iRT_i), argumentos (iP_i) y el flujo de trabajo correspondiente (iW_i).
- **Definición de las tareas extras del LOA (MI-4.5-b).** El uso de tareas adicionales es técnicamente factible para un LOA. Un LOA tiene asociada la tarea denominada *Injection Task* que le permite integrar tareas adicionales si las funcionalidades genéricas no le

permiten alcanzar sus objetivos. Las tareas adicionales pueden ser descritas usando *ExternalBehavior*. Por lo tanto, pueden describirse con los mismos descriptores que *InternalBehavior*; pero incluyendo otros descriptores como *className* (ec_i) que define el nombre de clase del artefacto de agente y que el compilador *on-the-fly* usará como argumento para crear el ejecutable de la tarea específica. Es importante destacar que la integración de la tarea descrita se realiza en tiempo de ejecución mediante la aplicación de técnicas de metaprogramación como se realiza en la herramienta ADELE (*Agent Dynamic EvoLutionary at runtimE*) [208].

6.2.4.6. Definición del comportamiento del LOA (MI-4.6)

Este paso se centra en la integración y modelado del mecanismo empleado por el LOA para actuar de forma coherente y autónoma basado en el procesamiento de los descriptores semánticos almacenados en su LAC y la lógica de control definida en su WAC. Este es uno de los pasos más importantes, ya que es la primera aproximación en términos de modelado del comportamiento inteligente de los sistemas.

Las implicaciones de inteligencia de un LOA, a nivel de entidad, consisten básicamente en determinar cómo debe comportarse el LOA para lograr sus objetivos. Para ello, recomendamos modelar este aspecto introduciendo los siguientes mecanismos:

- Caracterización de las acciones de control que el agente debe realizar sobre el ecosistema del IoT. Es esencial definir la lógica de control que el LOA debe realizar sobre los recursos y objetos de IoT para que el proceso de control automático pueda ser ejecutado por el LOA, utilizando sus propios recursos o descubriendo los agentes contrapartes más adecuadas que le ayuden a alcanzar sus objetivos. En otras palabras, se requiere especificar el WAC del LOA.
- Programación de la tarea *Execution Workflows Task* del LOA. El flujo de trabajo (WAC) asociado con el LOA proporciona las directrices en las que debe basarse el LOA para ejecutar procesos automáticos en IoA. Sin embargo, este componente no puede ejecutarse hasta que el LOA no defina y programe temporalmente una

tarea específica llamada *Workflow for Agent Execution Task*, la cual forma parte de un LOA genérico. Luego, es necesario programar la frecuencia con la que el LOA debe realizar esta tarea.

- Planificación de la tarea *Agent Discovery Task* del LOA. Al igual que la planificación de la tarea anterior, también es necesario planificar una segunda tarea. Esta tarea se denomina *Agent Discovery Task* e indica al LOA con qué frecuencia debe solicitar a la infraestructura central *IoA Central Infrastructure* la ejecución de procesos de razonamiento semántico. Este tipo de proceso tiene como objetivo descubrir agentes contrapartes que colaboran con él para lograr sus objetivos.
- Delimitación de la tarea *Asynchronous Agent Response Tasks* del LOA. Esta tarea se ejecuta con una frecuencia específica y por lo tanto, no es una tarea que el LOA deba planificar. Sus acciones se ejecutan siempre que el LOA recibe un mensaje de otros LOAs externos. Por lo tanto, y para evitar comunicaciones fraudulentas que conduzcan a una violación de la integridad de la LOA, de los datos que maneja, de los objetos de IoT que controla y accede, y de los agentes contrapartes con los que se comunica, es necesario definir el dominio del vocabulario al que el agente debe dar respuesta. Dependiendo de ello, el LOA sabrá cómo reaccionar a los mensajes que reciba y las acciones de control ejecutadas sobre su entorno.

Para modelar los mecanismos reactivos que le permitan al LOA soportar un comportamiento inteligente a nivel microscópico, sugerimos el uso de un Diagrama de Máquina de Estado de Comportamiento UML. Este diagrama se utiliza para especificar el comportamiento discreto de una parte de un sistema diseñado mediante transiciones de estado finitas. [195]. Además, se puede utilizar un diagrama de secuencia UML para representar las tareas que ejecuta el LOA cuando se recibe un mensaje [195]. Todas las tareas definidas en el paso anterior (MI-4.4) deben ser integradas de manera lógica para componer el comportamiento del LOA reactivo. Al final, estos diagramas junto con el esquema genérico del WAC asociado al LOA, serán útiles para entender el mecanismo de activación de eventos que dirige cada LOA en el ecosistema de IoA.

6.2.4.7. Publicación de dataset con los componentes del LOA como datos abiertos enlazados (MI-4.7)

Los descriptores semánticos asociados con los artefactos del LOA definidos en el paso MI-4.5 deben publicarse siguiendo el mismo enfoque empleado para la publicación de datos anterior. Todos los descriptores publicados en este paso deben ser privados para los usuarios y públicos para el desarrollador de la aplicación, quien los puede emplear posteriormente para construir el agente.

6.2.5. Enriquecimiento semántico del LAC (MI-5)

Esta etapa, ilustrada en la Figura 6.7, es responsable de crear una instancia de cinco de los componentes integrados por la ontología IoA-OWL—*AgentProfile*, *AgentContext*, *AgentService*, *AgentModel*, *AgentSmartThing*—que el LOA administra. Para cumplir con los campos de estas instancias ontológicas se pueden utilizar tipos de datos primitivos (i.e., cadenas, numéricas, valores de verdad). Sin embargo, la mayoría de estos deben introducirse a través de URIs que enlazan con los descriptores semánticos publicados previamente como datos enlazados.

Para lograr esta etapa, recomendamos usar los descriptores obtenidos en cada una de las etapas anteriores de la siguiente manera: (MA-5.1) enriquecimiento del *AgentProfile* con los descriptores definidos en MI-2.2.a, (MA-5.2) enriquecimiento de *AgentContext* con descriptores definidos en MI-2.2.b, (MA-5.3) enriquecimiento de *AgentService* con descriptores definidos en MI-3.2, (MA-5.4) enriquecimiento de *AgentModel* con descriptores definidos en MI-4.5 y (MA-5.5) enriquecimiento de *AgentSmartThing* con los descriptores definidos en MI-1.1. Estos descriptores pueden ser introducidos a través de un módulo específico de la herramienta *IoA-Building Tool*. El esquema general de la interfaz gráfica de este módulo se muestra en el Anexo D.

Después de realizar el último paso, esta etapa proporciona como resultado una instancia de datos de la ontología IoA-OWL, con la excepción del componente *AgentSocial*, porque se enriquecerá en la etapa MA-8.

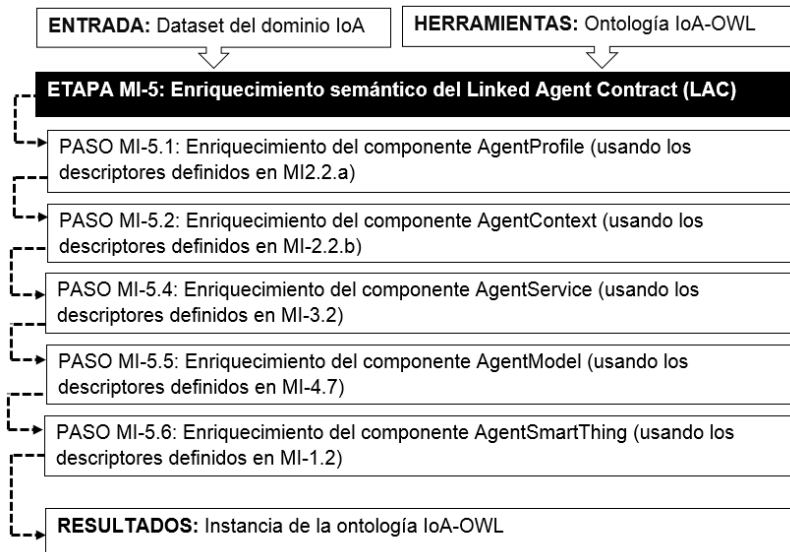


FIGURA 6.7: Flujo, insumos y resultados de la etapa MI-5.

6.2.6. Generación del LAC del LOA (MI-6)

En esta etapa se esboza el proceso de auto-generación del LAC y sus seis secciones contractuales según la secuencia ilustrada en la Figura 6.8. Los siete pasos ilustrados se ejecutan secuencialmente como sigue: (MI-6.1) auto-generación de la sección contractual *AgentProfile*, (MI-6.2) auto-generación de la sección contractual *AgentContext*, (MI-6.3) auto-generación de la sección contractual *AgentSocial*, (MI-6.4) auto-generación de la sección contractual *AgentService*, (MI-6.5) auto-generación de la sección contractual *AgentModel*, (MI-6.6) auto-generación de la sección contractual *AgentSmartThing* y (MI-6.7) auto-generación del contrato general de LOA.

Todos los pasos previamente especificados se realizan de forma automatizada utilizando una la herramienta *IoA Building Tool*. Por lo tanto, esta etapa no requiere que los desarrolladores hagan esfuerzos adicionales. Sin embargo, es importante señalar que el éxito del proceso de generación de contratos dependerá de la consistencia de los datos y las URIs introducidas en el proceso de enriquecimiento semántico realizado en la etapa MI-5. Es importante señalar que la auto-generación de la

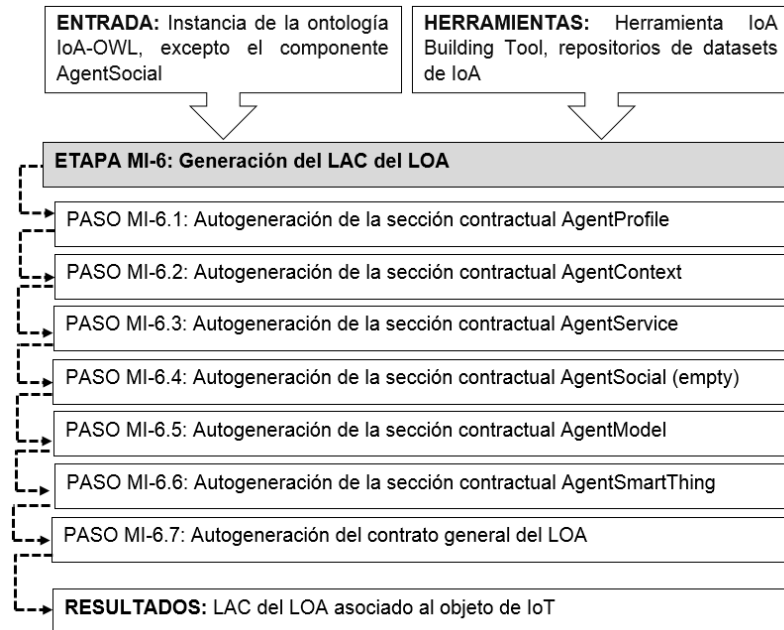


FIGURA 6.8: Flujo, insumos y resultados de la etapa MI-6.

sección contractual *AgentSocial* (MI-6.3) está vacía porque este aspecto se modela y enriquece semánticamente a nivel macroscópico.

6.3. Visión macroscópica

La segunda fase, ilustrada en la parte inferior de la Figura 6.1, se compone de cinco etapas. De manera general, una de las etapas constituye una etapa híbrida (MA-7) y las cuatro etapas restantes constituyen tareas automatizadas (MA-8, MA-9, MA-10, MA-11).

Esta fase básicamente tiene como objetivo modelar los aspectos relacionados con la coordinación de los LOAs *Agent of Things*, la creación de la infraestructura global y el modelado del comportamiento de los agentes, tales como: comunicación, redes y círculos sociales y procesos de colaboración.

6.3.1. Planificación de las particiones de agentes 1, 2 y 3 (MA-7)

En esta etapa, las instancias del tipo *Agents of Coordination* deben ser planificadas para generar sus LACs correspondientes de manera similar a nivel microscópico. Además, es necesario que este tipo de LOAs tengan asociado un WAC que incluya comunicaciones con los agentes de tipo *Agents of Things* que deben coordinar. Por lo tanto, para que esta etapa sea cumplida, las etapas MI-2 a MI-6 deben llevarse a cabo como en el caso de los LOAs de tipo *Agents of Things*; pero orientadas a cubrir tareas relacionadas con el establecimiento de procesos de comunicación y coordinación con los LOAs *Agents of Things*, *Agents of Data* y *Agents of People*.

6.3.2. Planificación social del ecosistema global (MA-8)

Esta etapa, ilustrada en la Figura 6.9, está esencialmente orientada a modelar el aspecto social entre los que se incluyen la interacción y colaboración asociada a cada uno de los LOAs previamente definidos. Esta etapa es completamente ejecutada por el LOA en tiempo de ejecución en base a su WAC y LAC, y mediante la ejecución de procesos de razonamiento solicitados por el LOA a la infraestructura *IoA Central Infrastructure*. La información recuperada por el LOA es el insumo para actualizar la sección contractual social del LAC que hasta ahora se encuentra vacía. Esta información se almacena siguiendo la especificación del componente *AgentSocial*.

En este paso el equipo de trabajo es libre de definir agentes contrapartes específico estáticos para cubrir las restricciones de seguridad. No obstante, el LOA también tiene la capacidad de descubrir automáticamente los agentes contrapartes adecuados en tiempo de ejecución basándose en el contenido de su LAC y WAC. De hecho esta es una de las potencialidades de los LOAs con respecto a los agentes genéricos.

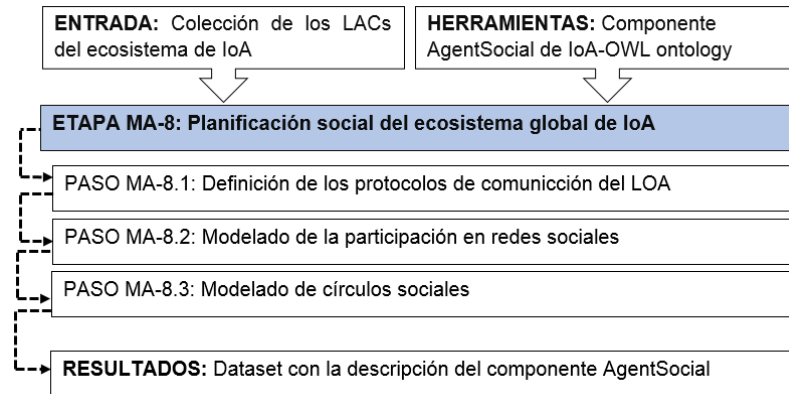


FIGURA 6.9: Flujo, insumos y resultados de la etapa MA-8.

6.3.2.1. Definición de los protocolos de comunicación del LOA (MA-8.1)

Algunas de las contribuciones más importantes que FIPA ha proporcionado a la comunidad de desarrolladores de agentes son: el lenguaje de comunicación de agentes (FIPA-ACL) y un conjunto de protocolos de comunicación del FIPA—*Request*, *Query*, *Propose*, *Subscribe*, y *Contract-Net*—a partir de los cuales es posible establecer interacción entre agentes de una manera estándar [95].

En este paso, para cada LOA definido, se debe definir el protocolo de comunicación para establecer el canal de comunicación entre los LOA específicos y los usuarios finales. Además, se debe definir el protocolo de comunicación social que los LOAs colaborativos usarán para establecer procesos de cooperación lógica y eficiente con los LOAs que comparten su círculo social. En este último caso, recomendamos el uso del protocolo *Contract-Net*, cuyo propósito es llevar a cabo el proceso de colaboración entre agentes [271].

6.3.2.2. Modelado de la participación en redes sociales (MA-8.2)

Es importante definir si cada LOA debe operar de manera colaborativa o de manera individual. Este aspecto se determina a través del

descriptor de grado de colaboración (c_i). Si se trata de un LOA colaborativo, la información del LAC se almacena automáticamente en la infraestructura *IoA Central Infrastructure* para que los LOAs seas descubiertos por otros LOAs interesados en ecosistema de IoA.

Genéricamente, un LOA se integra en una red social genérica llamada IoA. Sin embargo, en el caso de que un LOA se integre en una red diferente, se debe describir una nueva red social. Para describir las redes sociales (SN_i) en las que participa un LOA, es necesario introducir el descriptor de la red social *nombre* (n_i). Otros descriptores, como *densidad* (d_i), *matriz de adyacencia* (M_i), y las medidas de centralidad—*InDegree* (I_i), *OutDegree* (O_i), *Degree* (D_i), *Betweenness* (B_i), *Closeness* (C_i)—son inicializados automáticamente de acuerdo a las relaciones y círculos sociales que vayan siendo definidos gradualmente.

El modelado de redes sociales no es una tarea trivial, por lo que sugerimos emplear un sociograma [77, 238] para facilitar la representación, análisis y gestión de los nodos dentro de cada grupo perteneciente al ecosistema de IoA.

6.3.2.3. Modelando los círculos sociales (MA-8.3)

Este paso es complementario al paso anterior (MA-8.2) y se centra en la introducción de los principales círculos sociales en los que los LOAs participan. Este proceso se puede realizar según dos casos. El primero incluye LOAs que los desarrolladores tienen que definir sus círculos sociales manualmente. Por el contrario, el segundo incluye LOAs que pueden descubrir automáticamente a otros LOAs. Por lo tanto, el segundo caso es un paso totalmente automático abordado por la herramienta *IoA-Building Tool*.

Independientemente de si este proceso es manual o automático, para describir las relaciones de los LOAs (R_i) es necesario definir los siguientes descriptores: *nombre* (n_i), *descripción*. (dx_i), *weight* (w_i), *date of creation* (d_i), *source* ($From_i$), *destino* (To_i), y la ruta de acceso (P_i) de la relación (R_i). Además, se deben verificar algunas restricciones para crear únicamente relaciones consistentes: **(i)** una relación con un LOA externo sólo puede crearse si ambos, la fuente y el destino guardan como mínimo un tema de interés (o datos de interés), **(ii)** una relación puede crearse

sólo si el origen del LOA forma parte de una red social en la que participa el destino del LOA, y (iii) no crear relaciones reflexivas pues no contribuyen al modelo social.

Como recomendación, las relaciones deben introducirse utilizando el sociograma construido en el paso anterior. De esta manera, es posible analizar todos los grupos en los que cada LOA opera así como los contrapartes con los que se comunica y colabora.

6.3.3. Planificación del comportamiento del ecosistema de IoA (MA-9)

El esquema general de esta etapa es ilustrado en la Figura 6.10. Su objetivo es coordinar el comportamiento individual asociado a cada LOA definido en el paso MI-4, y extenderlo hacia un comportamiento global. Así, el comportamiento modelado por los LOAs puede cubrir los requisitos de la infraestructura de IoA, tales como la interacción y la colaboración con homólogos externos.

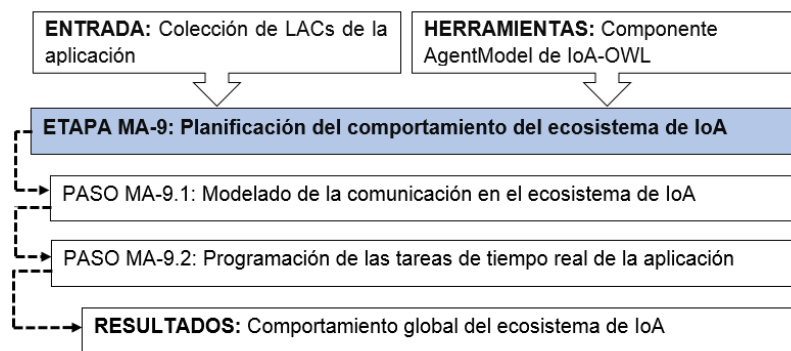


FIGURA 6.10: Flujo, insumos y resultados de la etapa MA-9.

Los principales mecanismos enfocados en modelar el comportamiento de un LOA en el ecosistema de IoA enfatizan en los siguientes aspectos:

- Descubrimiento de los LOAs contrapartes. Una de las capacidades de un LOA es que emplea los LOAs contrapartes más adecuados que operan en el ecosistema del IoA. Para ello, cada LOA tiene la capacidad de solicitar la ejecución de procesos de razonamiento

semántico a la infraestructura *IoA Central Infrastructure*. Este proceso es automático y realiza un proceso de comparación a partir del conocimiento registrados en los LACs de los LOAs que forman el ecosistema IoA, y que son almacenados en formato tripletas RDF.

- Selección de los LOAs contrapartes adecuados. Una vez que la infraestructura *IoA Central Infrastructure* proporciona una lista de contrapartes de tipo LOA, el LOA en cuestión aplica un operador de selección basado en un criterio específico (i.e., restricciones en tiempo real, poder social) que permite filtrar y seleccionar los LOA más idóneos que ofrecen la funcionalidad al LOA que ha ejecutado el proceso de descubrimiento.
- Definición de la política de recuperación de fallos para objetos de IoT y agentes. El proceso de descubrimiento de contrapartes proporciona datos para actualizar el LAC y el WAC del LOA que hizo la solicitud de descubrimiento. Sin embargo, estos LOAs pueden estar sujetos a fallas una vez que hayan comenzado a operar. Para cubrir este aspecto, es necesario definir una política de recuperación ante fallos que objetos y agentes puedan presentar (i.e. descubrir nuevos LOAs para reemplazar los LOAs fallidos, ejecutar los LOAs de contraparte y usarlos mientras se recuperan los LOAs fallidos). El desarrollador tiene la opción de decidir qué política de recuperación de fallos usar para garantizar que el sistema mantenga la funcionalidad con los recursos disponibles.

6.3.3.1. Modelado de la comunicación en el ecosistema de IoA (MA-9.1)

Este paso se realiza con el fin de definir el modelo de comunicación de cada LOA a ser coordinado de acuerdo con el modelo social y los requerimientos a nivel de interacción general. Este es un proceso que es independiente de la arquitectura de agente adoptada para modelar el ecosistema de IoA. Por lo tanto, es aconsejable modelar el comportamiento global a través del uso del Diagrama de Comunicación UML. Este diagrama en específico muestra las interacciones entre objetos y/o partes usando mensajes secuenciados en un formato libre [195].

6.3.3.2. Programación de las tareas de tiempo real de la aplicación (MA-9.2)

En este paso se aplica una prueba de programación con el fin de determinar la factibilidad de que las tareas requeridas por el sistema puedan ser programadas. Para este cometido se recomienda emplear un planificador dinámico de tiempo real como puede ser: *Earliest Deadline First* (EDF), *Least Laxity First* (LLF) [244], y *Receding Horizon Control* (RHC) [253]. Las tareas que dirigen el comportamiento global de la aplicación pueden representarse utilizando un diagrama de secuencia UML [195] en el que deben incluirse las restricciones en tiempo real.

6.3.4. Validación de los elementos del LOA (MA-10)

Esta etapa, ilustrada en la Figura 6.11, está destinada a realizar el análisis a nivel microscópico de cada uno de los elementos (i.e., LACs, WACs y funciones de los LOAs) generados por la herramienta *IoA-Building Tool* en las etapas anteriores. De esta manera, es posible identificar y diagnosticar errores que podrían causar inconsistencias dentro del LAC y sus seis secciones contractuales. El proceso de validación se lleva a cabo teniendo en cuenta las siguientes cuestiones:

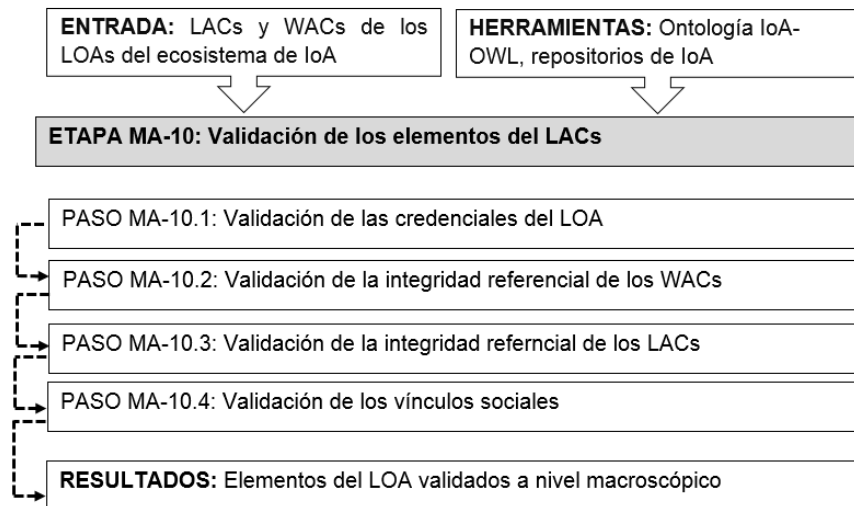


FIGURA 6.11: Flujo, insumos y resultados de la etapa MA-10.

6.3.4.1. Validación de las credenciales del LOA (MA-10.1)

Antes de la generación del LAC que describe cada LOA, se deben verificar los datos relativos al Identificador Universal de Agente (UIA) para validarlos y evitar la inclusión de LOAs peligrosos en el ecosistema de IoA. Cada UIA se caracteriza por ser un identificador único proporcionado por la infraestructura *IoA Central Infrastructure* tras realizar la solicitud correspondiente. Es importante señalar que la verificación de credenciales se realiza en ambos casos, antes y después de la creación de cada LOA, con el fin de poder auditar cuáles de los UIAs solicitados están autorizados para operar en el ecosistema de IoA.

6.3.4.2. Validación de la integridad referencial de los WACs (MA-10.2)

Las posibles inconsistencias que pueda tener un LAC impacta el contenido del WAC de un LOA. Por ello, es importante validar los datos del WAC de cada LOA para evitar que los LOAs dejen de operar debido a la presencia de datos inconsistentes en el contenido de su flujo de trabajo. Para cubrir este aspecto, cada LOA tiene un mecanismo que le permite adaptarse y recuperarse de las inconsistencias en su WAC (i.e., falta de disponibilidad de objetos de IoT y de agentes contrapartes).

6.3.4.3. Validación de los vínculos sociales (MA-10.3)

La razón principal por la que surgen inconsistencias en los LACs es porque uno o más URIs ingresadas en el proceso de enriquecimiento semántico han incluido una entidad referenciada que no se encuentra disponible. En consecuencia, es muy importante probar la disponibilidad de cada URI que forme parte del contenido de los LACs. Si hay inconsistencias, es posible que los mecanismos de razonamiento y descubrimiento de agentes se vean afectados y, por lo tanto, el sistema reduce su nivel de inteligencia y autonomía, y en el peor de los casos, el sistema puede llegar a fallar.

6.3.5. Ejecución del proceso de modelado basado en contratos (MA-11)

Esta etapa, ilustrada en la Figura 6.12, se centra en iniciar los principales recursos de cómputo de tipo lógico y físico para determinar los parámetros básicos relacionados con el ecosistema de IoA (i.e., configuraciones de servidores, hosts, puntos finales) necesarios para ejecutar el proceso de generación de código fuente y luego la ejecución del código fuente compilado en la infraestructura de IoA.

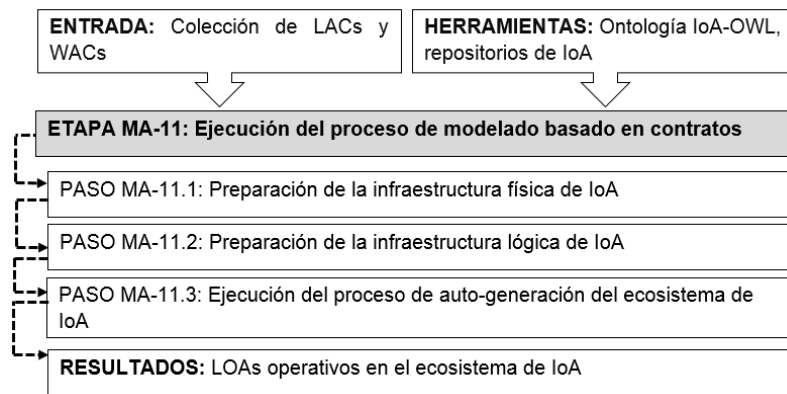


FIGURA 6.12: Flujo, insumos y resultados de la etapa MA-11.

Antes de llevar a cabo los pasos correspondientes a esta etapa, es importante asegurarse de que los LACs y WACs estén bien definidos y validados. Además, recomendamos comprobar el correcto funcionamiento de los recursos complementarios, tales como servidores centralizados (i.e., la infraestructura *IoA Central Infrastructure*) y distribuidos (i.e., la plataforma IoA), el acceso a los conjuntos de datos públicos, la disponibilidad de los LACs previamente publicados, la accesibilidad al WAC de cada agente, el adecuado funcionamiento de los agentes contrapartes, los objetos IoT, los recursos IoT y la red física IoT.

6.3.5.1. Preparación de la infraestructura física de IoA (MA-11.1)

Teniendo en cuenta que una aplicación de IoA necesita algunos recursos de computación física para operar, en este paso es esencial preparar la infraestructura de IoT física requerida para ejecutar los LOAs. Entre los principales recursos físicos que deben prepararse deben ser contemplados los siguientes: objetos IoT con LOA compatibles, comunicaciones y funciones de la red IoT, servidores empleados por la aplicación (i.e. servicios, agentes, *datasets*, ontologías) y cualquier otro elemento físico recursos que forman parte de la aplicación IoA.

6.3.5.2. Preparación de la infraestructura lógica de IoA (MA-11.2)

De manera similar al paso relacionado con la preparación de los recursos físicos, los recursos lógicos también son importantes para iniciar con éxito la aplicación IoA. Algunos de los recursos lógicos que deben prepararse en este paso son los siguientes: la plataforma para soportar el modelo de agente seleccionado, la herramienta *IoA-Building Tool* para gestionar LOAs, LACs y WACs, el middleware para que el LOA pueda añadir nuevas tareas, los servicios web que permiten el acceso y la comunicación con los objetos físicos, los microservicios invocados por la aplicación IoA y el servidor que soporte los procesos de razonamiento y persistencia del conocimiento semántico.

6.3.5.3. Ejecución del proceso de auto-generación del ecosistema de IoA (MA-11.3)

Este paso es automatizado y consiste en procesar los LACs para generar automáticamente los LOAs que participarán en la aplicación específica. En síntesis, este proceso crea la aplicación objetivo y por lo tanto, después de este paso, la herramienta de construcción de IoA habrá comenzado. Todos los LOAs construidos ejecutan sus tareas correspondientes sobre el ecosistema IoA. Adicionalmente, en este paso el equipo de trabajo debe probar la funcionalidad de cada LOA a nivel de implementación. Además, se debe revisar su comportamiento a nivel microscópico y macroscópico.

6.4. Beneficios del método

El método sistemático propuesto provee beneficios en términos de un instrumento orientado al software que los desarrolladores pueden explotar para crear sistemas agentificados de IoT mediante IoA. A continuación, se resumen los principales beneficios del método:

- Generación semiautomática de aplicaciones de IoT. El método sistemático propuesto en este documento permite a los desarrolladores crear aplicaciones IoA sin necesidad de programación. Esencialmente, el método se enfoca en completar las descripciones semánticas del LAC mediante la aplicación de un proceso de enriquecimiento semántico, así como en determinar el comportamiento del agente mediante el modelado de un WAC específico. La rigurosa descripción de ambos documentos y el uso de herramientas semiautomáticas proporcionarán aplicaciones completas de IoA. Como resultado, los desarrolladores pueden crear agentes más fácilmente que los marcos de trabajo de agentes más actuales (i.e. JADE, Jason), donde los agentes tienen que ser programados completamente.
- Sistemático y fácil de aplicar. El método propuesto es sistemático porque define claramente en detalle las directrices y cuestiones de diseño relevantes que los desarrolladores deben seguir para crear aplicaciones IoA. El desarrollador no tiene el grado de libertad que ofrecen otros métodos más genéricos para crear un proyecto basado en enfoques de IoT o agentes. Por lo tanto, es menos probable que el desarrollador caiga en ambigüedades que podrían tener efectos negativos en el desarrollo de sistemas IoA. De esta forma, los sistemas de IoA se desarrollan siguiendo las mismas directrices y el mismo modelo de agente y, por tanto, es más fácil llevar a cabo el proceso de integración entre sistemas de este tipo orientados a diferentes dominios.
- Menor complejidad en el diseño de aplicaciones de IoT. El método propuesto fue diseñado para reducir la complejidad de emprender un proyecto de IoA, incluso si los desarrolladores no tienen experiencia previa. Para lograrlo, el método establece el conjunto de pasos que los desarrolladores deben seguir para construir sistemas de

IoA colaborativos, inteligentes y autónomos a partir de una infraestructura IoT. A diferencia de una metodología general de agentes, este método está específicamente orientado a trabajar con ecosistemas de IoT. Entonces, las etapas se centran en crear ecosistemas agentificados de IoT dirigidos por LOAs. Como resultado, la curva de aprendizaje requerida no es elevada. Es importante resaltar que no es nuestra intención posicionar nuestro método como una alternativa única para crear aplicaciones inteligentes de IoT, sino más bien proporcionar una alternativa fácil de usar.

La evaluación del método de diseño descrito en este capítulo es presentado en la Sección 8.

6.5. Discusión

Tecnologías emergentes como el IoA no sólo se necesitan mecanismos especializados para gestionar los recursos de IoT de manera inteligente, sino que también es esencial definir métodos formales y sistemáticos que impulsen el proceso de creación de aplicaciones basadas en el enfoque de IoA de manera semiautomática. En este capítulo se ha presentado un método especializado que permite agentificar ecosistemas de IoT. Por otra parte, el método propuesto abarca el modelado de mecanismos de coordinación e inteligencia de los agentes que gestionan los recursos de IoT desde un nivel microscópico y macroscópico. Este método es fácil de seguir y aplicar, y además es perfectamente compatible con el proceso de agentificación del IoT propuesto en este trabajo el cual es dirigido por LOAs.

6.6. Contribuciones relacionadas

El método propuesto en este apartado ha sido evaluado por el comité científico del siguiente medio científico:

P. Pico-Valencia, J.A. Holgado-Terriza and P. Padereski. A Systematic Method for Building Internet of Agents Applications Based on the Linked Open Data Approach. *Future Generation Computer Systems Journal*, 2017. [*En Revisión, Segunda Ronda*].

Pablo A. Pico-Valencia and Juan A. Holgado-Terriza, *ADELE: A Middleware for Supporting the Evolution of Multi-agents Systems Based on a Metaprogramming Approach*, In: Trends in Practical Applications of Scalable Multi-Agent Systems, Seville, Spain, 2016, pp. 297-310.

Capítulo 7

Evaluación del modelo de IoA basado en LOAs

“La historia de la ciencia nos demuestra que toda teoría es perecedera. Con cada verdad que nos es revelada ganamos un mejor entendimiento de la naturaleza, y nuestras concepciones y vicisitudes cambian por completo”—Nikola Tesla.

En el campo científico, cualquier postulado, hipótesis y modelo planteado requieren de previa validación experimental antes de su aceptación y aplicación en proyectos científicos y empresariales.

Con el propósito de validar el modelo de agente semántico propuesto se presentan los resultados de un proceso de evaluación cuantitativa aplicada al modelo de IoA basado en entidades LOAs (*Linked Open agentes*), contratos semánticos LACs (*Linked Agent Contracts*) y flujos de control de agentes WACs (*Workflow for Agent Control*).

Los experimentos realizados para validar los planteamientos del modelo de agente LOA se llevaron a cabo en un ecosistema—que modela una comunidad de vecinos—enfocado en gestionar el confort término y lumínico. La evaluación cuantitativa se ha direccionado en medir el desempeño de los LOAs y de sus mecanismos asociados (i.e., LAC y WAC). Asimismo, se presentan los resultados de la evaluación del desempeño del algoritmo de descubrimiento semántico de agentes contrapartes propuesto para IoA. Adicionalmente, se ha determinado también el grado de precisión y exactitud de los resultado obtenidos por dicho algoritmo de descubrimiento el cual habilita a los LOAs para que sean capaces de ejecutar objetivos colaborativos. Finalmente, los resultados del modelo LOA se comparan con los obtenidos por los mecanismos de descubrimiento implementados por JADE y jUDDI.

7.1. Definición del escenario de IoT

Hemos definido como escenario una comunidad de vecinos denominada $S_{ioa} < SmartCommunity >$. Una comunidad consiste en un grupo de habitantes que viven bajo ciertas normas en apartamentos de diferentes edificios en los que comparten intereses comunes. Para delimitar el escenario propuesto hemos considerado un conjunto de tres comunidades $C = \{c_1, c_2, c_3\}$. Cada comunidad consiste de un edificio individual. Además, cada uno de estos edificios está distribuido en tres plantas (i.e., un apartamento en cada planta). Por lo tanto, el escenario (S) consta de 9 pisos, $F = \{\{c_1f_1, c_1f_2, c_1f_3\}, \{c_2f_1, c_2f_2, c_2f_3\}, \{c_3f_1, c_3f_2, c_3f_3\}\}$.

Cada comunidad ha instalado varios objetos individuales de IoT en sus hogares y uno de ellos dispone de un sistema basado en agentes que controla automáticamente varios objetos de IoT. Los objetos IoT instalados permiten proporcionar confort de temperatura e iluminación. La interacción entre los usuarios y el IoT se lleva a cabo mediante diferentes aplicaciones móviles proporcionadas por los fabricantes de los objetos de IoT. Por ejemplo, los vecinos de la comunidad 1 y 3 pueden controlar los objetos de IoT en sus apartamentos mediante aplicaciones móviles; mientras que la comunidad 2 dispone de un sistema multiagente que controla automáticamente los apartamentos de su comunidad.

Para simplificar el sistema hemos considerado que todos los objetos de IoT pueden ser accesibles a través de openHAB (<https://www.openhab.org/>), un middleware especializado en el desarrollo de hogares inteligentes basados en IoT que separa los objetos físicos de IoT y la aplicación empleando la noción de cosas y elementos. Una cosa representa cualquier objeto IoT que pueda ser accesible por cualquier aplicación, mientras que un elemento es un componente que representa una funcionalidad que es usada por la aplicación en términos de interfaces de usuario o lógica de automatización sobre cosas. Cada ítem tiene asociado un estado, cuyo cambio puede ser supervisado a través de eventos). Ahora bien, en el caso del sistema multiagente, éste se encuentra gestionado por agentes JADE. El diagrama del escenario descrito, así como los objetos de IoT que se instalaron en el interior (i.e., $ext_tempe_c2_f3$) y el exterior (i.e., $ext_tempe_c2_1$) de cada apartamento de cada comunidad se ilustran en la Figura 7.1.

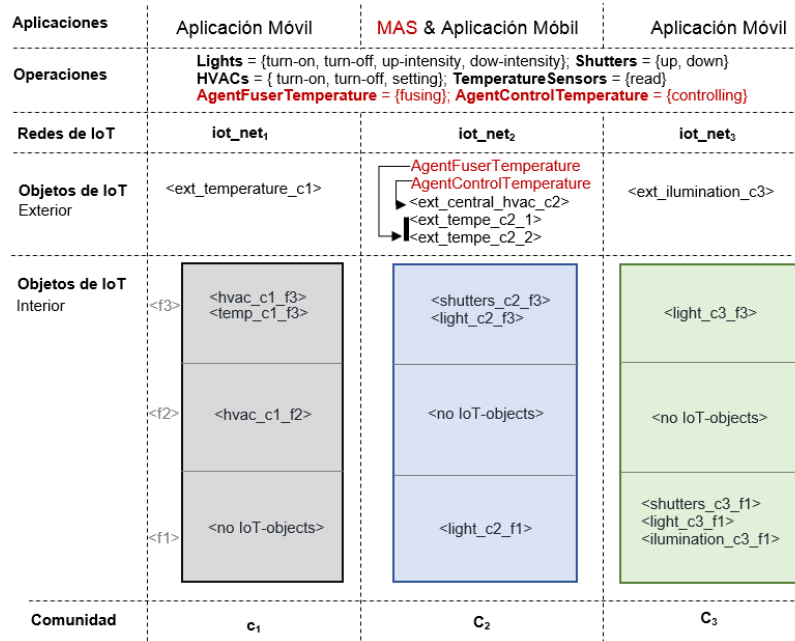


FIGURA 7.1: Diagrama general del escenario de IoT.

7.2. Despliegue de la infraestructura IoA

El desarrollo de un sistema de IoA siguiendo la arquitectura de referencia propuesta (Capítulo 5) ha implicado el despliegue de una infraestructura centralizada donde se implementa la capa semántica denominada *IoA Central Infrastructure*. Esta infraestructura es responsable de recibir, procesar y responder a todas las peticiones de las plataformas de agentes que forman parte del ecosistema de IoA. Para ello, se llevaron a cabo las siguientes tareas: **(i)** persistencia de la ontología IoA-OWL, **(ii)** generación de identificadores únicos para los LOAs y sus LACs, **(iii)** almacenamiento de los *datasets* suplementarios de IoA en el repositorio *IoA Dataset Repository*, **(iv)** generación y almacenamiento de los descriptores semánticos en el repositorio *Centralized-Repository of LACs*, y **(v)** ejecución de los procesos de razonamiento semántico en el ecosistema de IoA.

Todas las tareas listadas se desarrollaron utilizando servicios RESTful para facilitar las peticiones de las plataformas de agentes gestionados en la capa IoA. La implementación de los servicios requeridos por la

infraestructura *IoA Central Infrastructure* que habilitan la ejecución de las tareas anteriores fueron las siguientes:

- Java JDK v1.8 (<https://www.oracle.com/java/index.html>) para crear y ejecutar los servicios RESTful de IoA.
- servidor Glassfish v4.1.1 para desplegar los servicios RESTful de IoA (<https://javaee.github.io/glassfish/>).
- CouchDB v1.6.1 para gestionar los identificadores y los datasets de IoA.
- Virtuoso v7.x (<https://www.openlinksw.com/>) para ejecutar los procesos de razonamiento semántico solicitados por los LOAs.
- Apache Jena v3.0 (<https://jena.apache.org/>) orientados a procesar los grafos RDF y las consultas semánticas.

Por otro lado, a nivel de plataforma de agentes, cada plataforma creada no sólo se deben encargar de crear las instancias LOAs requeridas y mantener la integridad del hábitat de los agentes; sino que también debe interactuar con la infraestructura *Central Infrastructure of IoA* cuando requiere la ejecución de procesos de descubrimiento de agentes contrapartes. Para ello, cada plataforma implementó las siguientes tareas: **(i)** enriquecimiento semántico de los LOAs, **(ii)** generación automática de los LACs y LOAs, **(iii)** ejecución de flujos de control de agente (WAC), **(iv)** gestión del comportamiento de los agentes, **(v)** gestión de las solicitudes receptadas por la infraestructura *IoA Central Infrastructure* y **(vi)** la comunicación a nivel de agente mediante la interfaz asociada a los LOAs. Para la implementación de estos procesos se emplearon las siguientes herramientas:

- JADE v4.3.2 (<http://jade.tilab.com/download/jade/>) para manejar agentes reactivos.
- JSON-LD v1.0 (<https://json-ld.org/>) para procesar documentos JSON-LD.
- OpenHAB v2.x (<https://www.openhab.org/download/>) para abstraer las comunicaciones con los objetos de IoT.
- Java para ejecutar agentes JADE y la herramienta *IoA-Building Tool*.

7.3. Proceso

Con el propósito de proporcionar las principales pautas que un desarrollador debe seguir para modelar un ecosistema de IoA, a continuación se definen los principales pasos que aborda el proceso sistemático para crear un ecosistema de IoA basado en el modelo LOA. El primer procedimiento tiene como finalidad delimitar los objetivos de control del escenario de IoA a nivel de infraestructura. Para ello, se llevaron a cabo los siguientes pasos: **(I-1)** definición del escenario y redes de IoA, **(I-2)** definición del objetivo general y específico de control del escenario de IoA, **(I-3)** selección de los objetos de IoT y su conversión a LOAs, y **(I-4)** despliegue de la plataforma distribuida de LOAs para cada red de IoA.

Por otra parte, el segundo procedimiento se lleva a cabo a nivel de LOA. Por lo tanto, para cada objeto de IoT seleccionado es necesario definir los LOA adecuados y su lógica de control. Para ello, se dieron los siguientes pasos: **(O-1)** modelando el flujo de trabajo (WAC) del LOA, **(O-2)** enriquecimiento semántico del LAC asociado al LOA y **(O-3)** generación del LOA.

7.4. Ecosistema de IoA: Comunidad inteligente

En el modelo propuesto $S_{ioa} < SmartCommunity >$ identificamos 3 redes de $IoT = \{iot_{c_1}, iot_{c_2}, iot_{c_3}\}$. A partir de estas redes se modelaron 3 redes de $IOA = \{ioa_{c_1}, ioa_{c_2}, ioa_{c_3}\}$ orientadas a gestionar los recursos de IoT de cada comunidad. Estos elementos se definieron siguiendo el paso **I-1** de las directrices anteriormente detalladas.

El paso **I-2** requirió la definición del alcance y propósito del sistema modelado. El objetivo general del sistema planteado fue proporcionar un confort colaborativo e inteligente a las familias de las comunidades involucradas, aprovechando los recursos tecnológicos que cada comunidad ha desplegado. El espectro del confort aplicado en nuestro caso se delimitó sólo en el confort térmico y lumínico.

Una vez finalizado el paso **I-3** y basándose en la Figura 7.1, se definió una lista de objetos de IoT. Todos estos objetos de IoT fueron

implementaron implementados y accedidos a través de las interfaces proporcionadas por OpenHAB.

Partiendo de los aspectos técnicos de cada uno de los objetos de IoT, y así dar cumplimiento del paso **O-1**, se creó un flujo de trabajo de control (WAC) para cada objeto. El WAC incluyó la lógica de control para que los LOAs pudieran ejecutar las acciones de control sobre el objeto. Luego, en el paso **O-2**, se realizó el enriquecimiento semántico del LAC de cada LOA ilustrado en la Figura 7.2. Para este cometido se utilizó la herramienta *IoA-Building Tool*, la cual proporciona una interfaz gráfica de usuario sencilla a partir de la cual se introdujeron los valores de los descriptores de los LACs.

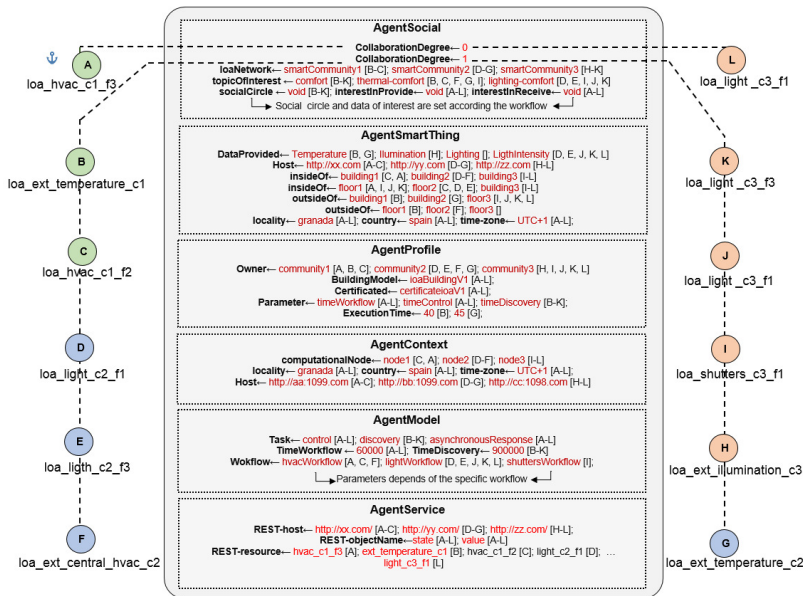


FIGURA 7.2: Descriptores usados para el enriquecimiento semántico de los LOAs del escenario modelado.

La Figura 7.2 muestra también algunos de los descriptores utilizados para describir los LACs de los LOAs que componen el ecosistema propuesto. Dichos descriptores están organizados por componentes incluyendo una letra (A-L) que corresponde al agente asociado específico (i.e., el grado de colaboración de *AgentSocial* para A es 1, Granada es la localidad donde se han instalado todos los objetos).

Luego, mediante la herramienta *IoA-Building Tool*, se ejecutó automáticamente el proceso de generación de la instancia LAC y LOA. En consecuencia, el paso **O-3** fue completado con éxito. Después de esto, cada LOA tuvo su propio LAC y WAC. Consecuentemente, cada LOA pudo ejecutar la tarea de control correspondiente a nivel individual o colaborativo, según el caso. En la Figura 7.3 se ilustra un esquema general del ecosistema IoA creado.

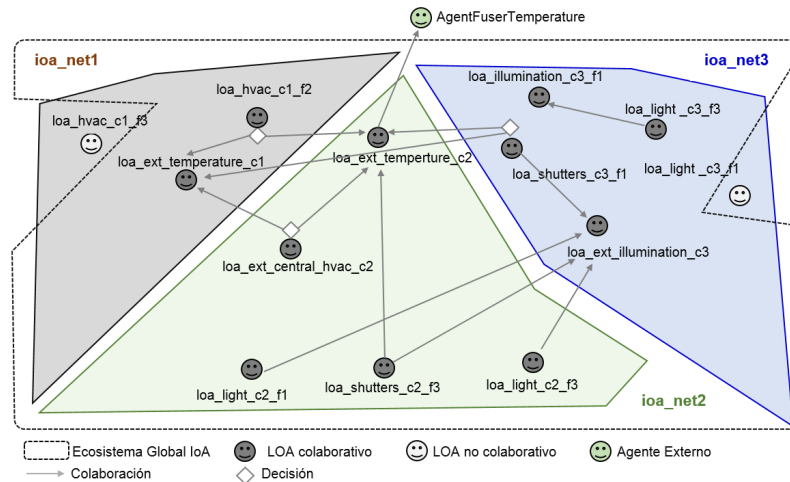


FIGURA 7.3: Esquema del ecosistema de IoA.

La Figura 7.3 muestra el ecosistema global de IoA modelado. Este escenario requirió la organización de todos los objetos de IoT distribuidos en 3 redes de IoA. Todas estas redes constituyeron el ecosistema global de IoA. Es importante señalar que el sistema incorporó LOAs no colaborativas (*loa_hvac_c1_f3* y *loa_light_c3_f1*) y LOAs colaborativas (los restantes). En el primer caso, las LOAs no precisaron interacción con otros LOAs del ecosistema de IoA. Estos agentes eran completamente autónomos y cumplieron sus objetivos utilizando únicamente sus propios recursos de IoT. En cambio, en el segundo caso (LOAs colaborativas), los agentes fueron responsables de lograr sus objetivos mediante la ejecución de un proceso de descubrimiento y selección de las contrapartes disponibles en el ecosistema de IoA. Estos procesos habilitaron a los agentes para que fueran capaces de colaborar con otros agentes en función de sus objetivos.

Para describir más específicamente el comportamiento de cada agente del ecosistema IoA que se muestra en la Figura 7.3, hemos detallado las principales tareas ejecutadas por cada LOA. Dado que este ecosistema contenía un conjunto de 13 LOAs distribuidos en tres redes (*ioa_net1*, *ioa_net2* y *ioa_net3*), agrupamos a los agentes en función de su grado de colaboración y de la naturaleza del objeto de IoT vinculado a cada LOA. A continuación, se formula una breve descripción de cada tipo de agente.

7.4.1. LOAs no colaborativos

En este caso particular, el ecosistema de IoA incluyó dos de estos agentes, *loa_hvac_c1_f3* y *loa_light_c3_f1*. Ambos LOAs no colaborativos fueron introducidos en el ecosistema de IoA con el propósito de contrastar el comportamiento específico de este tipo de LOAs con respecto a los de tipo colaborativo, los cuales son detallados posteriormente.

- *loa_hvac_c1_f3*. Tiene la responsabilidad de gestionar un objeto HVAC utilizando su propio sensor de temperatura integrado en el piso 3 de la comunidad 1. El LOA fue capaz de controlar el encendido y apagado del HVAC dependiendo de la temperatura ambiente y del valor de consigna por medio de una regla específica en el WAC. La temperatura de consigna es fijada o modificada siguiendo una regla definida en el WAC del LOA.
- *loa_light_c3_f1*. En este caso, el LOA específico gestionó un objeto de IoT con una bombilla que integraba un sensor para medir el nivel de iluminación en la planta 1 de la comunidad 3. De acuerdo con el nivel de iluminación ambiental medido en el exterior del edificio, el LOA pudo aplicar una regla WAC para encender y apagar la luz, respectivamente.

7.4.2. LOAs colaborativos

A diferencia de los LOAs no colaborativos, los LOAs colaborativos fueron en su mayoría las entidades que formaron parte del ecosistema de IoA. A continuación se describen dichos agentes:

- *LOAs de bombillas de luz.* Los LOAs de las bombillas, *loa_light2c2f1*, *loa_light2c2f3* y *loa_light3c3f3*, fueron los responsables de encender y apagar las bombillas en los pisos donde fueron instalados. La decisión fue tomada automáticamente en base a los valores de nivel de iluminación recuperados a partir de otros agentes contrapartes debido a que en este caso estas bombillas no integraron ningún sensor a diferencia de la bombilla *loa_light_c3_c3_f1*. Las reglas a seguir por los LOAs vinculados con las bombillas se describen en el WAC asociado. En este caso, el único LOA, *loa_ext;illumination_c3*, fue descubierto y seleccionado por los LOAs, *loa_light2c2_f1* y *loa_light_c2_f3* para solicitar el nivel de iluminación periódicamente y tomar la decisión de si la bombilla asociada debe encenderse o apagarse. De la misma manera, el LOA *loa_light_c3_f3* descubrió y seleccionó el LOA *loa_illumination_c3_f1*.
- *LOAs de persianas.* En este caso, los LOAs de persiana, *loa_shutters_c2_f3* y *loa_shutters_c3_f1*, tuvieron asociados como objeto de IoT un dispositivo que controlaba respectivamente las persianas de la planta 3 de la comunidad 2 y las persianas de la planta 1 de la comunidad 3. Basado en las reglas descritas en el WAC, el LOA pudo decidir si las persianas deberían abrirse o cerrarse dependiendo de los niveles de temperatura e iluminación obtenidos desde otros LOAs. Ambos LOAs de persianas recuperaron los niveles de iluminación del LOA de iluminación, *loa_ext;illumination_c3* que proporcionaba valores de iluminación ambiental externa al edificio. El LOA *loa_shutters_c3_f1* también recuperó el nivel de iluminación interior a partir del agente *loa_illumination_c3_f1*. Por otro lado, los niveles de temperatura pudieron ser obtenidos a partir de uno de los LOAs de temperatura disponibles en el ecosistema de IoA que proporciona la temperatura ambiente fuera del edificio. En este caso, se seleccionó el LOA *loa_ext_temperature_c2* en lugar del LOA *loa_ext_temperature_c1*.
- *LOAs de iluminación.* Estos LOAs, *loa_ext;illumination_c3* y *loa_illumination_c3_f1*, definieron un WAC que proporcionaba

los datos del nivel de iluminación del objeto de IoT de iluminación enlazado a otros LOAs. En este caso específico, este LOA sólo devuelve los valores de iluminación de su objeto IoT enlazado a cualquier petición realizada por otros LOAs, *Light-bulb* y *Shutters*.

- *LOAs de temperatura.* Los LOAs de temperatura, *loa_ext_temperature_c1* y *loa_ext_temperature_c2*, tenían asociado un WAC con una regla enfocada sólo en lecturas precisas de los valores de temperatura ambiente. El agente *loa_ext_temperature_c1* obtuvo el valor de temperatura del objeto de temperatura enlazado *ext_temperature_c1*. Por otro lado, el agente *loa_ext_temperature_c2* obtuvo el valor de temperatura a partir de un agente JADE. Este agente es externo al ecosistema IoA y proporcionó el valor de temperatura fusionado después de acceder a los objetos de temperatura *ext_empe_c2_1* y *ext_empe_c2_2*.
- *LOAs de HVAC.* Estos LOAs, *loa_hvac_c1_f2* y *loa_ext_central_hvac_c2*, se encargaron de controlar el sistema HVAC en el piso 2 de la comunidad 1 y el HVAC de todos los pisos de la comunidad 2. De acuerdo con el WAC asociado, un LOA de HVAC podía encender y apagar el sistema HVAC dependiendo de la temperatura ambiente y de la consigna que tenía que ser recuperada de otros LOAs después de un proceso de descubrimiento. El LOA *loa_hvac_hvac_c1_f2* encontró al LOA *loa_ext_temperature_c1*; mientras que el LOA *loa_ext_central_hvac_c2* seleccionó el LOA contraparte *loa_ext_temperature_c2*.

En general, el ecosistema de IoA modelado (Figura 7.3) proporcionó algunos beneficios a las comunidades que previamente gestionaban el confort térmico y lumínico aplicando tecnologías IoT. En resumen, el modelado de las comunidades mediante un ecosistema de IoA habilitaron a las comunidades para compartir todos sus recursos, así como las colaboraciones potenciales entre los participantes con el fin de realizar tareas de manera proactiva, colaborativa e inteligente.

De manera más específica los beneficios obtenidos tras modelar el escenario de comunidades mediante un ecosistema de IoA se resumen como sigue:

- Los objetos de IoT instalados en una comunidad específica pudieron utilizarse para controlar a las comunidades cercanas que no tenían objetos de IoT específicos (i.e., se encendieron y apagaron las bombillas de la comunidad 2 teniendo en cuenta el nivel de iluminación externa proporcionada por un sensor instalado en la comunidad 3).
- Se reutilizaron sistemas existentes y operativos que controlaban parcialmente ciertos objetos de IoT para que otras comunidades que lo requieran pudieran aprovecharlos (i.e., las persianas de los pisos en la comunidad 3 aprovecharon el valor de temperatura proporcionado por el agente JADE que se ejecutaba en el sistema multiagente implementado por la comunidad 2).
- Se protegieron objetos específicos de IoT a fin de evitar el acceso indiscriminado de cualquier persona a través de LOAs no colaborativos (i.e., el HVAC en el piso 3 de la comunidad 1 y la bombilla en el piso 1 de la comunidad 3 no son visibles en ningún caso dentro del ecosistema de IoA).
- Se realizó un control preciso de los objetos de IoT mediante el uso de magnitudes relacionadas que contribuyeron a mejorar su comportamiento (i.e., las persianas se controlaron mediante el nivel de iluminación y la temperatura del entorno).

7.5. Evaluación experimental

Para evaluar el escenario planteado se implementaron tres plataformas de IoA, *ioa_net1*, *ioa_net2* y *ioa_net3*. Estas plataformas se implementaron en diferentes nodos y con distintas características de procesamiento. El nodo 1 fue a un ordenador Intel Core i7 a 2,5 GHz, 16 GB de RAM y sistema operativo Linux. El Nodo 2 fue un ordenador Intel Core i7 con un sistema operativo de 3,1 GHz, 16 GB de RAM y MAC OS. Asimismo, se implementó el nodo 3 en un ordenador Core i5 con 4 GHz, 16 RAM y sistema operativo Linux. La herramienta *IoA Building Tool* se ejecutó en cada uno de los nodos especificados. La infraestructura *IoA Central Infrastructure* fue implementada en el nodo 1. Sin embargo, esta infraestructura pudo haberse implementado en cualquier lugar, aunque es preferible desplegarla en un nodo independiente.

7.5.1. Desempeño de los LOAs

A partir de las características técnicas de los ordenadores explicados anteriormente, se llevó a cabo una evaluación del rendimiento del modelo de agente LOA. Por las diferencias en las capacidades de cómputo, realizamos todas las pruebas en una sola máquina, el nodo 1, para lograr resultados más consistentes en la ejecución del LOA sujeto a estudio.

El primer experimento cuantitativo se centró en medir el tiempo total ($T_{LOAtotal}$) en mili-segundos (ms) necesario para procesar un LOA y sus elementos. Esto se determinó para los LOAs colaborativos aplicando la ecuación 7.1:

$$T_{LOAtotal} = T_{LACgeneration} + T_{TRIPLETSgeneration} + T_{LOAgeneration} \quad (7.1)$$

donde $T_{LACgeneration}$ corresponde al tiempo requerido para la generación del LAC local, $T_{TRIPLETSgeneration}$ es el tiempo requerido para la creación de las tripletas RDF y su correspondiente publicación en el ecosistema global de IoA, y finalmente $T_{LOAgeneration}$ constituye el tiempo requerido para generar la entidad LOA y su correspondiente inclusión en la plataforma de IoA. Para el caso de los LOAs no colaborativos este tiempo se calcula de manera similar y a partir de la misma ecuación (7.1); pero descartando el tiempo $T_{TRIPLETSgeneration}$ porque éstos no llegan a publicar sus datos en el ecosistema global de IoA.

Con la finalidad de mostrar el tiempo requerido para generar e implementar un LOA y sus elementos, se seleccionó una instancia del escenario modelado. La entidad seleccionada fue *loa_light_c2_f1* y se muestra en la Figura 7.3. La interfaz proporcionada por la herramienta *IoA-Building Tool* se utilizó para llevar a cabo el proceso de enriquecimiento semántico de la entidad LOA, que estaba compuesta por 58 descriptores. Luego, se midieron los tiempos $T_{LACgeneration}$, $T_{TRIPLETSgeneration}$ y $T_{LOAgeneration}$ y a partir de estos resultados se calculó el tiempo ($T_{LOAtotal}$) necesario para generar el LOA *light_c2_f1*. Para obtener tiempos más reales este experimento se ejecutó durante 100 iteraciones. Consecuentemente, en cada iteración se añadió un nuevo LOA al entorno de ejecución del agente del nodo 1 hasta que tengamos 100 LOAs. Además, se descartó la primera iteración, ya que Java requiere inicialmente más tiempo para cargar la máquina virtual Java y la memoria requerida por la aplicación.

Adicional al experimento descrito, también se diseñó un experimento basado en un LOA equivalente al agente *light_c2_c2_f1*. Esta nueva instancia integró el doble de descriptores (58×2) para analizar los requisitos de tiempo relativos al número de descriptores de un LOA. Los resultados de ambos experimentos obtenidos para $T_{LACgeneration}$, $T_{TRIPLETSgeneration}$ y $T_{LOAgeneration}$, se muestran respectivamente en las Figuras 7.4, Figura 7.5 y Figura 7.6.

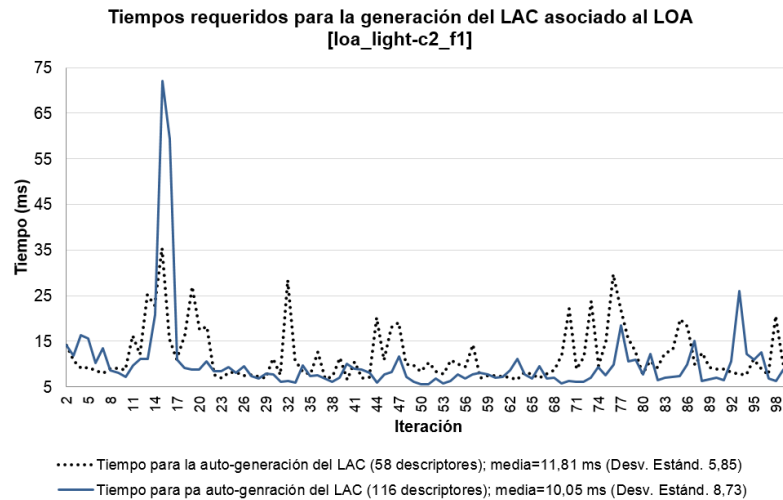


FIGURA 7.4: Tiempos requeridos para el procesamiento de un LAC

Los resultados de las Figuras 7.4 y Figura 7.6 muestran, por un lado, que el tiempo requerido para la creación de LAC y LOA no depende de la cantidad de descriptores integrados por *loa_light_c2_f1*. De hecho, obtuvimos fluctuaciones no significativas alrededor del tiempo medio sin ninguna tendencia notable. El tiempo medio requerido para la generación y almacenamiento del LAC fue de 11,81 ms para el LOA con 58 descriptores y 10,05 ms para el LOA con 116 descriptores—una diferencia del 5%. En cuanto al tiempo necesario para generar el agente y añadirlo a su correspondiente plataforma de IoA, el tiempo medio obtenido fue de 18,04 ms y 16,69 ms para la entidad LOA con 58 y 116 descriptores, con un comportamiento similar en el tiempo de cada iteración con una diferencia de sólo el 1% en este caso.

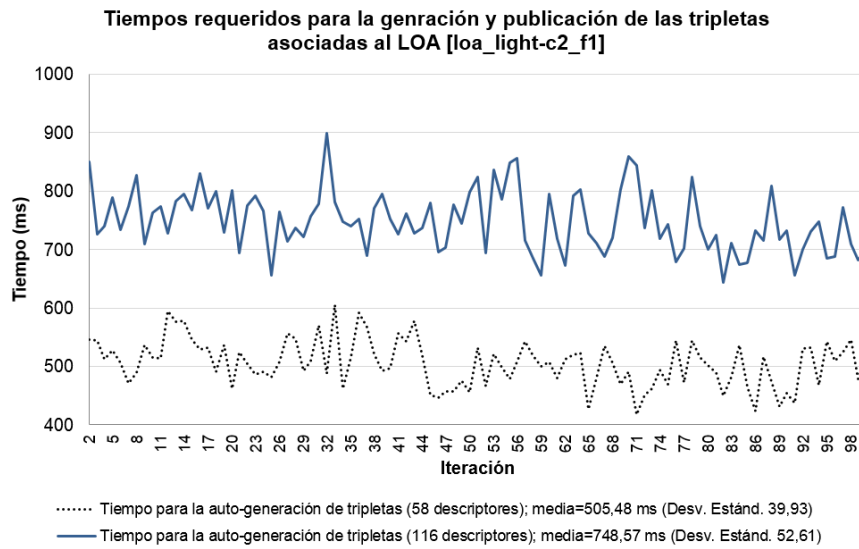


FIGURA 7.5: Tiempos requeridos para el procesamiento de tripletas RDF.

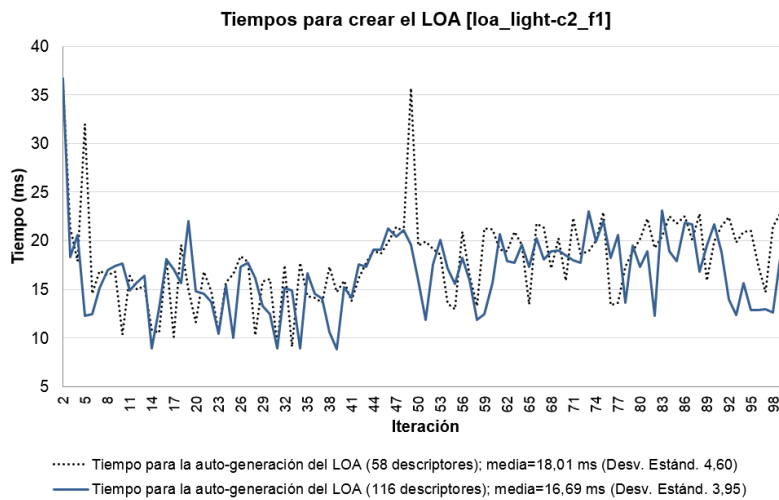


FIGURA 7.6: Tiempos requeridos para el procesamiento de un LOA.

El tiempo requerido para generar las tripletas RDF de un LAC, mostrado en la Figura 7.5, parece indicar una dependencia proporcional directa de $T_{LOA\text{generation}}$ con respecto a la cantidad de descriptores especificados. Para el caso de los 58 descriptores, el tiempo medio requerido fue de 505,48 ms; mientras que para generar las tripletas del LAC de 116 descriptores fue de 748,57 ms—una diferencia del 32%. Obviamente, esta diferencia afectó también al tiempo total requerido para crear el LOA completo (agente, contrato y tripletas). Se obtuvo un tiempo medio de 599,57 ms cuando la instancia de LOA estaba compuesta por 58 descriptores y 841,45 ms para el caso de 116 descriptores. Las pequeñas variaciones observadas en el tiempo de creación del LOA entre iteraciones (especialmente con respecto a $T_{TRIPLETSGeneration}$) revelan que la escalabilidad se mantiene cuando se agregan nuevos LOAs. El aumento de los descriptores semánticos de los LACs podría tener algún impacto en la creación de LOAs (para la creación de tripletas en el repositorio central) y el subsiguiente proceso de razonamiento con tripletas en el descubrimiento de agentes, pero es limitado ya que la diferencia en el tiempo no es demasiado alta.

7.5.2. Desempeño del proceso de descubrimiento

Los LOAs colaborativos deben ejecutar regularmente la tarea de descubrimiento de agentes para encontrar posibles agentes colaboradores como parte de su planificación. Esta tarea realiza un proceso de comparación (*matching*) basada en descriptores para recuperar contrapartes que ayuden en el cumplimiento de los objetivos de los LOAs. El proceso de comparación incluido en la tarea de descubrimiento se basa generalmente en la sintaxis de los descriptores (datos proporcionados), como las Páginas Amarillas de JADE y los catálogos de servicio UDDI utilizados en SOA [87]. Sin embargo, este proceso de correspondencia también puede realizarse basándose en la semántica de los descriptores, como es el caso del modelo LOA.

Con fines experimentales, se prepararon tres escenarios, compuestos por 13 (Figura 7.3 escenarios), 100 y 10.000 agentes, en los que se llevó a cabo el proceso de descubrimiento de contrapartes basado en descriptores. Estos tres escenarios corresponden a (i) las Páginas Amarillas utilizadas por JADE para registrar los servicios asociados con los agentes, (ii) el catálogo de servicios web de jUDDI a los que un agente puede acceder

para recuperar servicios que pueden ayudar a la ejecución de sus tareas (se sugirió una analogía de agente y servicio para llevar a cabo el proceso de descubrimiento), y **(iii)** el repositorio de LACs almacenados en formato tripletas RDF.

El proceso de *matching* asociado a la tarea de descubrimiento de agentes se aplicó a las tres tecnologías especificadas, y para ser equitativos, se registraron dos descriptores básicos, tales como: el nombre y el tipo de datos proporcionados por los agentes o servicios, según corresponda. En el caso de JADE, los agentes se centralizaron en una plataforma y cada agente registró los datos proporcionados bajo el uso de servicios en sus páginas amarillas. En el caso de jUDDI, no se registraron agentes sino servicios web. Así, el agente que ejecutó el proceso de búsqueda tenía la misión de descubrir servicios en lugar de agentes. Estos servicios se agruparon en 3 modelos de negocio diferentes. Finalmente, en el caso de los LACs representados en formato tripletas RDF, éstos se almacenaron en un *endpoint* gestionado por el servidor Virtuoso.

De forma similar al proceso de experimentación anterior, el proceso de descubrimiento de agentes aplicado a las tres tecnologías previamente especificadas se ejecutó durante 100 iteraciones y el proceso de *matching* basado en 1 descriptor (datos proporcionados). Los resultados obtenidos para el agente seleccionado, sin considerar la primera iteración por las razones anteriormente descritas, se muestran en las Figura 7.7 (JADE), Figura 7.8 (jUDDI) y Figura 7.9 (repositorio de LACs).

En primer lugar, en cuanto al descubrimiento con las Páginas Amarillas de JADE (Figura 7.7), hay un incremento en el tiempo en función del número de agentes incluidos en la plataforma. En el caso del ecosistema formado por 13 agentes, el tiempo medio requerido para completar el proceso de descubrimiento fue de 12,13 ms considerando un ecosistema con 13 agentes, 19,46 ms con 100 agentes y 25,64 ms con 1000 agentes. Aunque el tiempo medio fue aumentando con el número de agentes recuperados por el proceso de descubrimiento, el tiempo medio por agente fue mucho menor; por lo tanto, la escalabilidad está garantizada con la tecnología JADE. Además, la Figura 7.9 muestra que el tiempo para descubrir agentes fue disminuyendo a medida que aumentaba el número de iteraciones.

En segundo lugar, el descubrimiento ejecutado en jUDDI (Figura 7.8) necesitó requisitos de tiempo medio mucho más altos que el mecanismo

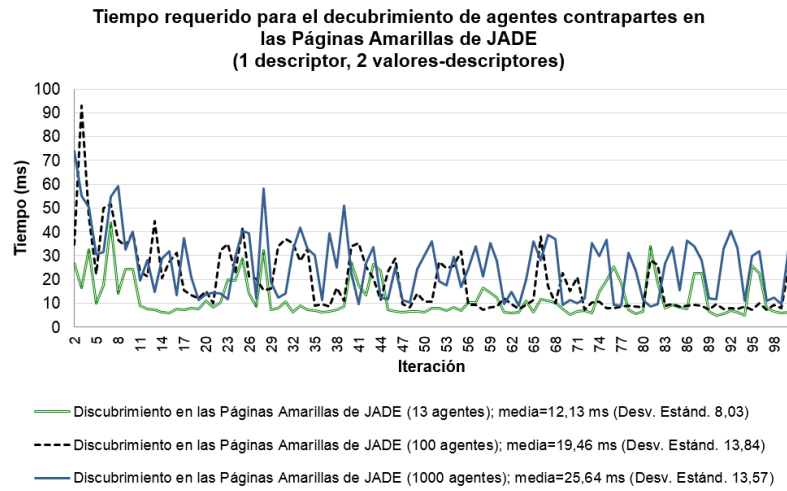


FIGURA 7.7: Tiempos requeridos para el descubrimiento en las Páginas Amarillas de JADE.

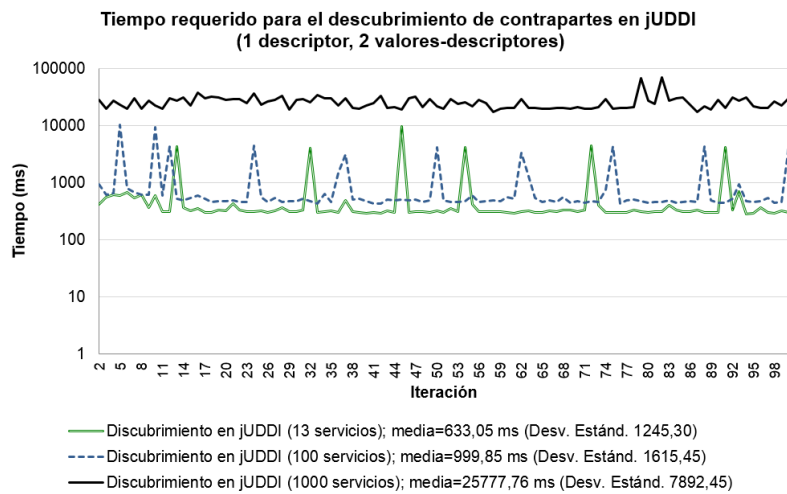


FIGURA 7.8: Tiempos requeridos para el descubrimiento en jUDDI.

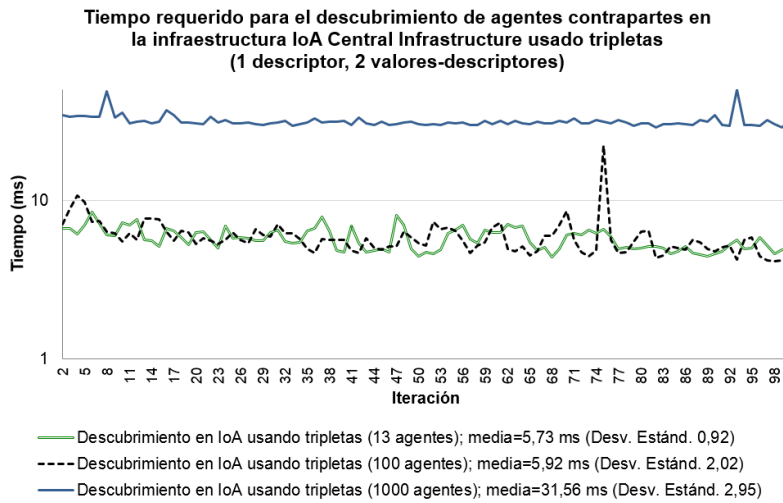


FIGURA 7.9: Tiempos requeridos para el descubrimiento en el repositorio de LACs.

anterior; es decir, 633,05 ms para el ecosistema de 13 agentes, 995,85 ms para el ecosistema de 100 agentes y 25777,76 ms para el ecosistema de 1000 agentes. En este caso, el tiempo medio fue aumentando en función del número de agentes recuperados.

En tercer lugar, se ejecutó un proceso de descubrimiento equivalente sobre el repositorio de LACs almacenados en formato tripletas RDF. Los resultados (Figura 7.9) también muestran una dependencia del número de agentes en el ecosistema de IoA. Sin embargo, estos tiempos son relativamente inferiores. Por ejemplo, en el caso del ecosistema formado por 13 agentes, se requirieron 5,73 ms, lo que equivale a un incremento del 3,20 % y del 81,84 % respecto al mismo escenario compuesto por 100 y 1000 agentes. A pesar de este incremento, como se muestra en la Figura 7.10, el proceso es más eficiente en comparación con los dos métodos anteriores; es decir, un aumento del 71,01 % sobre las Páginas Amarillas de JADE y del 99,87 % sobre el ecosistema de 1000 instancias de agentes.

Los resultados obtenidos anteriormente mostraron un mejor rendimiento del mecanismo de descubrimiento propuesto con respecto a las Páginas Amarillas de JADE y al catálogo de servicios de jUDDI. Para demostrar que los beneficios de la propuesta radican no sólo en el desempeño, sino también en la precisión de los colaboradores recuperados, se

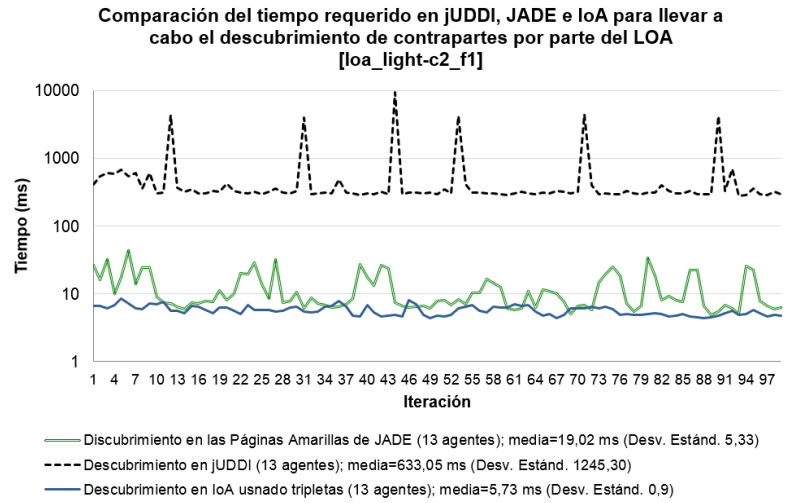


FIGURA 7.10: Tiempos requeridos para el descubrimiento en JADE, jUDDI y el repositorio de LACs.

diseñó un tercer experimento. Este experimento consistió en evaluar el algoritmo de descubrimiento propuesto utilizando un conjunto variable de descriptores semánticos y descriptores-valores sobre el mismo ecosistema de IoA formado por 100 LOAs empleados en el experimento anterior. A partir de estas pruebas, evaluamos el algoritmo de descubrimiento planteado en el Capítulo 5 en términos de la precisión y exactitud de los agentes recuperados en IoA.

7.5.3. Precisión y exactitud del proceso de descubrimiento

El ecosistema de IoA analizado incluyó los 13 LOAs ilustrado en la Figura 7.3 y 87 LOAs adicionales que se generaron estratégicamente para introducir agentes falsos positivos (F^+) y verdaderos positivos (T^+) que pudieran obstaculizar el proceso de descubrimiento de agentes. Un F^+ es un agente recuperado pero no completamente satisfecho con la solicitud y un T^+ es un agente recuperado que satisface completamente la solicitud.

En la Tabla 7.1 se muestran los valores de la precisión (P) y la exactitud (ACC) obtenidas al ejecutar el proceso de razonamiento en JADE, jUDDI y la infraestructura *IoA Central Infrastructure* utilizando los

descriptores 1, 2 y 6. La precisión P en este contexto es la fracción de agentes recuperados que son relevantes para la tarea de razonamiento y la exactitud ACC es el grado en que el resultado de un descubrimiento se ajusta a los valores correctos. Para el cálculo de los valores de P y ACC , se usaron las ecuaciones 7.2 y 7.3, respectivamente:

$$P = T'^+ / (T'^+ + F'^+) \quad (7.2)$$

$$ACC = (\Sigma T'^+ + \Sigma T'^-) / (TotalPopulation) \quad (7.3)$$

CUADRO 7.1: Nivel de precisión y exactitud del proceso de descubrimiento semántico.

Mecanismo	D	V	S	C	F^+	F'^+	T^+	T'^+	P	T^-	ACC
JADE	1	2	Y	100	6	6	20	14	14/20 \simeq 0.70	60	0.74
jUDDI	1	2	Y	100	6	6	20	14	14/20 \simeq 0.70	60	0.74
IoA	1	2	Y	100	6	5	20	15	15/20 \simeq 0.75	60	0.75
JADE	2	4	N	*	*	*	*	*	*	*	*
jUDDI	2	4	N	*	*	*	*	*	*	*	*
IoA	2	4	Y	100	6	3	20	15	15/17 \simeq 0.88	63	0.77
JADE	6	9	N	*	*	*	*	*	*	*	*
jUDDI	6	9	N	*	*	*	*	*	*	*	*
IoA	6	9	Y	100	6	1	20	15	15/16 \simeq 0.94	63	0.79

*El mecanismo no pudo manejar los descriptores del experimento

D=Descriptores; V=Valores; S=Soporte de los descriptores;

C=agentes desplegados dentro del ecosistema IoA;

F^+ =Falsos positivos introducidos; F'^+ =Falsos positivos recuperados;

T^+ =Falsos positivos introducidos; T'^+ =Falsos positivos recuperados;

P =Precisión; T^- Verdaderos negativos; ACC =Exactitud.

Los resultados detallados en la Tabla 7.1 muestran un aumento en la exactitud y precisión de los datos recuperados por el algoritmo de descubrimiento, ejecutado en este caso específico por el agente *loa_shutter* sobre JADE, jUDDI e IoA. En la explicación de las pruebas realizadas, nos referimos al LOA *loa_shutter_c2_f3* como el agente estudiado.

En el primer caso, el mecanismo de descubrimiento utilizó un único descriptor (magnitud del interés) y dos valores descriptivos (temperatura

e iluminación). Estos valores permitieron que el algoritmo de descubrimiento de JADE y jUDDI recuperara un total de 14 agentes verdaderos positivos (T^+) y 6 agentes falsos positivos (F^+). Por lo tanto, los datos recuperados se obtuvieron con una precisión del 70 y el 74 %, respectivamente. Es importante destacar que se recuperaron 6 falsos positivos. Estos agentes proporcionaron las magnitudes requeridas por el agente estudiado pero se encontraban en un contexto diferente al solicitado por el agente. Además, se recuperaron 14 de los 20 agentes verdaderos positivos existentes.

Por otro lado, en el caso del proceso de descubrimiento aplicado sobre IoA, el conocimiento semántico tenía incorporada una regla de igualdad (*sameAs*) entre iluminación y el valor de luz, y a partir de ello se recuperó el agente que describía la magnitud proporcional de luz. Esto no podía hacerse mediante los mecanismos de descubrimiento de las otras dos tecnologías (JADE y jUDDI). Con estas nuevas restricciones, los agentes falsos positivos F^+ se redujeron en 1, de 6 a 5. Esto permitió alcanzar un nivel de precisión y exactitud en el descubrimiento de agentes del 75 % para ambos casos. Esto fue equivalente a una mejora del 5 % en la precisión y del 1 % en la precisión sobre el descubrimiento aplicado a JADE y JUDDI.

En el segundo caso, el mecanismo de descubrimiento ejecutado añadió un nuevo descriptor (contexto de localización *insideOf*) junto con 2 valores adicionales (*community_c1*, *community_c2* y los dos anteriores). Con estos nuevos descriptores, el número de agentes falsos positivos F^+ se redujo de 6 a 3. Consecuentemente, se logró un nivel de exactitud de 83 % y una exactitud de 77 %. Esto representó un aumento del 13 % y del 2 % en ambas medidas. Es importante señalar que se recuperaron 2 agentes falsos positivos F^+ más. Los 3 agentes falsos positivos restantes (F^+) fueron hallados dentro de las casas y correspondieron al contexto requerido.

Finalmente, en este último caso, el algoritmo de descubrimiento fue capaz de discriminar entre dos agentes adicionales porque se especificaron dos nuevos descriptores. Estos descriptores fueron, el tiempo máximo de ejecución requerido por el agente para ejecutar la tarea solicitada y el ancho de banda del objeto fue de 54 Mbit/s o 11 Mbit/s. Por lo tanto, la precisión del algoritmo con respecto a los agentes recuperados fue del 94 %. Esto corresponde a un incremento del 18 y 5 % sobre el uso de 1 y 2

descriptores respectivamente. Además, la tendencia de precisión muestra un aumento del 79 %.

El análisis anterior demuestra que el modelo de descubrimiento de agentes propuesto aumenta su nivel de precisión a medida que se incluyen descriptores adicionales. Esto se debe básicamente a que, como en los casos 2 y 3, el algoritmo discrimina a los agentes que no cumplen con las restricciones que el agente requiere para llevar a cabo tareas de control en el ecosistema de IoA. Por otro lado, aunque el número de descriptores utilizados para realizar un proceso de descubrimiento puede ser una condición para que los agentes verdaderos positivos (T^+) sean discriminados, como se muestra en el caso 1, un razonamiento semántico permite reducir el número de agentes verdaderos negativos (T^-) y aumentar la probabilidad de descubrir agentes verdaderos positivos (T^+). Sin embargo, el modelo semántico debe incluir las reglas apropiadas (por ejemplo, igualdad, transitivo, inverso) para que el motor de razonamiento funcione de manera coherente.

7.6. Discusión

Los descriptores semánticos integrados en la ontología IoA-OWL a través de sus seis componentes ontológicos—*AgentProfile*, *AgentContext*, *AgentSocial*, *AgentService*, *AgentModel* and *AgentSmartThing*—habilitaron a los LOAs para ser capaces de conocer la suficiente información que puede ser utilizada por los razonadores para tomar decisiones autónomas a nivel global y con mayor precisión que los mecanismos dirigidos por operaciones puramente sintácticas como muestra la Tabla 7.1. Además, el procesamiento de enriquecimiento semántico de los LACs reduce el tiempo requerido para el desarrollo del agente. Esto se debe a que el proceso de creación de los LOAs es completamente automático y no requiere programar ninguna entidad.

El proceso de razonamiento semántico solicitado por los agentes desde estas plataformas distribuidas se realiza sobre una infraestructura centralizada (*IoA-Central Infrastructure*). Las tripletas RDF (réplica de LACs distribuidos) también se almacenan en esta infraestructura. Esto significa que si la infraestructura *IoA-Central Infrastructure* no está disponible, los agentes que necesitan completar su WAC no pueden hacerlo hasta que la infraestructura esté disponible de nuevo. Sin embargo, esta limitación

puede superarse creando réplicas de la infraestructura central de IoA o creando un mecanismo de federación para este tipo de infraestructura.

7.7. Contribuciones relacionadas

Los resultados descritos en este capítulo han evaluados por parte de un medio científico de alto impacto. Los detalles de la publicación se describen a continuación:

P. Pico-Valencia, J.A. Holgado-Terriza and J. Senso. Towards an Internet of Agents Model Based on Linked Open Data Approach. *Journal of Autonomous Agents and Multi-Agent Systems* [En Revisión, Segunda Ronda].

Capítulo 8

Aplicación del método sistemático de agentificación del IoT

“Hay tres etapas en el descubrimiento de la ciencia. Primero, las personas niegan que es cierto, luego niegan que es importante; finalmente le dan el crédito a la persona equivocada”—Bill Bryson.

En el área de la Ingeniería de Software los métodos y metodologías formales orientadas al desarrollo de sistemas informáticos requieren de herramientas especializadas que guíen paso a paso el proceso que los equipos de desarrollo deben seguir. Dichas herramientas son imprescindibles sobre todo cuando se trata de la aplicación de nuevos paradigmas como es el caso del modelado de ecosistemas de IoA basado en entidades LOAs, propuesto en este trabajo.

En términos generales, se describe de manera detallada la aplicación de cada una de las fases, etapas y pasos definidos por el método sistemático planteado en este trabajo para realizar el proceso de agentificación del IoT mediante entidades LOAs (*Linked Open Agents*), LACs (*Linked Agent Contracts*) y WACs (*Workflow for Agent Control*). El método es aplicado para el modelado de un ecosistema de IoA en el área de Inteligencia Ambiental que busca proveer de confort térmico en una oficina.

La evaluación del método sistemático se realizó en función del sistema resultante. Para su evaluación, se ha planteado un instrumento de valuación cualitativo—constituido por 22 criterios—a partir del cual se evaluó la funcionalidad, escalabilidad, adaptabilidad, interoperabilidad, usabilidad y fiabilidad del ecosistema de IoA modelado.

8.1. Descripción del escenario de IoT

La red de IoT se ha desarrollado utilizando la tecnología OpenHAB está formado de 14 objetos de IoT—2 sensores de iluminación, 3 sensores de temperatura, 3 bombillas de luz compatibles con los cambios de color y brillo, 2 mecanismos de ventana, 2 mecanismos de persianas, 1 sensor de movimiento y 1 sistema de calefacción, ventilación y aire acondicionado (Calefacción, Ventilación y Aire Acondicionado (*Heating, Ventilation, and Air Conditioning*, HVAC por sus siglas en inglés)—categorizados en 7 tipos de objetos virtuales—dimmer, contacto, color, interruptor, persiana de rodillo, cadena y número.

8.2. Despliegue de la infraestructura de IoA

El escenario específico de IoA debe cubrir los siguientes requisitos funcionales: **(i)** el confort de la temperatura en la oficina debe fijarse normalmente antes y durante la jornada laboral, y también después de la misma si algún investigador sigue trabajando, **(ii)** la aplicación de políticas de ahorro energético en el sistema es estrictamente necesaria con una prioridad menor al servicio de confort de la temperatura, **(iii)** durante la jornada laboral debe mantenerse un nivel de luz confortable para no afectar a los usuarios que utilizan frecuentemente ordenadores. Por último, **(iv)** el funcionamiento del sistema debe basarse en predicciones meteorológicas para proporcionar comodidad en función de las condiciones meteorológicas futuras de la zona.

8.3. Modelado a nivel microscópico

De acuerdo con el escenario descrito anteriormente, a continuación en la Figura 8.1 se presenta el mapa de la red física de IoT, incluyendo los componentes para componer el modelado microscópico (MI-1 a MI-6) y sus respectivos pasos. Este mapa muestra tres tipos de componentes. El primero (punto de acceso a Internet) y el segundo (dispositivos de red) definen los principales recursos asociados a una red global de Internet y los recursos para conectarse de forma inalámbrica en la red específica. En cambio, el tercer componente del mapa (dispositivos) corresponde a los 14 objetos de IoT. Esta información es esencial para llevar a cabo la

fase microscópica en la que los objetos de IoT se han de modelar como LOAs de tipo *Agent of Things*.

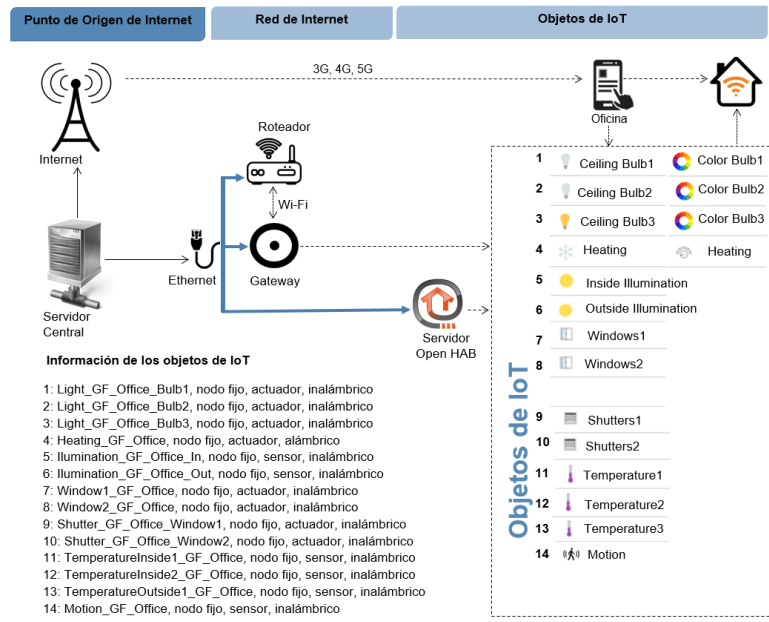


FIGURA 8.1: Mapa de la red física de IoT.

Una vez definido el mapa de la red de IoT, cada una de las etapas a nivel microscópico (MI-1 a MI-6) fueron realizadas iterativamente por cada objeto integrado en el mapa de la red de IoT de la Figura 8.1. Aunque el proceso ha sido realizado individualmente por cada objeto de IoT, a continuación se describen brevemente las seis primeras etapas.

8.3.1. Planificación de recursos del objeto de IoT (MI-1)

La realización del paso MI-1.1 por cada uno de los 14 objetos que componen el mapa de IoT, nos ha permitido recopilar los principales descriptores asociados a los recursos de los objetos de IoT, tales como: identificador, datos generados, acciones realizadas, marca, ubicación, geo-localización y contexto temporal basado en el entorno real donde los dispositivos físicos fueron instalados. Esta información se ilustra en la Figura 8.2-a.

(a)	(b)	(c)	(d)	(e)
Nombre de agente (n)	Descripciones AgentSmartThing: Object Identifier (Id), Time context (Ti), Location Context (L), Capabilities (C), Own Agents (A), Own Services (S), DataProvided (D), Measurement Unit (U)	Descripciones AgentProfile: Agent Name (n), Owner (O), CreatedFrom (B), Status (S), Version (V), Date Submission (DS), Date Publishing (DP), Certificate (C)	Descripciones AgentContext: Context Identifier (Id), Time context (Ti), Location Context (L), Geolocation context (G), Agent Platform (P), Computation Node (N)	Descripciones AgentModel: Generic Tasks (Ti), Workflow for Agent Control (W), Message Supported (M)
ALightBulb1	id="{Light_GF_Office_Bulb1}", L="{on the left side of ceiling}", D={luminancia}, U={lux}	n="{ALightBulb1}", C="{CERT-AoT-00001}"	id="{Light_GF_Office_Bulb1}_context"	W={WACLighting}, M={GetLightLevel, SetLightColor, GetLightColor, TurnOnLight, TurnOffLight}
ALightBulb2	id="{Light_GF_Office_Bulb2}", L="{on the right side of ceiling}", D={luminancia}, U={lux}	n="{ALightBulb2}", C="{CERT-AoT-00002}"	id="{Light_GF_Office_Bulb2}_context"	
ALightBulb3	id="{Light_GF_Office_Bulb3}", L="{on the center of ceiling}", D={luminancia}, U={lux}	n="{ALightBulb3}", C="{CERT-AoT-00003}"	id="{Light_GF_Office_Bulb3}_context"	
AHVAC	id="{Heating_GF_Office}", L="{far away from the temperature sensors}", D={temperature}, U={C}	n="{AHVAC}", C="{CERT-AoT-00004}"	id="{Heating_GF_Office}_context"	W={WACHVAC}, M={TurnOnHVAC, TurnOffHVAC, SetTemperature, GetTemperature}
AlluminationIn	id="{Illumination_GF_Office_In}", L="{indoor the building}", D={luminancia}, U={%}	n="{IlluminationIn}", C="{CERT-AoT-00005}"	id="{Illumination_GF_Office_In}_context"	W={WACIllumination}, M={GetIlluminationLevel}
AlluminationOut	id="{Illumination_GF_Office_Out}", L="{outdoor the building}", D={luminancia}, U={%}	n="{IlluminationOut}", C="{CERT-AoT-00006}"	id="{Illumination_GF_Office_Out}_context"	
AWindows1	id="{Window1_GF_Office}", L="{in the living room}", D={State}, U={boolean}	n="{AWindows1}", C="{CERT-AoT-00007}"	id="{Window1_GF_Office}_context"	W={WACWindows}, M={OpenWindow, CloseWindows}
AWindows2	id="{Window2_GF_Office}", L="{in the living room, near to the TV}", D={State}, U={boolean}	n="{AWindows2}", C="{CERT-AoT-00008}"	id="{Window2_GF_Office}_context"	
AShutter1	id="{Shutter_GF_Office_Window1}", L="{on the window1}", D={State}, U={boolean}	n="{AShutter1}", C="{CERT-AoT-00009}"	id="{Shutter_GF_Office_Window1}_context"	W={WACShutters}, M={ShuttersUp, ShuttersDown}
AShutter2	id="{Shutter_GF_Office_Window2}", L="{on the window2}", D={State}, U={boolean}	n="{AShutter2}", C="{CERT-AoT-00010}"	id="{Shutter_GF_Office_Window2}_context"	
ATemperature1	id="{Temperature1_GF_Office}", S={TemperatureConv}, L="{far from HVAC}", D={temperature}, U={C}	n="{ATemperature1}", C="{CERT-AoT-00011}"	id="{Temperature1_GF_Office}_context"	W={WACTemperature}, M={GetTemperature}
ATemperature2	id="{Temperature2_GF_Office}", S={TemperatureConv}, L="{far from HVAC}", D={temperature}, U={C}	n="{ATemperature2}", C="{CERT-AoT-00012}"	id="{Temperature2_GF_Office}_context"	
ATemperatureOut	S={TemperatureConv}, L="{outdoor the building, outside of Bulb1}", D={temperature}, U={C}	n="{ATemperatureOut}", C="{CERT-AoT-00013}"	id="{Temperature_GF_Build}_context"	
AMotion	id="{Motion_GF_Office}", L="{on the ceiling, indoor the building}", D={presence}, U={%}	n="{AMotion}", C="{CERT-AoT-00014}"	id="{Motion_GF_Office}_context"	W={WACMotion}, M={DetectPresence}
Todos los LOAs	A={O, S, V, V1}, T={UTC}, D={Generic, Operate, S, C, M, Maracena, Spain, Lat:37.205421, Long:-3.623795}	O="{SmartTech}", V="{1.0}", S="{device}", B="{Agent_Cert_Ting_Building_V1}"	T="{UTC}", C="{Maracena, Spain, Lat:37.205421, Long:-3.623795}", N="{https://openhab.org}", P="{smarthome}", L="{in local platform}"	T={Workflow Execution Task, Agent Discovery Task, Injection Task}, M={Injection}



FIGURA 8.2: Principales descriptores empleados para el enriquecimiento semántico de los LOAs.

Por otro lado, dado que todos los objetos son compatibles con interfaces REST implementadas por el middleware *OpenHAB*, los agentes software pueden gestionar estos objetos. Por último, una vez completada la etapa MI-1.3, se obtiene una colección de *datasets* con las descripciones de los objetos de IoT del escenario modelado. Estos datos fueron almacenados en un servidor CouchDB.

8.3.2. Descubrimiento de agentes públicos especializados asociados al objeto de IoT (MI-2)

Teniendo en cuenta que esta es la primera aplicación construida al aplicar esta propuesta, el paso MI-2.1 no pudo llevarse a cabo porque no existen LOAs públicos que puedan ser integrados como parte de la aplicación modelada. Por lo tanto, como el ecosistema de IoA está actualmente vacío, el proceso continúa hasta el siguiente paso (MI-2.2). Sin embargo, la ejecución del paso MI-2.2 tampoco tuvo éxito. En resumen, como no fue posible descubrir agentes genéricos que ayudaran a resolver tareas de confort en sistemas de oficinas inteligentes, entonces esta etapa concluyó sin un resultado específico.

8.3.3. Planificación del ecosistema de servicios del objeto de IoT (MI-3)

Dado que el middleware OpenHAB se empleó para modelar el escenario de IoT, el paso MI-3.1 fue fácil de cubrir, ya que este entorno proporciona una API de REST que permite la gestión de la acción de los objetos de IoT a través de servicios de REST. Esta etapa también fue aplicada de forma iterativa por cada uno de los objetos de IoT. Finalmente, se obtuvo un conjunto de descriptores que representan el ecosistema de servicios gestionado por los agentes *Agents of Things*, como muestra la Figura 8.2-b.

Es importante destacar que tras la finalización del paso MI-3.2 se publicaron los *datasets* que describen el ecosistema de IoS. Por último, en el paso MI-3.3 se puso a prueba cada servicio asociado con cada objeto de IoT interconectado dentro del escenario específico. Todos fueron ejecutados con éxito.

8.3.4. Planificación de la arquitectura y el comportamiento del LOA (MI-4)

En primer lugar, en el paso MI-4.1 definimos el rol *Agent of Thing* como el rol principal para todos los agentes modelados en esta fase. También se definió la arquitectura de agente reactivo como base para construir cada LOA modelado a nivel microscópico.

En segundo lugar, en ambos pasos MI-4.2 y MI-4.3 los descriptores relacionados con el *AgentProfile* (Figure 8.2-c) y *AgentContext* (Figura 8.2-d) fueron definidos iterativamente para cada uno de los LOAs *Agent of Thing* incluidos en el mapa de IoT. Estos descriptores fueron definidos siguiendo las mismas especificaciones en los pasos MI-2.2.a y MI-2.2.b, respectivamente.

En tercer lugar, en el paso MI-4.4 definimos y construimos el WAC específico para cada LOA siguiendo el esquema correspondiente.

En cuarto lugar, definimos las tareas de agente que dirigen el comportamiento de cada LOA. Fue así como se completó el paso MI-4.5. Es importante destacar que no fue necesario integrar tareas adicionales para ninguno de los LOAs modelados ya que la lógica de control almacenada en su correspondiente WAC satisfizo las funcionalidades en su totalidad. Al final, la información más destacada obtenida tras la realización de este paso se resume en la Figura 8.2-e.

El paso MI-4.5 se llevó a cabo mediante el modelado del comportamiento de los LOAs usando un Diagrama de Estados UML. Para ejemplificar cómo se usó este tipo de diagrama para representar el comportamiento de un LOA reactivo, en la Figura 8.3 se ilustran los principales estados del agente y las tareas especializadas del agente AHVAC. De la misma manera, son definidos los diagramas de los LOAs restantes. En el caso específico del LOA AHVAC, éste agente tuvo un WAC enfocado en el control del sistema HVAC para proporcionar confort de temperatura en el interior del escenario modelado.

En términos generales, el agente AHVAC tuvo tres estados genéricos—*initialized*, *active* y *waiting*—, tres nuevos estados—*reinitializing* y *processing*—y tres tareas temporales asociadas. Los estados genéricos permitieron el funcionamiento normal de los agentes JADE, los nuevos estados *injecting* y *reinitializing* se ejecutan cuando se produce un proceso de inyección de tarea externa dentro del modelo de

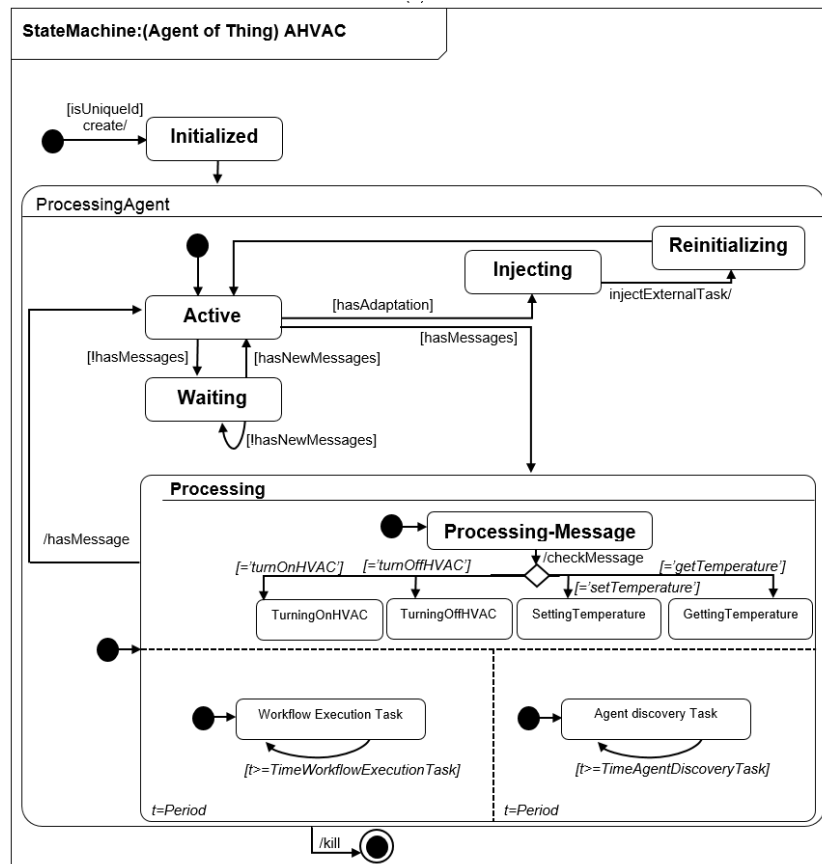


FIGURA 8.3: Diagrama de estado UML del LOA AHVAC.

agente (este no fue el caso) y, por último, el estado *processing* se ejecutó cuando se desencadenó un evento (i.e., recibir un mensaje, activar una tarea programada) y el agente debió interceptarlo para su respectivo procesamiento.

Además, la Figura 8.4 ilustra que el agente *AHVAC* es capaz de reaccionar a tareas temporales que han sido planificadas por el programador del LOA (*Asynchronous Agent Response Task*, *Agent Discovery Task*, *Agent Injection Task* y *Workflow Execution Task*) para que éste reaccionara cuando el LOA recibiera mensajes con contenido “*textitturnOnHVAC*”, “*textitturnOffHVAC*”, “*textitsetTemperature*” o “*getTemperature*”. Estos mensajes pueden ser enviados desde agentes externos pertenecientes a una de las categorías de las particiones definidas en el método

sistemático. Por ejemplo, cuando el agente recibe el mensaje “textitturnOnHVAC”, el LOA AHVAC ejecutó la acción “textitturnOnHVAC” que activa el sistema HVAC. Esta acción se realiza mediante el servicio web *Heating_GF_Office*. Los agentes restantes fueron modelados de manera similar al agente *AHVAC*, pero definieron sus comportamientos correspondientes. Por este motivo no se ilustró el diagrama correspondiente a cada uno de los LOAs del ecosistema de IoA. Así es como completamos el paso MI-4.6.

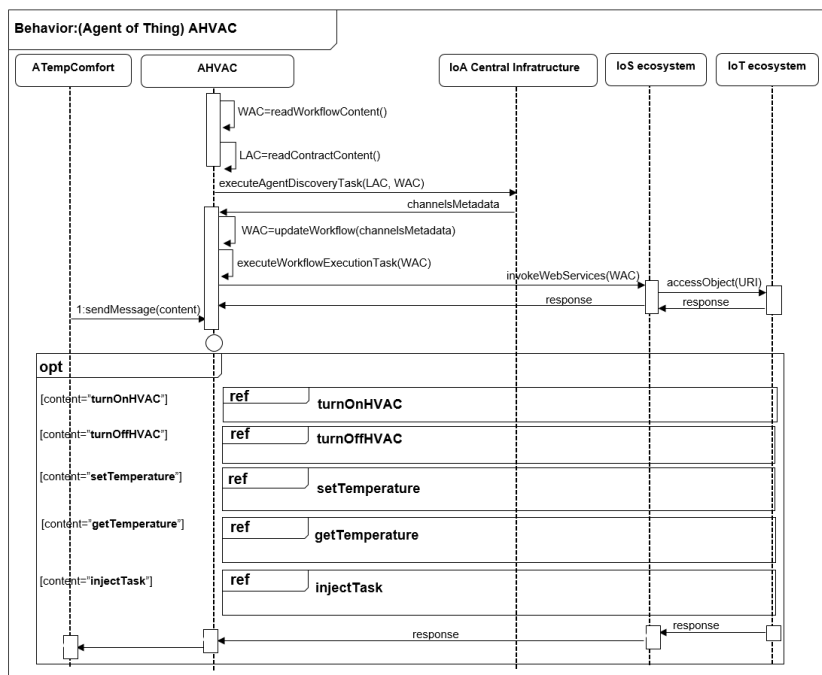


FIGURA 8.4: Diagrama de actividad UML del LOA AHVAC.

Finalmente, en el paso MI-4.7 se publicaron dos datasets públicos (parámetros y restricciones en tiempo real) y uno privado (comportamientos/tareas) relacionados con el modelo de agente LOA.

8.3.5. Enriquecimiento semántico del LAC (MI-5)

La finalización de las etapas MI-1, MI-2, MI-3 y MI-4 permitió que la ontología IoA-OWL contara con datos recopilados para llevar a cabo

el proceso de enriquecimiento semántico (asignación de valores a los descriptores semánticos) de las secciones contractuales *AgentProfile*, *AgentSmartThing*, *AgentContext*, *AgentService* and *AgentModel*. Estas secciones contractuales se enriquecieron a partir de los descriptores de la Figura 8.2-c, Figure 8.2-a, Figure 8.2-d, Figure 8.2-b. Además, se empleó también la lista de tareas de cada agente del ecosistema IoA detalladas en el paso MI-4.5. Este proceso se llevó a cabo mediante el empleo de la herramienta *IoA-Building Tool*.

8.3.6. Generación del LAC del LOA (MI-6)

Esta etapa de generación automática de secciones contractuales que conforman un LAC—con la excepción de la parte social que debe enriquecerse en la fase macroscópica—se llevó a cabo para cada LOA de manera iterativa utilizando la herramienta *IoA Building Tool* y los descriptores especificados en el proceso de enriquecimiento semántico de la etapa anterior (MI-5). Por ejemplo, en el caso específico de la *ATemperatura* LOA vinculado al sensor *TemperatureInside1_GF_Office* se obtuvieron las seis secciones contractuales incluidas en un único documento vinculado, siguiendo la sintaxis del LAC general. También se generó una copia del LAC en format RDF (asociados a los 14 objetos) y se los almacenó en la infraestructura *IoA Central Infrastructure*.

8.4. Modelado a nivel macroscópico

Una vez efectuada la fase microscópica, todos los agentes de la Partición 1, junto con sus contratos correspondientes habían sido creados. No obstante, modelar un ecosistema de IoA implicaba el uso de LOAs adicionales cuya función es coordinar las acciones que los LOAs extras que coordinaron las acciones que los LOAs *Agents of Thing* debían ejecutar para lograr los objetivos del ecosistema de IoA. Siguiendo el método sistemático propuesto, definimos 8 nuevas instancias del agente de acuerdo con las particiones de agente 1, 2 y 3.

8.4.1. Planificación de las particiones de agentes 1, 2 y 3 (MA-7)

A partir de las particiones de agente 1, 2 y 3, en esta etapa se definieron los siguientes LOAs: *Agents of Coordination* (ALightComfort, ATemperatureComfort y AGeneralComfort), *Agents of Data* (ADataFusion, AWeather e AHistoricize), y *Agents of People* (AMonitorOffice y AUserPreferences). Estos agentes estuvieron organizados como ilustra la Figura 8.5 y sus funciones estuvieron centradas en el control de los siguientes aspectos:

- *ALightComfort* para controlar los objetos de IoT adecuados buscando todo tipo de valores de luz y así proporcionar confort lumínico dentro del escenario.
- *ATemperatureComfort* similar al LOA *ALightComfort* pero enfocado en buscar todo tipo de valores de temperatura.
- *AGeneralComfort* para coordinar el LOA *ALightComfort* y el LOA *ATemperatureComfort*, y así cambiar al modo de confort (green o basado en la política de confort extra definida por el usuario).
- *Weather* para obtener datos de estaciones meteorológicas remotas y obtener las condiciones de temperatura y humedad de una zona específica.
- *ADataFusion* para fusionar los datos relacionados con las condiciones meteorológicas dentro y fuera del escenario y así obtener información más precisa que ayude a tomar decisiones de acuerdo con la política de confort adoptada por el sistema

Por otro lado, incluimos dos agentes adicionales tales como: (vi) *AHistoricize* para guardar información sobre las condiciones meteorológicas y las acciones aplicadas por el sistema, (vii) *AMonitorOffice* para proporcionar información sobre las condiciones de confort tanto en formato de texto como gráfico del escenario específico, y finalmente (viii) *AUserPreferences* para gestionar los perfiles de usuarios que definen los parámetros de configuración del sistema.

Para cada uno de los agentes listados se llevaron a cabo los pasos desde MI-2 hasta MI-6, de una manera iterativamente similar a la de

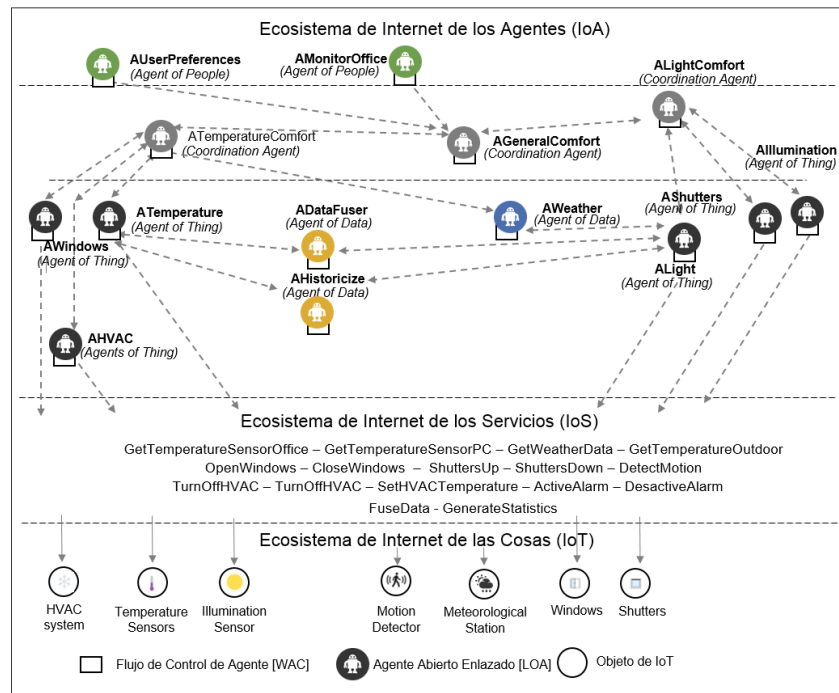


FIGURA 8.5: Representación del ecosistema de IoA.

los agentes de la partición 1, descritos en la aplicación de la fase 1. Con respecto al paso MI-2, no se descubrieron agentes genéricos ni LOAs existentes. Además, en el paso MI-3 no se descubrió ningún servicio existente en el ecosistema de IoS y, por tanto, todos los servicios tuvieron que ser creados. Esto significó la publicación de nuevos registros en el *dataset* de servicios. En relación con la etapa MI-4, se definieron los descriptores de las secciones contractuales *AgentProfile*, *AgentContext* y *AgentModel*, junto con los descriptores de la sección contractual *AgentService* especificados en la etapa MI-3. A partir de estos descriptores, se llevó a cabo el proceso de enriquecimiento semántico (paso MI-5) y la generación de los contratos de agentes (paso MI-6). Es importante destacar que cada uno de los agentes se definió siguiendo el mecanismo similar de representación que se ilustra en la Figura 8.3 y Figura 8.4.

8.4.2. Planificación social del ecosistema global (MA-8)

El uso del protocolo de comunicación adecuado es clave para que los agentes logren ser sociales y colaborativos. En el paso MA-8.1 se definieron los protocolos para cada categoría de agente. Se empleó el protocolo *FIPA-Request-Protocol* para agentes de la partición 4 (*Agente de Personas*) ya que éstos sólo acceden a la configuración del usuario y utilizan estos datos para interactuar con agentes de la partición 2. El mismo protocolo fue usado por los agentes de la Partición 3 (*Agente de Datos*) porque sólo actúan responsivamente cuando una solicitud es hecha por agentes de la Partición 2 y 1. Por otro lado, la cuestión social entre agentes de la Partición 1 y 2 fue soportada por el protocolo *FIPA-Contract-Net*. Dichos protocolos habilitaron a los LOAs para que pudieran comunicarse con sus contrapartes de acuerdo a su WAC.

En cuanto al tema y los datos de interés, la definición de redes y círculos sociales realizada en los pasos MA-8.2, éstos fueron definidos por los propios LOAs después de la ejecución de los procesos de razonamiento, como se ilustra en la Figura 8.6. Esta figura muestra los agentes que forman una red social (denominada *SNBuildingMaracena1*) y la relación (denominada *colabora*). A partir de esta información, la matriz de adyacencia (denominada *MBuildingMaracena1*) y las medidas de centralidad de cada actor (agente) fueron inicializadas por la herramienta *IoA Building Tool*. En resumen, como muestra el sociograma de la Figura 8.7, al principio, el agente con mayor poder social fue el agente *ALightComfort*— $\text{degree}=12$, $\text{closeness}=0.51$, $\text{betweenness}=0.43$ —. Sin embargo, estos valores cambian a medida que nuevas relaciones se incluyan en el modelo social del sistema. En conclusión, la información ilustrada se utilizó para suministrar el componente *AgentSocial* que se generó previamente pero que estuvo vacío hasta que cada agente empezó a ejecutarse.

8.4.3. Planificación del comportamiento del ecosistema de IoA (MA-9)

En el primer paso de esta etapa (MA-9.1) empleamos un Diagrama de Comunicación UML para definir todas las interacciones y mensajes que se transmiten entre los LOAs. La información incluida en este diagrama sigue los alineamientos modelados dentro de la red social definida. Así,

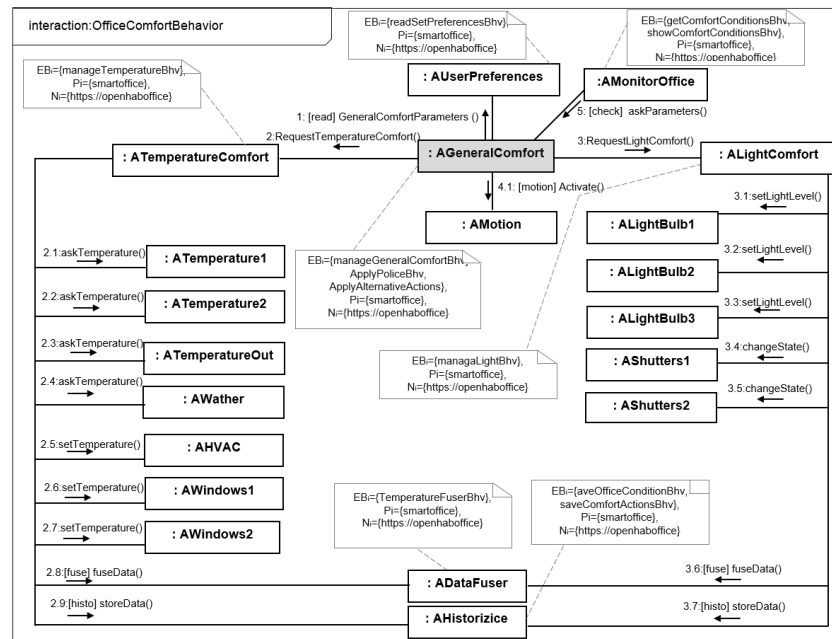


FIGURA 8.6: Relaciones entre los LOAs del ecosistema de IoA.

el esquema general que dirige el comportamiento global del sistema se resume en la Figura 8.6.

Aunque el diagrama de esta figura sigue las bases de un Diagrama de Comunicación UML, dejamos caminos abiertos que se pueden emplear para actualizar el modelo de agente de acuerdo a los cambios del ecosistema social que cada agente maneja. Esto añade dinamismo al modelo de agente, lo que implica que los LOAs pueden añadir nuevos agentes contrapartes para optimizar las tareas de colaboración y no limitarlas a cuestiones sociales estáticas muy concretas. Sin embargo, la adaptabilidad del sistema no sólo se consigue actualizando el ecosistema social de cada LOA, sino también re-configurando la estructura de cada agente. Para este cometido se lleva a cabo el borrado e inyección de nuevas tareas de agentes con el objetivo de adaptar su comportamiento según sea necesario.

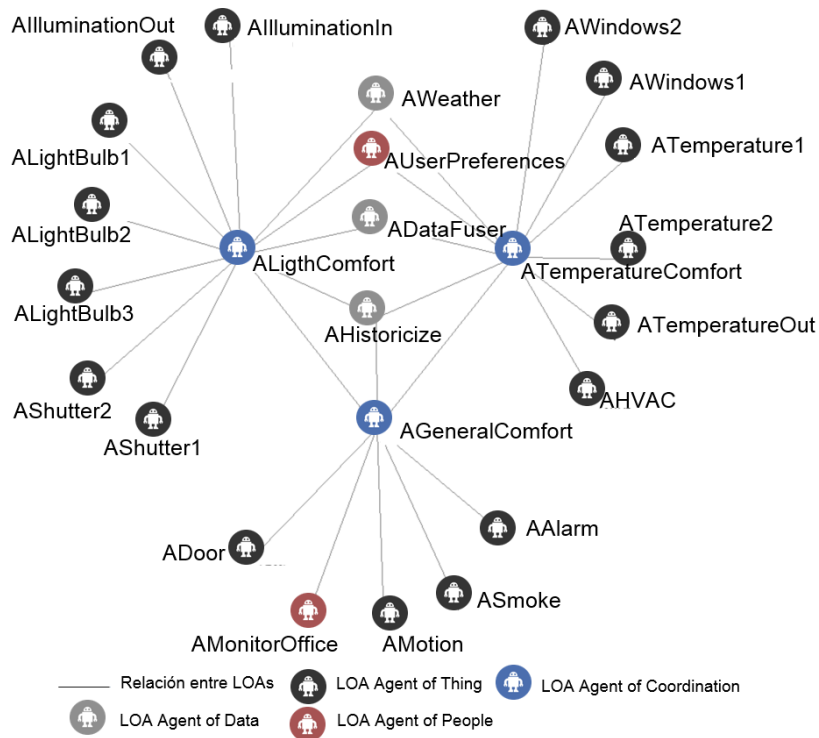


FIGURA 8.7: Sociograma del ecosistema de IoA.

8.4.4. Validación de los elementos del LOA (MA-10)

El objetivo del paso MA-10.1 para validar las credenciales de los agentes se realizó accediendo al repositorio central gestionado por el enfoque IoA, en el que estaban registrados todos los agentes pertenecientes a un ecosistema de este tipo. Este proceso se llevó a cabo con éxito.

Por otra parte, también se comprobó que todos los LOAs de tipo *Agents of Things* y *Agents of Coordination* tuvieran su correspondiente WAC vinculado a ellos. Cada agente tuvo su WAC asociado con él. Por lo tanto, la ejecución del paso MA-10.2 se llevó a cabo sin reportar ninguna novedad.

Además, en cuanto al paso MA-10.3, todas las relaciones que intervinieron en las redes sociales gestionadas por el sistema (*SmartComfort-BuildingMaracena*) fueron totalmente coherentes. Finalmente, el proceso de verificación de la integridad referencial de los LACs y sus URIs fue

verificado a través de la herramienta de Google especializada en validar datos enlazados (<https://search.google.com/structured-data/testing-tool>). De esta manera se cumplieron las especificaciones recomendadas en el paso MA-10.4. Como resultado, se obtuvo una colección de URIs consistentes y bien estructuradas.

8.4.5. Ejecución del proceso de modelado basado en contratos (MA-11)

La herramienta *IoA-Building Tool* nos permitió realizar la generación de los LACs. A partir de estos contratos se crearon con éxito y de forma automática los LOAs que formaban el ecosistema de IoA modelado. La aplicación fue creada en su totalidad y todos los agentes operaron de acuerdo con el comportamiento deseado; es decir, los agentes fueron capaces de comportarse de forma auto-adaptable, colaborativa e inteligente. El ecosistema de IoA resultante se comportó de acuerdo a los requerimientos funcionales modelados.

8.5. Evaluación cualitativa del sistema

8.5.1. Formulación del instrumento de evaluación

Con el fin de evaluar el sistema desarrollado mediante el uso del método sistemático propuesto en el Capítulo 6, se desarrolló un instrumento orientado a evaluar de forma cualitativa—en base a un conjunto de criterios—el nivel de funcionalidad, escalabilidad, adaptabilidad, interoperabilidad, usabilidad y confiabilidad de un ecosistema de IoT agenticado usando LOAs. Cada aspecto del instrumento está basado en la comprobación del cumplimiento de 22 criterios. Dichos criterios son incluidos en el cuestionario de evaluación listado en la Tabla 8.1.

En primer lugar, para evaluar el aspecto funcionalidad (F) fueron definidos seis criterios. Dichos criterios validan de manera empírica la función que desempeñan los LOAs colaborativos y no-colaborativos en el ecosistema de IoA y verificar el cumplimiento de sus objetivos en función de sus componentes (i.e., LAC y WAC) y tareas.

En segundo lugar, la evaluación del aspecto relacionado con la escalabilidad (S) implicó un proceso de validación similar al anterior; pero

enfocadas en verificar cuatro criterios elementales. Estos criterios verifican la factibilidad del modelo de IoA y de los LOAs para integrarse con otras tecnologías de IoT y de agentes.

En tercer lugar, la evaluación del aspecto asociado a la adaptación (A) requirió definir un total de tres criterios. Dichos criterios verifican si los LOAs son capaces de auto-adaptarse cuando sus contrapartes presentan fallas en la gestión de sus elementos o tareas, o inconsistencia en los datos ofrecidos. Estos criterios determinan en parte si las entidades LOA son operacionalmente fiables (R) en los ecosistemas de IoT e IoA.

En cuarto lugar, el aspecto relacionado con la interoperabilidad (I), incluye un total de tres criterios. Dichos aspectos validan los procesos de comunicación de los LOAs con objetos y agentes distribuidos en ecosistemas heterogéneos de IoT e IoA.

Finalmente, para evaluar el aspecto asociado a la usabilidad (U), se definieron cuatro criterios elementales. Dichos criterios están enfocados en verificar cuán usable es la herramienta *IoA-Building Tool* para la creación de ecosistemas de IoA; es decir, para la creación de los LOAs y sus elementos.

8.5.2. Aplicación del instrumento de evaluación propuesto

Con respecto al aspecto de la funcionalidad (F), la solicitud abarcó con éxito todos los criterios planteados. Los LOAs del ecosistema de IoA trabajaron de manera coordinada para lograr los objetivos colaborativos y no colaborativos definidos por cada agente. Además, las tres plataformas de IoA creadas interactuaron de forma apropiada entre sí, valiéndose de los servicios desplegados en la infraestructura *IoA Central Infrastructure*, para satisfacer las restricciones funcionales del escenario implementado.

Con respecto a la escalabilidad, los LOAs no experimentaron ningún comportamiento anómalo a lo planificado. Las plataformas de IoA permitieron la adición de nuevos LOAs en tiempo de ejecución y también fue posible ejecutar la terminación de LOAs existentes si fuera necesario. El hecho de que un agente contraparte cambiara su disponibilidad no implicó en ningún momento un error fatal para ninguno de los LOAs del ecosistema de IoA.

TABLA 8.1: Principales aspectos evaluados en sistemas IoA.

Aspecto	Criterio	Resultados
F	¿Cumplen los agentes sus objetivos (colaborativo, no colaborativo)?	Sí/Parcial/No
F	¿Ejecutan los agentes sus tareas (control, descubrimiento, comunicación)?	Sí/Parcial/No
F	¿Gestionan los LOAs su LAC consistentemente?	Sí/Parcial/No
F	¿Gestionan los LOAs su WAC consistentemente?	Sí/Parcial/No
F	¿Almacenan los LOAs su LAC en formato tripletas?	Sí/Parcial/No
F	¿Ejecutan los LOAs el proceso de descubrimiento de contrapartes?	Sí/Parcial/No
S	¿Permiten las plataformas de agentes la incorporación de nuevos LOAs?	Sí/Parcial/No
S	¿Soportan las plataformas de agentes la terminación de LOAs específicos?	Sí/Parcial/No
S	¿Soportan las plataformas IoA la integración de nuevas tecnologías de agentes?	Sí/Parcial/No
S	¿Pueden integrar los LOAs nuevas tecnologías de IoT?	Sí/Parcial/No
A	¿Pueden los LOAs no colaborativos llegar a ser colaborativos?	Sí/Parcial/No
A	¿Son los LOAs capaces de adaptarse a los cambios de su WAC?	Sí/Parcial/No
A	¿Son los LOAs capaces de adaptarse a los cambios de su entorno?	Sí/Parcial/No
I	¿Pudieron los LOAs comunicarse con agentes compatibles con FIPA-ALC?	Sí/Parcial/No
I	¿Se comunican los LOAs con agentes no LOAs externos?	Sí/Parcial/No
I	¿Soportan los LOAs comunicación con objetos de IoT externos?	Sí/Parcial/No
U	¿Qué nivel de dificultad se requiere para enriquecer semánticamente un LAC?	<Fácil/Normal/Complejo
U	¿Qué nivel de dificultad se requiere para crear un LAC?	Fácil/Normal/Complejo
U	¿Qué nivel de dificultad se requiere para crear un LOA?	Fácil/Normal/Complejo
U	¿Qué nivel de dificultad se requiere para crear una plataforma de LOAs?	Fácil/Normal/Complejo
R	¿Se recuperan los LOAs ante fallos de sus contrapartes?	Sí/Parcial/No
R	¿Se recuperan los LOAs ante fallos originados por sus objetos de IoT asociados?	Sí/Parcial/No

*F=Funcionalidad, S=Escalabilidad, A=Adaptabilidad, I=Interoperabilidad, U=Usabilidad y R=Confiableidad

En términos de adaptación (A) del sistema, los LOAs no sólo fueron adaptados para utilizar los datos proporcionados por los nuevos agentes, sino que también fueron capaces de auto-adaptarse cuando sus contrapartes reportaron inconsistencia o falta de datos de soporte. A partir de esta premisa, es evidente que los LOA fueron operacionalmente fiables (R), ya que pueden recuperarse de los cambios de su círculo social y de los objetos de IoT que ellos gestionan.

En cuanto a la interoperabilidad (I), los LOA modelados pudieron comunicarse tanto con objetos de IoT y agentes genéricos de JADE que adoptaban el estándar del FIPA a través de interfaces REST. No fue posible el control de objetos de IoT que no disponían de una interfaz de comunicación REST ni tampoco fue factible establecer comunicación con agentes externos que adoptaban un modelo de comunicación diferente a FIPA-ACL.

Finalmente, en términos de usabilidad (U) de la aplicación, la herramienta *IoA-Building Tool* fue bastante simple de usar ya que evitó que el desarrollador tuviera que realizar tareas técnicas complejas. Sin embargo, para mejorar el tiempo de enriquecimiento semántico, en el que el usuario participa, existe la posibilidad de mejorar la interfaz empleada incorporando componentes gráficos más amigables que los actuales.

En resumen, la aplicación implementada logró satisfactoriamente los requerimientos funcionales propuestos manteniendo un nivel aceptable de confiabilidad; así como la adaptación y recuperación de posibles fallas de los agentes homólogos en tiempo de ejecución. Cada uno de los LOAs que formaban parte del ecosistema IoA se comportó de acuerdo con lo planificado. Por otro lado, los aspectos evaluados sobre el sistema de comunidades inteligentes evidenciaron la viabilidad técnica del modelo teórico del IoA basado en LOAs aplicados en redes a pequeña escala.

8.6. Discusión

Los beneficios potenciales del métodos sistemático para el desarrollo de sistemas de IoT agentificados fue discutido en el Capítulo 6. La aplicación de este método sistemático para agentificar ecosistemas de IoT a pequeña escala (como es el caso particular de la oficina inteligente) permitió crear objetos dirigidos por LOAs capaces de comportarse de manera pro-activa, colaborativa e inteligente. Asimismo, el hecho de abordar la

creación de sistemas agentificados dividiendo el proceso en dos niveles (microscópico y macroscópico) facilitó la creación de los agentes (LOAs) que gobiernan los recursos de IoT. Además, el nivel macroscópico permitió que cambios futuros pudieran ser introducidos en el ecosistema de IoA sin modificar la estructura del sistema ni de los LOAs.

Por otro lado, el método sistemático es un primer acercamiento a las herramientas de la Ingeniería de Software para el desarrollo de sistemas inteligentes de IoT. Por ello, el método sistemático propuesto y aplicado en un sistema de uso cotidiano plantea retos enfocados en el desarrollo de una metodología especializada que soporte el ciclo de vida completo para la creación de sistemas agentificados de IoT a escala pequeña, mediana y extensa.

8.7. Contribuciones relacionadas

El caso de estudio descrito en este apartado ha sido evaluado por un comité científico. Los detalles del manuscrito en el que consta este caso de estudio se describen a continuación:

P. Pico-Valencia, J.A. Holgado-Terriza and P. Padereski. A Systematic Method for Building Internet of Agents Applications Based on the Linked Open Data Approach. *Future Generation Computer Systems Journal*, 2017. [*En Revisión, Segunda Vuelta*].

Capítulo 9

Conclusiones y trabajos científicos futuros

“El investigador sufre las decepciones, los largos meses pasados en una dirección equivocada, los fracasos. Pero los fracasos son también útiles, porque, bien analizados, pueden conducir al éxito. Y para el investigador no existe alegría comparable a la de un descubrimiento, por pequeño que sea...”—Alexander Fleming.

En este capítulo se presenta un resumen de los principales resultados que se han obtenido en esta investigación, así como los trabajos futuros que podrían dar continuidad a esta investigación y que pueden ser abordados para fortalecer los mecanismos y modelos propuestos. Finalmente, se listan las contribuciones científicas ya publicadas y aquellas que se encuentran en proceso de revisión que fueron producto del trabajo descrito en esta memoria.

9.1. Conclusiones

Los resultados de esta investigación demuestran que el Internet de los Agentes (IoA) puede constituirse como un nuevo paradigma que puede contribuir a dotar de inteligencia, proactividad, adaptabilidad, sociabilidad y capacidad de colaboración al IoT y, por consiguiente, al Futuro Internet.

A lo largo de esta investigación hemos fijado las bases de lo que entendemos debe ser el IoA, y hemos desarrollado tanto un marco de trabajo para la creación de una plataforma de agentes LOA sobre ecosistemas IoA, así como un método sistemático que determina cómo pueden construirse dicha plataforma conforme a un proceso de agentificación.

En la tesis se ha presentado cómo abordar el proceso de agentificación del IoT para agentes que llevan a cabo la gestión de los recursos de redes de IoT a través de sistemas multiagentes. sin embargo, dicho proceso se podría aplicar embebidos en objetos IoT, aprovechando la misma ontología, o extendiéndola si fuera necesario.

- Se ha planteado un nuevo paradigma denominado Internet de los Agentes (IoA) como la próxima generación del IoT. Para describir las directrices de este paradigma se ha creado un marco de trabajo especializado que ayuda a crear ecosistemas de IoT más inteligentes, pro-activos y colaborativos a través de un proceso de integración con agentes (agentificación).
- Se determinaron cuáles son los principales modelos usados para modelar procesos inteligentes en escenarios de IoT agentificados. Aunque se constató que el uso de reglas, componentes deliberativos, lógica semántica, modelos probabilísticos y mecanismos de Inteligencia Artificial son los más usados, se pudo confirmar que por su mínima exigencia de recursos de cómputo, el modelo basado en reglas ha sido el más aplicado en soluciones reales.
- Se ha definido un vocabulario común que los agentes de ecosistemas de IoA pueden utilizar para ejecutar procesos de razonamiento semántico. Dicho vocabulario fue formalizado mediante una ontología denominada IoA-OWL.
- Se ha planteado un novedoso modelo de agente semántico (denominado LOA) para agentificar objetos de IoT, el cual es capaz de

ejecutar razonamiento semántico basado en la reutilización de descriptores almacenados como datos abiertos enlazados.

- Se ha replanteado el concepto de contrato en el dominio de agente para que éste guarde un símil con el contrato de servicios usado en SOA. De un contrato comúnmente orientado a ser el soporte de tareas de negociación entre agentes, se ha pasado a concebir una unidad más bien descriptiva y rica semánticamente hablando, la cual ayuda a mejorar la interoperabilidad semántica entre agentes de distintas plataformas distribuidas.
- Se ha creado un modelo de contrato semántico (denominado LAC) para describir agentes que operan en ecosistemas de IoA, el cual sigue la especificación de la ontología IoA-OWL. Esto habilita a los agentes para ejecutar procesos de razonamiento y de descubrimiento semántico más preciso y exacto que los usados por modelos de agentes actuales aplicados en IoT.
- Se creó un modelo de flujo de control de agente (denominado WAC) basado en micro- y macro-recetas IFTTT fácil de definir por los usuarios y simple de actualizar automáticamente por los propios agentes que operan en ecosistemas de IoA.
- Se ha formulado un mecanismo automático de descubrimiento de agentes contrapartes en ecosistemas de IoA el cual se basa en una colección de descriptores semánticos que los agentes describen en su contrato. Esto habilita a los agentes para que sean capaces de recuperar los agentes contrapartes más idóneos y de manera más precisa que cualquier otro método de descubrimiento que opera solo basado en la sintaxis.
- Se ha introducido un mecanismo de tolerancia ante fallos para que los agentes sean capaces de detectar anomalías en su flujo de trabajo de manera pro-activa, y reemplazar aquellos agentes que hayan producido fallos con agentes análogos funcionales que ofrezcan una mejor calidad de servicio. De esta manera se asegura un mejor aprovechamiento de los recursos disponibles en IoT y un aseguramiento de la disponibilidad de los agentes en IoA.

- Se ha creado un mecanismo especializado en la inyección de tareas de agentes en tiempo de ejecución. Esto permite que los agentes puedan incorporar tareas adicionales a las tareas genéricas definidas por los agentes de tipo LOA y que así los agentes sean funcionalmente escalables.
- Se ha formulado un método de diseño que describe de forma sistemática el proceso global en: fases, etapas y pasos distribuidos en dos niveles, microscópico y macroscópico. Esto ayuda para que los desarrolladores de ecosistemas agentificados de IoT puedan saber cómo emplear el modelo y la arquitectura de referencia propuesta en términos prácticos.
- Se ha desarrollado una herramienta especializada, denominada *IoA-Building Tool*, que ayuda a automatizar el proceso de agentificación de los objetos de IoT a través de entidades LOAs que operan en sistemas multiagentes desarrollados en Java.
- Se ha desarrollado una API en Java que abstrae la complejidad inherente al consumo de servicios heterogéneos y que además provee un método uniforme para que los agentes software puedan modelar procesos complejos sin que el modelo de servicio usado sea un factor que lo limite.
- Se ha aplicado el modelo de agente, la arquitectura de referencia y el método sistemático propuesto en el modelado de dos ecosistemas de Inteligencia Ambiental (AmI) para validar los planteamientos propuestos y demostrar que las herramientas desarrolladas cumplen su objetivo en casos prácticos.
- Se ha propuesto un catálogo de posibles aplicaciones potenciales que los ecosistemas de IoA pueden brindar en escenarios reales y que los desarrolladores pueden tomar como base para idear ecosistemas de IoT más inteligentes usando el proceso de agentificación del IoT.
- Finalmente, se han planteado los principales retos y tendencias que el IoA debe afianzar para llegar a convertirse en un paradigma consolidado entre la comunidad científica y de desarrollo.

9.2. Trabajos futuros

El modelo arquitectónico y herramienta metodológica abordados en esta investigación y descritas en esta memoria, abren nuevos caminos que pueden permitir su mejora en el futuro. A continuación, se describen algunas de las líneas de investigación prometedoras en el campo del IoA, un área prometedora en torno al Futuro Internet a corto, mediado y largo plazo, que ha sido la temática estudiada en este trabajo:

Diseño de hardware para el soporte de agentificación basada en agentes embebidos en los propios objetos de IoT.

- Diseñar objetos sofisticados de IoT con suficiente capacidad de procesamiento y almacenamiento que soporten el proceso de agentificación y la correspondiente ejecución de procesos de razonamiento semántico, selección y recuperación de contrapartes desde el propio objeto de IoT.
- Crear puentes (*gateways*) de IoT con soporte del proceso de agentificación que coordinen los objetos genéricos de IoT y entidades de naturaleza *agent of things* que operen en ecosistemas heterogéneos de IoA.

Herramientas de Ingeniería de Software para el apoyo del proceso de agentificación del IoT.

- Crear una metodología formal que contemple el ciclo de vida del software completo para llevar a cabo el análisis, diseño, desarrollo, implementación y mantenimiento de sistemas de IoT basados en el proceso de agentificación aplicando sistemas multiagentes y agentes embebidos en los propios dispositivos de IoT.
- Desarrollar una herramienta CASE (*Computer Aided Software Engineering*) que permita automatizar y validar el proceso completo de desarrollo de sistemas de IoT agentificados: (i) enriquecimiento semántico de LACs usando interfaces gráficas accesibles y usables por parte de los usuarios, (ii) composición de flujos de trabajo especializados para la gestión de los recursos de ecosistemas de IoA.

Ontologías para describir el IoA y tecnologías semánticas para gestionar el conocimiento semántico.

- Crear un mecanismo para la federación de grafos RDF que permita distribuir el contenido de los LACs en varias plataformas.
- Extender la ontología IoA-OWL de manera que se introduzcan componentes ontológicos adicionales y descriptores semánticos relevantes en ecosistemas de IoA.
- Publicar la ontología IoA-OWL en la plataforma Linked Open Data para que adquiera mayor relevancia entre la comunidad de desarrollo.

Creación de mecanismos especializados para optimizar los procesos ejecutados por los LOAs.

- Crear operadores especializados para el descubrimiento de agentes contrapartes en ecosistemas de IoA.
- Crear operadores especializados para realizar la selección de los agentes contrapartes más adecuados, recuperados en ecosistemas de IoA
- Crear una herramienta que permita crear LOAs a partir de otras herramientas populares para el desarrollo de agentes distinta de JADE tales como: Jadex, JADE LEAP, Jason, entre otros.

9.3. Contribuciones científicas derivadas

9.3.1. Revistas indexadas con impacto JCR

N. de contribución: 1

P. Pico-Valencia, J.A. Holgado-Terriza and José Senso, *Towards an Internet of Agents Model Based on Linked Open Data Approach*. Autonomous Agents and Multi-Agent Systems Journal, ISSN 1387-2532 (2017) (En Revisión - Segunda Ronda).

Factor de Impacto Scimago (SJR 2017) 0.43 - Q2.
Factor de Impacto ISI (JCR) 1.176 - Q3.

Contribución del manuscrito: En este artículo se presentan las directrices asociadas al enfoque del IoA desde una perspectiva semántica en donde los ecosistemas de IoT basados en agentes son modelados a través de agentes del tipo *Open Linked Agent* y estas unidades, a la vez son descritas mediante el uso de contratos denominados *Linked Agent Contracts*. Se describe detalladamente cada sección contractual asociada al contrato y se aplica el modelo para diseñar una aplicación que provea confort en un hogar inteligente.

N. de contribución: 2

P. Pico-Valencia, J.A. Holgado-Terriza and Patricia Padereswi, *A Systematic Method for Building Internet of Agents Applications Based on the Linked Open Data Approach*. Future Generation Computer Systems Journal, ISSN 0167-739X (2017) (En Revisión, 2da Ronda).
Factor de Impacto Scimago (SJR 2017) 0.84 - Q1.
Factor de Impacto ISI (JCR) 4.639 - Q1.

Contribución del manuscrito: En este artículo se presente un método sistemático para modelar aplicaciones basadas en el enfoque del IoA. Dicho método adopta los principios de la metodología ágil SCRUM y divide el proceso de modelado desde dos niveles, microscópico (a nivel de agentes asociados a objetos de IoT) y macroscópico (a nivel de interacción de agentes). El método es aplicado en una aplicación para proveer confort en una oficina inteligente.

N. de contribución: 3

P. Pico-Valencia, and J.A. Holgado-Terriza, *Agentification of the Internet of Things: A Systematic Literature Review*, International Journal of Distributed Sensor Networks, ISSN 1550-1477 (2018). DOI:[10.1177/1550147718805945](https://doi.org/10.1177/1550147718805945). (En Proceso de Publicación)
Factor de Impacto Scimago (SJR 2017) 0.26 - Q2.

Factor de Impacto ISI (JCR) 1.787 - Q2.

Contribución del manuscrito: Este estudio presenta una visión general de los principales modelos, métodos, mecanismos, y herramientas de desarrollo y tecnológicas propuestas hasta la fecha para crear objetos y redes inteligentes de IoT a través de su integración con agentes de software —denominado también como agentificación del IoT—; determinando así, su aporte en términos de inteligencia y autonomía.

N. de contribución: 4

P. Pico-Valencia, J.A. Holgado-Terriza, D. Herrera-Sánchez and J. Sampietro, *Towards the Internet of Agents: An Analysis of the Internet of Things from the Intelligence and Autonomy Perspective*, 2018, Revista Ingeniería e Investigación, Colombia, 38 (1). DOI: [10.15446/ing.investig.v38n1.65638](https://doi.org/10.15446/ing.investig.v38n1.65638).

Factor de Impacto Scimago (SJR 2017) 0.19 - Q3.

Factor de Impacto ISI (JCR) 0.455 - Q4.

Contribución del manuscrito: En este trabajo se presenta un análisis de los principales beneficios derivados del enfoque IoA, visto desde las actuales necesidades que el ser humano demanda en su trabajo y vida cotidiana—industrial, ciudad inteligente, salud y bienestar—, las cuales pueden ser resueltas por redes de IoT modeladas mediante el uso de agentes software.

N. de contribución: 5

P. Pico-Valencia, J.A. Holgado-Terriza and X. Quiñónez-Ku, *Future Internet and the Next Generation of the Internet of Things: A Brief Survey*. IEEE Internet of Things Journal, ISSN 23274662 (2018) (En Revisión, 1ra Ronda).

Factor de Impacto Scimago (SJR 2017) 1.34 - Q1.

Factor de Impacto ISI (JCR) 5.863 - Q1.

Contribución del manuscrito: En este trabajo, presentamos una revisión exhaustiva en la que se esbozan los principales paradigmas asociados con la Internet del Futuro —IoT, la Internet de los Servicios (IoS),

la Internet de las Personas (IoP), la Internet de los Contenidos (IoC) y la Internet de los Agentes (IoA)—. Analizamos estos paradigmas en términos de su definición, alcance, objetivo y correlación directa con las tecnologías de IoT. Además, concebimos la Internet del Futuro como un paradigma que logrará un mundo inteligente conectado a través del uso del paradigma del IoA. Este paradigma emergente, considerado como la próxima generación del IoT, propone la agentificación del IoT para lograr ecosistemas inteligentes, proactivos y colaborativos tanto de dispositivos como de servicios a través de Internet. El proceso de agentificación, así como sus líneas de investigación, también son analizados en este trabajo.

N. de contribución: 6

P. Pico-Valencia, J.A. Holgado-Terriza and X. Quiñónez-Ku, *An Agent of Thing Model for IoT-objects Based on If-This Then-That Recipes*. IEEE Internet Computing, ISSN 10897801 (2018) (En Revisión, 1ra Ronda).

Factor de Impacto Scimago (SJR 2017) 0.38 - Q2.

Factor de Impacto ISI (JCR) 1.929 - Q1.

Contribución del manuscrito: El objetivo de este trabajo es desarrollar un modelo de agente personalizable orientado a objetos de IoT cuyo comportamiento de control está impulsado por un flujo de trabajo estructurado compuesto tanto por micro y macro-recebetas If-This Then-That (IFTTT). Para desarrollar el modelo de agente propuesto, se empleó una extensión del Java Agent DEvelopment Framework (JADE). Del mismo modo, en lo que respecta a los objetos de IoT, se construyeron a partir del marco de openHAB. Como resultado, se logró un modelo de flujo de trabajo fácil de personalizar que los agentes pueden utilizar para controlar los ecosistemas de IoA.

9.3.2. Revistas indexadas sin impacto JCR**N. de contribución: 7**

P. Pico-Valencia, *Managing the Heterogeneity in Multiagent Systems*

Based on Service Oriented Technologies. Revista Computación y Sistemas, México, 2017, ISSN 2007-9737. (En Revisión, 1ra Ronda).

Contribución del manuscrito: En este trabajo se presenta un análisis de los principales enfoques y entidades de emergencia originados en la integración de Sistemas Multiagente (MASs) y Arquitectura Orientada a Servicios (SOA), denominados MAS-SOA. Además, se describe y evalúa un método sistemático para el desarrollo de aplicaciones MAS-SOA compatibles con ecosistemas de servicios heterogéneos.

9.3.3. Publicaciones en congresos internacionales

N. de contribución: 8

P. Pico-Valencia and J. A. Holgado-Terriza, *Integration of Multi-Agent Systems with Resource-Oriented Architecture for Management of IoT-objects*. In: Seventh International Workshop on the Reliability of Intelligent Environments (WoRIE 2018), Rome, Italy, Vol. 23, 2018, pp. pp. 567-576, ISBN 978-1-61499-873-0. DOI: [10.3233/978-1-61499-874-7-567](https://doi.org/10.3233/978-1-61499-874-7-567).

Contribución del manuscrito: Este manuscrito propone una arquitectura de agentes de Sistemas Multiagente basada en Arquitectura Orientada a Recursos (MAS-ROA), donde el comportamiento de cada agente es impulsado por un flujo de trabajo de control específico a partir del cual los agentes realizar acciones de detección y control sobre objetos de IoT mediante procesos de colaboración. Un sistema de Inteligencia ambiental, implementado bajo el enfoque MAS-ROA fue contrastado con una implementación basada en la Arquitectura Orientada a Servicios (MAS-SOA) dirigida por el Perfil de Dispositivos para Servicios Web (DPWS), a fin de obtener información sobre las diferencias en capacidades y rendimiento.

N. de contribución: 9

P. Pico-Valencia and J. A. Holgado-Terriza, *Semantic Agent Contracts for Internet of Agents*. In: First International Workshop on

the Internet of Agents, IEEE/WIC/ACM International Conference on Web Intelligence, 2016, Omaha, USA, pp. 76-79, ISBN 978-1-5090-4771-0. DOI: [10.1109/WIW.2016.033](https://doi.org/10.1109/WIW.2016.033).

Contribución del manuscrito: Este artículo presenta una ontología para describir contratos semánticos de agentes que operan en entornos de IoA. Dicha ontología propuesta, denominada IoA-OWL, incluye los principales conceptos para describir agentes en términos de los siguientes aspectos: propiedades no funcionales de agentes, información contextual donde el agente opera, círculos sociales de colaboración, ecosistemas de servicios usados para ejecutar acciones, objetos de IoT y recursos asociados, y finalmente, artefactos de agentes.

N. de contribución: 10

Pablo A. Pico-Valencia and Juan A. Holgado-Terriza, *An Agent Middleware for Supporting Ecosystems of Heterogeneous Web Services*, Procedia Computer Science, Montreal, Canada, Volume 94, 2016, pp. 121-128, ISSN 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2016.08.020>.

Contribución del manuscrito: En este artículo se muestra el modelado de un Sistema Multiagente basado en arquitecturas SOA. Los agentes pueden consumir servicios web heterogéneos usando tres complementos especializados en la invocación de servicios SOAP, RESTful, DPWS y DOHA. Se evaluaron comportamientos orientados a proveer confort en una casa inteligente.

N. de contribución: 11

Pico-Valencia Pablo and Holgado-Terriza J.A., *A Framework for the Development of Smart Ubiquitous Real-Time Systems Based on the Internet of Agents and Internet of Services Approaches*, In: 5th International Workshop on the Reliability of Intelligent Environments, United Kingdom, London, Vol. 21, 2016, pp. 76 - 85, ISBN 978-1-61499-689-7. DOI: [10.3233/978-1-61499-690-3-76](https://doi.org/10.3233/978-1-61499-690-3-76).

Contribución del manuscrito: Este artículo presenta un marco de trabajo conceptual para el desarrollo de aplicaciones de tiempo real

basadas en los paradigmas del Internet de los Servicios e Internet de los Agentes. El marco de trabajo propuesto incluye seis dimensiones tales como interacción, ambiente, planificación, organización, y normativo. Dichas dimensiones han sido organizadas de manera que permiten modelar los principales aspectos gestionados actualmente en escenarios de IoT.

9.3.4. Publicaciones en congresos nacionales

N. de contribución: 12

Pablo A. Pico-Valencia and Juan A. Holgado-Terriza, *ADELE: A Middleware for Supporting the Evolution of Multi-agents Systems Based on a Metaprogramming Approach*, In: Trends in Practical Applications of Scalable Multi-Agent Systems, Seville, Spain, 2016, pp. 297-310, ISBN 978-3-319-40159-1. DOI: https://doi.org/10.1007/978-3-319-40159-1_25

Contribución del manuscrito: Este artículo presenta un middleware (denominado *Agent Dynamic EvoLutionary at runtimE*) compatible con el modelo de agente reactivo con soporte de evolución basado en un modelo normativo dinámico dirigido por normas. Los agentes tienen la capacidad de modificar su comportamiento en tiempo de ejecución aplicando cambios en el modelo normativo, lo cual conlleva a ejecutar un proceso de inyección de comportamientos aplicando técnicas de metaprogramación tal como la compilación *on-the-fly*.

N. de contribución: 13

Pablo A. Pico-Valencia. *Business intelligence applied to monitoring and meta-monitoring scenarios*, In: 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Las Palmas, Spain, 2016, pp. 1-6. DOI: [10.1109/CISTI.2016.7521536](https://doi.org/10.1109/CISTI.2016.7521536)

Contribución del manuscrito: Este artículo presenta cómo usar el proceso de inteligencia de negocios y algunas de sus tecnologías para llevar a cabo acciones de monitorización y meta-monitorización de los datos detectados y los componentes físicos que componen una plataforma

de ambiental. Para el desarrollo de la aplicación se utilizó la suite de Pentaho y como resultado se obtuvo una herramienta con una interfaz amigable, y con capacidades para filtrar y generar automáticamente tablas de resumen y gráficos estadísticos, basados en los datos de una infraestructura física de monitoración ambiental.

9.3.5. Publicaciones en preparación

N. de contribución: 14

P. Pico-Valencia and J.A. Holgado-Terriza, *A Semantic, Proactive and Open Source Smart-Object for the Internet of Things Agentification*.

Contribución del manuscrito: Este trabajo se presenta un modelo de objeto *open hardware* que permite realizar la agentificación del IoT basado en el embebido de agente dentro de la arquitectura del propio objeto. A partir de dicho artificio los objetos de IoT fueron capaces de extender sus capacidades para actuar con mayor de forma proactiva, y con mayor nivel de autonomía, inteligencia y colaboración. El desarrollo del modelo propuesto se basa en componentes Single Board Computer (Raspberry PI) como componentes hardware y agentes JADE basados en datos enlazados —que habilitan al objeto de capacidades semánticas para recuperar objetos contrapartes y recursos de IoT con mayor precisión—con soporte de reglas IFTTT. Como prueba de concepto, se creó una útil bombilla inteligente en Ambient Intelligence (AmI) que fue validada en una red de IoT.

Anexo A. Fundamentos sobre tecnologías de agentes

Paradigma orientado a agentes

El Paradigma Orientado a Agentes (POA), nace en los años 80 por requerimientos en el área de la Robótica. Investigadores de la época que trabajaban en éste campo, ahondaban esfuerzos para construir autómatas físicos mediante el uso de técnicas clásicas de la Inteligencia Artificial; pero notaban deficiencias en lo relacionado a la inflexibilidad y la carencia de reacción en tiempo real. Es así como un enfoque nuevo y radical basado en agentes [268], capaz de razonar sobre modelos simbólicos externos le permitió planificar las acciones a ejecutar sobre un entorno dinámico para el logro de objetivos delimitados.

Aunque el POA es un enfoque en donde el agente (agente simple) juega un papel importante; su funcionamiento se optimiza introduciendo el trabajo colaborativo de varios de ellos (multiagentes) [268]. Por ello, los Sistemas Multiagentes han despertado interés entre los investigadores del área de desarrollo de software distribuido como ecosistemas de IoA.

Agentes simples

En la Literatura no existe una única definición que describa el término agente en el ámbito de la Computación. Cada investigador, dependiendo del trabajo que esté abordando se ve influenciado por las necesidades en cuanto a características que el agente debe tener y las aplicaciones que le pueda dar [200].

Por un lado, se define a un agente como *“cualquier cosa que puede percibir de su entorno a través de sensores y actuar mediante actuadores”* [227]. Esta definición es una de las más aceptadas y por tanto se la ha tomado como referencia en conjunto con los planteamientos propuestos por Maes [170], y

Franklin y Graesser [101], para definir a un agente como *un componente evolutivo de software que se adapta a un entorno y actúa para cumplir objetivos, ayudado de un estado, un comportamiento y mecanismos de comunicación.*

Las características que incorpora un agente dependen principalmente de la misión que éste tenga y de la aplicación que se le esté dando. Entonces de manera general un agente está definido por características específicas (a-e) y características adicionales (f-i) que pueden ser introducidas según las necesidades y el tipo de sistema modelado:

- a. *Reactividad.* Responder ante cambios en el entorno.
- b. *Autonomía.* Ejercer control sobre las propias acciones ejecutadas sobre ecosistemas con la mínima intervención humana.
- c. *Proactividad.* No actuar solamente en respuesta al entorno; sino que tenga la capacidad de anticiparse a eventos futuros que puedan originarse.
- d. *Persistencia.* Ejecutar procesos de manera continua e ininterrumpida.
- e. *Social.* Establecer comunicación e interacción con otros agentes que se ejecutan en plataformas externas.
- f. *Adaptación.* Cambiar un comportamiento genérico en función de experiencias previas o basados en casos.
- g. *Movilidad.* Desplazamiento de una máquina a otra para apoyar en la ejecución de tareas en entornos con recursos limitados.
- h. *Inteligencia.* Ejecución de acciones en función de conocimiento adquirido o procesos de razonamiento automático.
- i. *Colaboración.* Formar parte de organizaciones dentro de sociedades que siguen estructuras de comportamiento similares.

Multiagentes

A un alto nivel de abstracción, un MAS puede ser concebido como: a) *un conjunto de agentes individuales que se comunican*, b) *un espacio de interacción con agentes individuales que interactúan unos con otros* o c) *una organización que regula el comportamiento de agentes individuales.*

Por otro lado, a un nivel de abstracción más bajo, un MAS puede ser visto como *un conjunto de agentes que trabajan de forma colaborativa para alcanzar un objetivo en común* [268]. Para cumplir este objetivo, los agentes deben tener la capacidad de utilizar un estándar de comunicación. En este sentido el lenguaje *Agent Communication Language*, propuesto por FIPA, constituye una primera aproximación para alcanzar la interoperabilidad entre sistemas

multiagentes. Una representación de sistemas multiagentes es ilustrado en la Figura A.1.

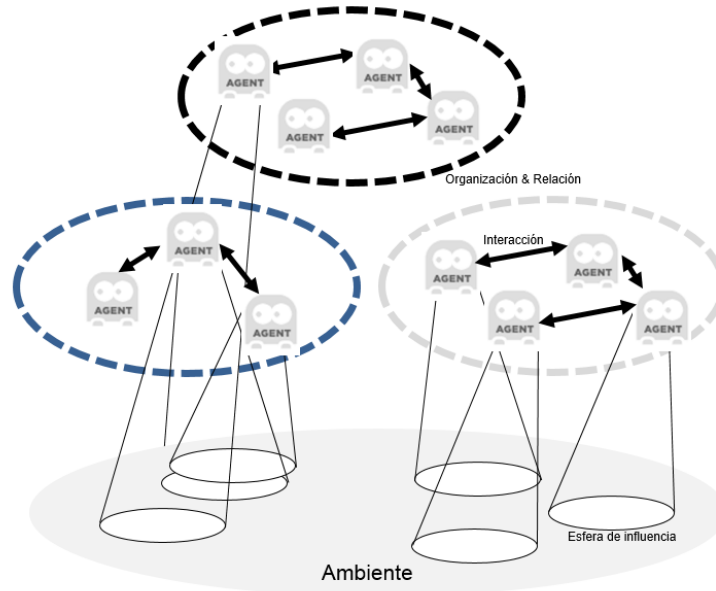


FIGURA A.1: Esquema de un sistema multiagente basado en la definición de Wooldridge.

Arquitecturas de agentes

En términos generales la arquitectura de un agente hace referencia a su estructura interna [170]. Por tanto la arquitectura especifica los componentes que forman los agentes y describe la forma cómo interactúan unos con otros [68]. Se pueden distinguir generalmente tres tipos de arquitecturas de agentes: reactiva, deliberativa e híbrida [271].

Arquitectura reactiva

La arquitectura reactiva, detallada en la Figura A.2.a, se basa en un enfoque conductista que funciona según las especificaciones de un modelo estímulo-respuesta [271]. Según éste modelo, un agente reacciona ante los cambios suscitados en el entorno que le rodea; es decir, que la activación de patrones

sensoriales bajo ciertas condiciones repercute directamente sobre los actuadores.

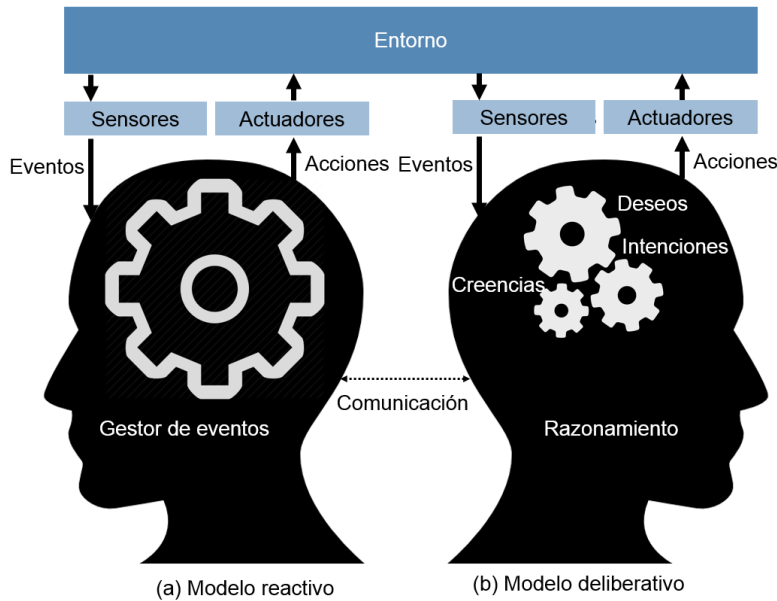


FIGURA A.2: Arquitectura de agente (a) reactiva y (b) deliberativa.

Ésta arquitectura ha sido la base de algunos modelos computacionales, entre los que figuran: *reglas situadas* (basados en reglas del tipo IF situación THEN acción(es)), *arquitecturas de subsunción* (capas que ejecutan una determinada conducta y cada una de ellas es la de una red de topología de máquina de estados finitas) y *tareas competitivas* (decisión sobre qué tarea realizar de entre varias posibles) [207].

Algunas de las herramientas que permiten construir este tipo de agentes son las siguientes: MASON (Multi-Agent Simulator Of Neighborhoods/Networks), Agent Factory, MAPS (Mobile Agent Platform for Sun SPOTs), SARL, 2APL (A Practical Agent Programming Language), Agent Factory Framework [147, 157]. Sin embargo, es importante resaltar que la herramienta mayormente empleada en el desarrollo de sistemas reales ha sido JADE, por sus siglas en inglés Java Agent DEvelopment Framework [28].

Arquitectura deliberativa

Una arquitectura deliberativa, ilustrada en la Figura A.2.b, aplica el paradigma de la Inteligencia Artificial simbólica. Esto significa que el comportamiento y el ambiente se expresa en términos de conocimiento representado a través de símbolos. Los agentes diseñados a partir de ésta arquitectura, tienen asociado a ellos un estado mental que les permite tomar decisiones sobre el entorno donde se encuentren, empleando mecanismos deductivos y siguiendo tres actitudes mentales básicas como son creencias, deseos e intenciones. De allí el nombre BDI.

Las *creencias* constituyen la información interiorizada que tiene el agente de sí mismo y del entorno que le rodea. Los *deseos* son todos los posibles estados u objetivos que un agente puede cumplir. Un típico caso de deseo son las metas que se ha planteado lograr un agente [75]. Finalmente, en vista de que un agente no puede cumplir con todos los objetivos a la vez, se introduce el concepto de *intención*, que son los objetivos que el agente intenta cumplir en cada momento. Generalmente, a las instancias de las intenciones se las denomina planes —un concepto que también se usa en arquitecturas BDI.

De manera práctica, algunas de las principales herramientas que permiten el desarrollo de agentes de esta naturaleza son las siguientes: *AgentSpeak(L)*, *JACK Intelligent Agents*, *SRI Procedural Agent Realization Kit (SPARK)*, *JADEX* y *2APL* [147, 157].

Agentes orientados a IoA

La arquitectura de agente reactivo propuesta por Russell y Norvig [227] puede emplearse para modelar entidades que ejecutan tareas programadas o repetitivas basadas en eventos desencadenados por un sistema. Este modelo puede resultar una buena alternativa cuando los recursos de computación no son sofisticados o cuando la entidad es más sencilla de modelar. Por otra parte, la arquitectura reactiva representa la mejor opción cuando se requiere un alto nivel de interacción como es el caso de los ecosistemas de IoA. Por lo tanto, esta arquitectura de agente es útil cuando se deben modelar tareas colaborativas.

Al contrario de la arquitectura reactiva, la arquitectura del agente deliberativo modelada a partir del uso de artefactos BDI [41] representa una mejor opción cuando la inteligencia se gestiona a nivel de un agente individual. Esto se debe a que “*el modelo BDI tradicional no proporciona ninguna especificación de la comunicación con el agente (o cualquier otro aspecto a nivel social)*”

[26]. Además, esta arquitectura se caracteriza por su complejidad inherente, los altos requerimientos de computación de potencia y los grandes esfuerzos que los desarrolladores deben dedicar para crear este tipo de agentes. Por lo tanto, se sugiere la arquitectura deliberativa en los casos en que el agente debe actuar de forma independiente con un bajo nivel de interacción. Pero este no es el caso. Los ecosistemas de IoA demandan un alto nivel de interacción.

Frente a las bases conceptuales de las arquitecturas reactivas y deliberativas, recomendamos el uso de uno u otro modelo para implementar el entorno de ejecución del agente, teniendo en cuenta las capacidades sociales, la disponibilidad de recursos de cómputo donde se ejecutará el agente, la complejidad del escenario y su nivel de dinamismo y, finalmente, el nivel de inteligencia requerido por el sistema desarrollado. Debido a que es muy complejo coordinar un alto número de agentes deliberativos como lo exige el IoA, las iniciativas propuestas en esta tesis están específicamente dirigidas a agentes reactivos. Sin embargo, esta es solo una recomendación y si el equipo de desarrollo cree que el uso de agentes deliberativos o híbridos es el más apropiado para el desarrollo de su sistema, el modelo también puede ser extendido para su uso en sistemas centralizados de IoA.

Contribuciones relacionadas

Los fundamentos conceptuales planteados en este anexo forman parte del trabajo de fin de máster que se realizó como parte de las actividades de formación realizada en el Programa de Doctorado. Los detalles del informe de dicho trabajo son descritos a continuación:

P. Pico-Valencia. *Middleware para el Desarrollo de Arquitectura Inteligente Orientada a Servicios Ubicuos*, 2015. Universidad de Granada, Departamento de Lenguajes y Sistemas Informáticos.

Anexo B. Fuentes de información

TABLA B.1: Fuentes primarias de información.

#	Fuente	Cadena	Año	Tipo de estudio	Referencia
E1	Scopus	4	2015	Conferencia	[213]
E2	IEEE	1	2013	Conferencia	[83]
E3	S. Direct	1	2016	Revista	[235]
E4	ACM	4	2016	Conferencia	[258]
E5	Scopus	3	2017	Conferencia	[234]
E6	Scopus	4	2015	Conferencia	[81]
E7	IEEE	1	2017	Revista	[104]
E8	IEEE	1	2013	Conferencia	[185]
E9	IEEE	1	2015	Conferencia	[191]
E10	S. Direct	1	2013	Revista	[265]
E11	IEEE	1	2014	Conferencia	[160]
E12	WoS	1	2017	Revista	[60]
E13	S. Direct	1	2013	Conferencia	[189]
E14	Scopus	4	2014	Revista	[164]
E15	IEEE	1	2013	Conferencia	[278]
E16	Scopus	4	2015	Conferencia	[39]
E17	IEEE	1	2014	Conferencia	[172]
E18	IEEE	4	2012	Conferencia	[284]
E19	IEEE	4	2016	Revista	[232]
E20	WoS	2	2015	Conferencia	[7]
E21	Scopus	4	2016	Conferencia	[143]
E22	IEEE	1	2015	Conferencia	[14]
E23	IEEE	1	2017	Conferencia	[3]
E24	IEEE	1	2016	Conferencia	[52]
E25	IEEE	4	2014	Conferencia	[201]

Fuentes primarias de información. Parte II

#	Fuente	Cadena	Año	Tipo de estudio	Referencia
E26	ACM	1	2015	Revista	[11]
E27	WoS	3	2013	Patente	[31]
E28	IEEE	1	2013	Conferencia	[161]
E29	IEEE	1	2013	Conferencia	[132]
E30	Scopus	4	2014	Conferencia	[160]
E31	IEEE	1	2016	Conferencia	[39]
E32	IEEE	1	2017	Revista	[100]
E33	Scopus	3	2016	Revista	[151]
E34	WoS	3	2017	Conferencia	[27]
E35	IEEE	1	2017	Conferencia	[96]
E36	IEEE	1	2014	Conferencia	[158]
E37	WoS	1	2017	Revista	[136]
E38	IEEE	1	2013	Conferencia	[109]
E39	IEEE	1	2017	Revista	[105]
E40	IEEE	1	2015	Conferencia	[167]
E41	WoS	1	2011	Conferencia	[67]
E42	IEEE	1	2014	Conferencia	[182]
E43	Scopus	1	2015	Conferencia	[99]
E44	Scopus	1	2016	Conferencia	[54]
E45	Scopus	3	2015	Conferencia	[12]
E46	ACM	1	2016	Conferencia	[48]
E47	ACM	1	2017	Conferencia	[246]
E48	IEEE	1	2017	Conferencia	[24]
E49	Scopus	1	2015	Revista	[98]
E50	Scopus	1	2014	Conferencia	[187]

Anexo C. Interfaz 1¹

The image shows a web interface titled "Building Macro-Recipes" with three main sections for defining recipes:

- Macro-Recipe IFTTT:** Includes fields for "Macro-Recipe name", "Macro-Recipe description", and "Macro-Recipe version", along with a "Generate Macro-Recipe" button and a "List of Micro-Recipes" dropdown.
- Simple Micro-Recipe IFTTT:** Includes fields for "Micro-Recipe name", "Micro-Recipe state" (set to "Active"), and "Micro-Recipe priority" (set to "1"). It also features "IF-THIS of the Simple Micro-Recipe Channel" and "THEN-THAT of the Simple Micro-Recipe Channel" sections, each with "Network", "Object", and "Resource" buttons, an "Operator" dropdown (set to "LESS THAN"), and a "Value" input field.
- Nested Micro-Recipe IFTTT:** Includes fields for "Micro-Recipe name", "Micro-Recipe state" (set to "Active"), and "Micro-Recipe priority" (set to "1"). It features "IF-THESE Channel" and "THEN-THOSE Channel" sections, each with "Network", "Object", and "Resource" buttons, an "Operator" dropdown (set to "LESS THAN"), and a "Value" input field. Additionally, it has "List of THEN-THIS" and "List of THEN-THAT" dropdowns, and "Operator for linking THEN-THIS components" and "Operator for linking THEN-THAT components" dropdowns.

FIGURA D.1: Interfaz para crear un WAC.

¹Interfaz web que forma parte de la aplicación IoA-Building Tool. Permite la creación de componentes WAC a partir de la definición de micro- y macro-recetas. La macro-receta IFTTT se define en la primera sección de la interfaz; mientras que las micro-recetas que la componen se definen en las secciones 2 (recetas simples) y 3 (recetas anidadas)

Anexo D. Interfaz 2 ²

The image shows a web-based interface for semantic enrichment, divided into several sections:

- Semantic Enrichment of the IoT object** (Header)
- IoT-Object Profile**
 - IoT-object name:
 - IoT-object keywords: +
 - IoT-object state: ▼
 - IoT-object description:
 - IoT-object type: ▼
- IoT-Object Provider**
 - Provider name:
 - Provider website:
 - Provider social media: +
 - Provider phone:
 - Provider location:
 - List of social media: ▼
- IoT-Object Context**
 - latitude: longitude: Country: locality: time-zone: Country:
 - locations: ▼ value: +
 - IoT-resource: ▼ value: + List of keywords: ▼
 - List of locations: ▼
- IoT-Object Channels**
 - Select a file with channel configuration
- Agent Profile**
 - Agent name:
 - Agent description:
 - Agent state: ▼
 - Interest in receive data
 - Unit name: ▼
 - Unit kind: ▼
 - List of interest data: ▼
 - Interest in provide data
 - Unit name: ▼
 - Unit kind: ▼
 - List of interest data: ▼

.....

FIGURA E.1: Interfaz de enriquecimiento semántico

²La interfaz gráfica ilustrada constituye una sección de la desplegada por la aplicación IoT-Building Tool para realizar el enriquecimiento semántico del contrato (LAC) de un agente (LOA). Dicha interfaz está distribuida en secciones acorde a los componentes ontológicos integrados en IoA-OWL. En el ejemplo se muestran dos secciones, AgentSmartThing y AgentProfile.

Anexo E. Especificación UML³

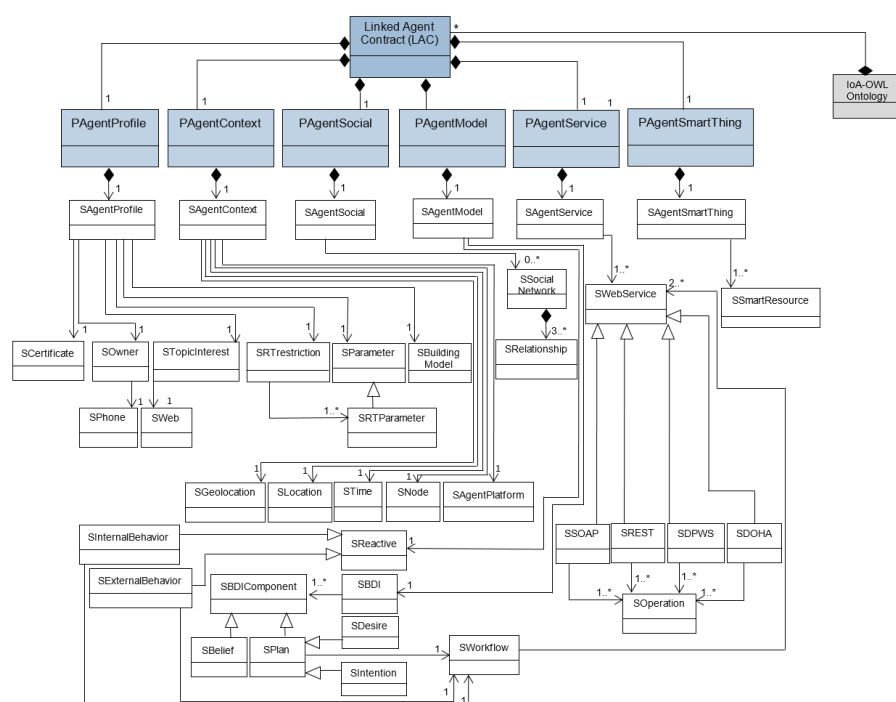


FIGURA F.1: Especificación UML usada para el desarrollo de la herramienta *IoA-Building Tool*.

³La especificación UML organiza los conceptos que definen la ontología IoA-OWL. Estos conceptos están organizados en tres tipo de clases: estructura de datos (prefijo S), estructura para crear secciones contractuales (prefijo P) y estructura del contrato (LAC)

Bibliografía

- [1] Wil MP Van der Aalst. «Data scientist: The engineer of the future». En: *Enterprise interoperability VI*. Springer, 2014, págs. 13-26.
- [2] Ruwantissa Abeyratne. «The Internet of Everything». En: *Mega-trends and Air Transport*. Springer, 2017, págs. 213-244.
- [3] Amit Adhikaree, Harshvardhan Makani y Jihoon Yun. «Internet of Things-Enabled Multiagent System for Residential DC Microgrids». En: *2017 IEEE International Conference on Electro Information Technology (EIT)*. 2017.
- [4] Sareh Aghaei, Mohammad Ali Nematbakhsh y Hadi Khosravi Farsani. «Evolution of the world wide web: From WEB 1.0 TO WEB 4.0». En: *International Journal of Web & Semantic Technology* 3.1 (2012), pág. 1.
- [5] O Zohreh Akbari. «A survey of agent-oriented software engineering paradigm: Towards its industrial acceptance». En: *International Journal of Computer Engineering Research* 1.2 (2010), págs. 14-28.
- [6] Ala Al-Fuqaha et al. «Internet of things: A survey on enabling technologies, protocols, and applications». En: *IEEE Communications Surveys & Tutorials* 17.4 (2015), págs. 2347-2376.
- [7] Hasan Al-Sakran. «Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology». En: *International Journal of Advanced Computer Science and Applications(IJACSA)* 5.2 (2015), págs. 37-43.
- [8] Riyadh Al-Shaqi, Monjur Mourshed y Yacine Rezgui. «Progress in ambient assisted systems for independent living by the elderly». En: *SpringerPlus* 5.1 (2016), pág. 624.

-
- [9] Fadi Al-Turjman y Sinem Alturjman. «Context-sensitive Access in Industrial Internet of Things (IIoT) Healthcare Applications». En: *IEEE Transactions on Industrial Informatics*. (2018). DOI: [10.1109/TII.2018.2808190](https://doi.org/10.1109/TII.2018.2808190).
- [10] Fadele Ayotunde Alaba et al. «Internet of Things security: A survey». En: *Journal of Network and Computer Applications* 88 (2017), págs. 10-28.
- [11] Alper T Alan et al. «Managing energy tariffs with agents: a field study of a future smart energy system at home». En: *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM. 2015, págs. 1551-1558.
- [12] Alper T Alan et al. «Tariff agent: interacting with a future smart energy system at home». En: *ACM Transactions on Computer-Human Interaction (TOCHI)* 23.4 (2016), pág. 25.
- [13] Unai Alegre, Juan Carlos Augusto y Tony Clark. «Engineering context-aware systems and applications: A survey». En: *Journal of Systems and Software* 117 (2016), págs. 55-83.
- [14] C Alexakos y Athanasios P Kalogeras. «Internet of Things integration to a Multi Agent System based manufacturing environment». En: *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE. 2015, págs. 1-8.
- [15] Saad Alqithami y Marco Lützenberger. *IoA 2017 Internet of Agents*. 2017. URL: <http://ioa.alqithami.com/2017/IoA'17-Proceedings.pdf#page=7>.
- [16] Mohammed Alreshoodi y John Woods. «Survey on QoE\QoS correlation models for multimedia services». En: *International Journal of Distributed and Parallel systems (IJDPS)* 4.3 (2013), págs. 1-20.
- [17] Francisco Andrade et al. «Intelligent contracting: Software agents, corporate bodies and virtual organizations». En: *Establishing The Foundation of Collaborative Networks*. Springer, 2007, págs. 217-224.

- [18] Abdullah Aref y Thomas Tran. «Acting as a Trustee for Internet of Agents in the Absence of Explicit Feedback». En: *International Conference on E-Technologies*. Springer. 2017, págs. 3-23.
- [19] Abdullah Aref y Thomas Tran. «Multi-criteria trust establishment for Internet of Agents in smart grids». En: *Multiagent and Grid Systems* 13.3 (2017), págs. 287-309.
- [20] Kevin Ashton et al. «That ‘internet of things’ thing». En: *RFID journal* 22.7 (2009), págs. 97-114.
- [21] Luigi Atzori, Antonio Iera y Giacomo Morabito. «From "smart objects" to "social objects": The next evolutionary step of the internet of things». En: *IEEE Communications Magazine* 52.1 (2014), págs. 97-105.
- [22] Luigi Atzori, Antonio Iera y Giacomo Morabito. «Siot: Giving a social structure to the internet of things». En: *IEEE communications letters* 15.11 (2011), págs. 1193-1195.
- [23] Luigi Atzori, Antonio Iera y Giacomo Morabito. «The internet of things: A survey». En: *Computer networks* 54.15 (2010), págs. 2787-2805.
- [24] Inmaculada Ayala, Mercedes Amor y Lidia Fuentes. «The Sol agent platform: Enabling group communication and interoperability of self-configuring agents in the Internet of Things». En: *Journal of Ambient Intelligence and Smart Environments* 7.2 (2015), págs. 243-269.
- [25] Costin Bădică et al. «Agent reasoning on the web using web services?» En: *Computer Science and Information Systems* 11.2 (2014), págs. 697-721.
- [26] Tina Balke y Nigel Gilbert. «How do agents make decisions? A survey». En: *Journal of Artificial Societies and Social Simulation* 17.4 (2014), pág. 13.
- [27] Komal Batool y Muaz A Niazi. «Modeling the internet of things: a hybrid modeling approach using complex networks and agent-based models». En: *Complex Adaptive Systems Modeling* 5.1 (2017), pág. 19.

-
- [28] Fabio Bellifemine et al. «JADE: A software framework for developing multi-agent applications. Lessons learned». En: *Information and Software Technology* 50.1 (2008), págs. 10-21.
- [29] Federico Bergenti, Eleonora Iotti y Agostino Poggi. «Outline of a Formalization of JADE Multi-Agent Systems.» En: *WOA*. 2015, págs. 123-128.
- [30] Federico Bergenti y Agostino Poggi. «Leap: A fipa platform for handheld and mobile devices». En: *8th International Workshop on Agent Theories, Architectures, and Languages*. Seattle, USA, 2001, págs. 436-446.
- [31] S Bhaksaran, J Shekhar y G Wildfred. «Mobile Agent Frame (MAF) based Internet of Things (IoT) system for allowing diverse heterogeneous devices to coexist, has MAF that is running on every device that can host static and mobile agents and receive mobile programs and messages». Pat. IN201200674-I2. 2013.
- [32] N. Bhaskaran, J. Shekhar y G. Wilfred. «Mobile Agent Frame (MAF) based Internet of Things (IoT) system for allowing diverse heterogeneous devices to coexist, has MAF that is running on every device that can host static and mobile agents and receive mobile programs and messages». Pat. IN201200674-I2. 2013.
- [33] Christian Bizer, Tom Heath y Tim Berners-Lee. «Linked data—the story so far». En: *International Journal on Semantic Web and Information Systems* 5.3 (2009), págs. 205-227.
- [34] Yolanda Blanco-Fernández et al. «AVATAR: An advanced multi-agent recommender system of personalized TV contents by semantic reasoning». En: *International conference on web information systems engineering*. Springer. 2004, págs. 415-421.
- [35] Fernando Boavida et al. «People-centric internet of things—challenges, approach, and enabling technologies». En: *Intelligent Distributed Computing IX*. Springer, 2016, págs. 463-474.
- [36] M Boniface, M SurrIDGE y C Upstill. *Position paper: Research challenges for the core platform for the future Internet*. Inf. téc. 1-9. University of Southampton IT Innovation Centre., 2010.

- [37] Flavio Bonomi et al. «Fog computing and its role in the internet of things». En: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM. 2012, págs. 13-16.
- [38] Eleonora Borgia. «The Internet of Things vision: Key features, applications and open issues». En: *Computer Communications* 54 (2014), págs. 1-31.
- [39] Stefan Bosse. «From the Internet-of-Things to Sensor Clouds - Unified Distributed Computing in Heterogeneous Environments with Smart and Mobile Multi-Agent Systems». En: *Proceedings of the Smart Systems Integration Conference 2015, (SSI-2015)*. 2015.
- [40] Stefan Bosse. «Mobile Multi-agent Systems for the Internet-of-Things and Clouds Using the JavaScript Agent Machine Platform and Machine Learning as a Service». En: *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (Fi-Cloud)*. IEEE. 2016, págs. 244-253.
- [41] Lars Braubach y Pokahr Alexander. *Jadex Tutorial*. Inf. téc. University of Hamburg, 2004. URL: <https://sourceforge.net/projects/jadex/>.
- [42] Lars Braubach, Alexander Pokahr y Winfried Lamersdorf. «Jadex: A short overview». En: *Main Conference Net.ObjectDays 2004*. 2004, págs. 195-207.
- [43] F. Brazier et al. «Can agents close contracts?» En: (2003). <https://pdfs.semanticscholar.org/bcf0/da4a0eabb55c8d5a4f865df6740d56ab0a19.pdf>. Accessed 14 December 2016.
- [44] D. Brickley y L. Miller. *FOAF Vocabulary Specification 0.99*. Accessed 14 December 2016. 2014. URL: <http://xmlns.com/foaf/spec/>.
- [45] Arne Bröring et al. «Semantically-enabled sensor plug & play for the sensor web». En: *Sensors* 11.8 (2011), págs. 7568-7605.
- [46] Khac-Hoai Nam Bui y Jason J Jung. «Internet of agents framework for connected vehicles: A case study on distributed traffic control system». En: *Journal of Parallel and Distributed Computing* (2017).

- [47] P. Buxmann, T. Hess y R. Bus Ruggaber. «Internet of Services». En: *Inf. Syst. Eng* (2009).
- [48] Davide Calvaresi et al. «The challenge of real-time multi-agent systems for enabling IoT and CPS». En: *Proceedings of the International Conference on Web Intelligence*. New York:ACM. Leipzig, Germany, 23-26 August 2017, 2017, págs. 356-364.
- [49] S Canale et al. «A Future Internet oriented user centric extended intelligent transportation system». En: *2016 24th Mediterranean Conference on Control and Automation (MED)*. IEEE. 2016, págs. 1133-1139.
- [50] Jorge Cardoso, Konrad Voigt y Matthias Winkler. «Service engineering for the internet of services». En: *International Conference on Enterprise Information Systems*. Springer. 2008, págs. 15-27.
- [51] Luciana Cardoso et al. «The Next Generation of Interoperability Agents in Healthcare». En: *Int. J. Environ. Res. Public Health* 11 (2014), págs. 5349-5371.
- [52] Florent Carlier y Valérie Renault. «IoT-a, Embedded Agents for Smart Internet of Things. Application on a Display Wall». En: *IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*. IEEE. 2016, págs. 80-83.
- [53] Everton Cavalcante et al. «On the interplay of Internet of Things and Cloud Computing: A systematic mapping study». En: *Computer Communications* 89 (2016), págs. 17-33.
- [54] Pablo Chamoso et al. «Swarm agent-based architecture suitable for internet of things and smartcities». En: *12th International Conference on Distributed Computing and Artificial Intelligence*. Springer. 2015, págs. 21-29.
- [55] Charalampos Chelmiss y Viktor K Prasanna. «Social networking analysis: A state of the art and the effect of semantics». En: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom)*. IEEE. 2011, págs. 531-536.

- [56] H Chen. *Context Broker Architecture: An Intelligent Broker for Context-Aware Systems in Smart Spaces*. Accessed 14 December 2016. 2004. URL: <http://dam1.umbc.edu/ontologies/cobra/0.4/location>.
- [57] Harry Chen, Tim Finin y Anupam Joshi. «The SOUPA ontology for pervasive computing». En: *Ontologies for agents: Theory and experiences*. Springer, 2005, págs. 233-258.
- [58] Min Chen, Shiwen Mao y Yunhao Liu. «Big data: A survey». En: *Mobile networks and applications* 19.2 (2014), págs. 171-209.
- [59] Krzysztof Chmiel et al. «Efficiency of JADE agent platform». En: *Scientific Programming* 13.2 (2005), págs. 159-172.
- [60] Sang-Il Choi y Seok-Joo Koh. «Distributed CoAP Handover Using Distributed Mobility Agents in Internet-of-Things Networks». En: *Journal of information and communication convergence engineering* 15.1 (2017), págs. 37-42.
- [61] Franco Cicirelli et al. «An edge-based platform for dynamic Smart City applications». En: *Future Generation Computer Systems* 76 (2017), págs. 106-118.
- [62] Cisco. *Connections Counter: The Internet of Everything in Motion*. <https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342>. 2014.
- [63] CoDAMoS-Project. *Context-Driven Adaptation of Mobile Services*. Accessed 14 December 2016. 2003.
- [64] Raphael Cohen-Almagor. *Internet history*. 2011.
- [65] Michael Compton et al. «A survey of the semantic specification of sensors». En: *Proceedings of the 2nd International Conference on Semantic Sensor Networks*. Washington DC, USA, 26 October 2009, 2009, págs. 17-32.
- [66] Marco Conti, Andrea Passarella y Sajal K Das. «The Internet of People (IoP): A new wave in pervasive mobile computing». En: *Pervasive and Mobile Computing* 41 (2017), págs. 1-27.

- [67] Bertrand Copigneaux. «Semi-autonomous, context-aware, agent using behaviour modelling and reputation systems to authorize data operation in the Internet of Things». En: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE. Seoul, 6-8 March 2014, 2014, págs. 411-416. DOI: [10.1109/WF-IoT.2014.6803201](https://doi.org/10.1109/WF-IoT.2014.6803201).
- [68] M. Corchado. *Modelos y Arquitecturas de Agente*. Inf. téc. Universidad de Vigo, 2004.
- [69] Graham Cormode y Balachander Krishnamurthy. «Key differences between Web 1.0 and Web 2.0». En: *First Monday* 13.6 (2008).
- [70] S. Cox y C. Little. *Time Ontology in OWL*. Accessed 14 December 2016. 2016. URL: <https://www.w3.org/2016/time>.
- [71] Li Da Xu, Wu He y Shancang Li. «Internet of things in industries: A survey». En: *IEEE Transactions on industrial informatics* 10.4 (2014), págs. 2233-2243.
- [72] Priyanka Dadhich, Kamlesh Dutta y MC Govil. «Security issues in mobile agents». En: *International Journal of Computer Applications* 11.4 (2010), págs. 1-7.
- [73] Abdelouahid Derhab, Mohamed Guerroumi y Mohamed Younis. *Wireless and mobile sensing technologies for the future Internet*. 2017.
- [74] Manuel Díaz, Cristian Martín y Bartolomé Rubio. «State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing». En: *Journal of Network and Computer Applications* 67 (2016), págs. 99-117.
- [75] Frank Dignum et al. «Towards socially sophisticated BDI agents». En: *Fourth International Conference on MultiAgent Systems*. IEEE. 2000, págs. 111-118.
- [76] Nicola Dragoni et al. «Microservices: yesterday, today, and tomorrow». En: *Present and Ulterior Software Engineering*. Springer, 2017, págs. 195-216.
- [77] Amy Drahota y Ann Dewey. «The sociogram: a useful tool in the analysis of focus groups». En: *Nursing research* 57.4 (2008), págs. 293-297.

- [78] Yucong Duan, Yuan Cao y Xiaobing Sun. «Various “aaS” of everything as a service». En: *2015 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE. 2015, págs. 1-6.
- [79] Dublin-Core-Metadata-Initiative. *DCMI Metadata Terms*. Accessed 14 December 2016. 2012. URL: <http://dublincore.org/documents/dcmi-terms/>.
- [80] Schahram Dustdar y Wolfgang Schreiner. «A survey on web services composition». En: *International journal of web and grid services* 1.1 (2005), págs. 1-30.
- [81] Michał Dyk, Andrzej Najgebauer y Dariusz Pierzchała. «Agent-based M&S of smart sensors for knowledge acquisition inside the Internet of Things and sensor networks». En: *Asian Conference on Intelligent Information and Database Systems*. Springer International Publishing. Bali, Indonesia, 23-25 March 2015, 2015, págs. 224-234.
- [82] Mirko D’Angelo y Mauro Caporuscio. «Pure Edge Computing Platform for the Future Internet». En: *Federation of International Conferences on Software Technologies: Applications and Foundations*. Springer. 2016, págs. 458-469.
- [83] Mahmoud Elkhodr, Seyed Shahrestani y Hon Cheung. «A contextual-adaptive location disclosure agent for general devices in the internet of things». En: *2013 IEEE 38th Conference on Local Computer Networks Workshops (LCN Workshops)*. IEEE. 2013, págs. 848-855.
- [84] JHP Eloff et al. «Internet of people, things and services-the convergence of security, trust and privacy». En: (2009).
- [85] G. Ereteo, F. Gandon y M. Buffa. *INRIA’s IRC ontology as RDF vocabulary*. Accessed 14 December 2016. 2008. URL: <http://ns.inria.fr/semsna/2008/10/13/voc.html>.
- [86] Thomas Erl. *Service-oriented architecture (SOA): concepts, technology, and design*. 2005.
- [87] Thomas Erl. *Soa: principles of service design*. ISBN:0136042597. Prentice Hall Press, New York, 2008.

- [88] Thomas Erl et al. *Web service contract design and versioning for SOA*. Prentice Hall, 2017.
- [89] Dave Evans. «The internet of everything: How more relevant and valuable connections will change the world». En: *Cisco IBSG 2012* (2012), págs. 1-9.
- [90] Jianqing Fan, Fang Han y Han Liu. «Challenges of big data analysis». En: *National science review* 1.2 (2014), págs. 293-314.
- [91] Bahar Farahani et al. «Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare». En: *Future Generation Computer Systems* 78.2 (2018), págs. 659-676.
- [92] F Fatemi et al. «Cloud Computing Challenges and Opportunities: A Survey». En: *Telematics and Future Generation Networks (TAFGEN)* 1 (2015), pág. 34.
- [93] FIPA. *FIPA ACL Message Structure Specification*. 2002. URL: <http://www.fipa.org/specs/fipa00061/SC00061G.html>.
- [94] FIPA. *FIPA Contract Net Interaction Protocol Specification*. 2002. URL: <http://www.fipa.org/specs/fipa00029/SC00029H.html>.
- [95] FIPA. *Interaction Protocol Specifications*. 2012, URL <http://www.fipa.org/repository/ips.php3> (accessed 3.11.16). FIPA.
- [96] Agostino Forestiero. «Multi-agent recommendation system in Internet of Things». En: *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press. 2017, págs. 772-775.
- [97] Giancarlo Fortino, Antonio Guerrieri y Wilma Russo. «Agent-oriented smart objects development». En: *2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE. 2012, págs. 907-912.
- [98] Giancarlo Fortino y Wilma Russo. «Towards a Cloud-assisted and Agent-oriented Architecture for the Internet of Things.» En: *WOA@ AI*IA*. 2013, págs. 60-65.
- [99] Giancarlo Fortino, Wilma Russo y Claudio Savaglio. «Simulation of Agent-oriented Internet of Things Systems.» En: *WOA*. 2016, págs. 8-13.

- [100] Giancarlo Fortino et al. «Modeling and Simulating Internet-of-Things Systems: A Hybrid Agent-Oriented Approach». En: *Computing in Science & Engineering* 19.5 (2017), págs. 68-76.
- [101] Stan Franklin y Art Graesser. «Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents». En: *International Workshop on Agent Theories, Architectures, and Languages*. Springer. 1996, págs. 21-35.
- [102] Paul Fremantle. «A reference architecture for the internet of things». En: *WSO2 White paper* (2014).
- [103] Emilia Garcia, Adriana Giret y Vicente Botti. «Regulated open multi-agent systems based on contracts». En: *Information systems development*. Springer, 2011, págs. 243-255.
- [104] Iván García-Magariño, Raquel Lacuesta y Jaime Lloret. «Agent-based simulation of smart beds with Internet-of-Things for exploring big data analytics». En: *IEEE Access* (2017).
- [105] Benjamin Gateau y Jarogniew Rykowski. «Personal e-comfort modelling and management based on Multi-Agent System and Internet of Things network». En: *2015 International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS)*. IEEE. 2015, págs. 83-89.
- [106] Vangelis Gazis et al. «A survey of technologies for the internet of things». En: *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2015, págs. 1090-1095.
- [107] David Gil et al. «Internet of things: A review of surveys based on context aware intelligent services». En: *Sensors* 16.7 (2016), pág. 1069.
- [108] Caire Giovanni. *JADE Tutorial. JADE programming for beginners*. Inf. téc. <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>. TILab S.p.A., 2009.
- [109] W Wilfred Godfrey, Shashi Shekhar Jha y Shivashankar B Nair. «On a mobile agent framework for an internet of things». En: *2013 International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE. 2013, págs. 345-350.

- [110] Veronica M Godshalk. «May the Internet of People, Things and Services Be Too Much of a Good Thing for Career and Human Resource Management?» En: *The Internet of People, Things and Services: Workplace Transformations* (2018).
- [111] Vladimir Gorodetsky. «Internet of Agents: From Set of Autonomous Agents to Network Object». En: *Second International Workshop on Internet of Agents*. 2017, págs. 1-17.
- [112] Benjamin N Grosz y Terrence C Poon. «SweetDeal: Representing agent contracts with exceptions using semantic web rules, ontologies, and process descriptions». En: *International Journal of Electronic Commerce* 8.4 (2004), págs. 61-97.
- [113] Dominique Guinard et al. «From the internet of things to the web of things: Resource-oriented architecture and best practices». En: *Architecting the Internet of things* (2011), págs. 97-129.
- [114] Bin Guo et al. «Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things». En: *Journal of Network and Computer Applications* 36.6 (2013), págs. 1531-1539.
- [115] Xiaofeng Guo, Jianjing Shen y Zuwei Yin. «On software development based on SOA and ROA». En: *2010 Chinese Control and Decision Conference (CCDC)*. IEEE. 2010, págs. 1032-1035.
- [116] A. Gyrard et al. *The Machine-to-Machine Measurement (M3) Lite Ontology (m3lite)*. Accessed 14 December 2016. 2016. URL: <http://lov.okfn.org/dataset/lov/vocabs/m3lite>.
- [117] Sara Hachem, Thiago Teixeira y Valérie Issarny. «Ontologies for the internet of things». En: *Proceedings of the 8th Middleware Doctoral Symposium*. ACM. 2011, pág. 3.
- [118] Hatem Hamad, Motaz Saad y Ramzi Abed. «Performance Evaluation of RESTful Web Services for Mobile Devices.» En: *Int. Arab J. e-Technol.* 1.3 (2010), págs. 72-78.
- [119] Qusay F. Hassan. *Internet of Things A to Z. Technologies and applications*. John Wiley & Sons, 2018.
- [120] Tom Heath y Christian Bizer. «Linked data: Evolving the web into a global data space». En: *Synthesis lectures on the semantic web: theory and technology* 1.1 (2011), págs. 1-136.

- [121] James Hendler. «Agents and the semantic web». En: *IEEE Intelligent systems* 16.2 (2001), págs. 30-37.
- [122] Marco E Pérez Hernández y Stephan Reiff-Marganiec. «Autonomous and self controlling smart objects for the future internet». En: *2015 3rd international conference on Future internet of things and cloud (FiCloud)*. IEEE. 2015, págs. 301-308.
- [123] José M Hernández-Muñoz et al. «Smart cities at the forefront of the future internet». En: *The Future Internet Assembly*. Springer. 2011, págs. 447-462.
- [124] Matthew B Hoy. «If this then that: An introduction to automated task services». En: *Medical reference services quarterly* 34.1 (2015), págs. 98-103.
- [125] Yung-Li Hu et al. «A programming framework for implementing fault-tolerant mechanism in IoT applications». En: *International Conference on Algorithms and Architectures for Parallel Processing*. Springer. 2015, págs. 771-784.
- [126] R. Iannella y J. McKinney. *vCard Ontology - for Describing People and Organizations*. Accessed 14 December 2016. 2014. URL: <https://www.w3.org/TR/vcard-rdf/>.
- [127] Saman Iftikhar, H Farooq Ahmad e Hiroki Suguri. «Agent based semantic interoperability between agents and semantic web languages». En: *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*. IEEE. 2008, págs. 1660-1665.
- [128] Saman Iftikhar et al. «Ontology agent for ensuring semantic interoperability among agents and semantic web». En: *Int. J. Knowledge Engineering and Data Mining* 1.3 (2011).
- [129] SM Riazul Islam et al. «The internet of things for health care: a comprehensive survey». En: *IEEE Access* 3 (2015), págs. 678-708.
- [130] Valérie Issarny et al. «Service-oriented middleware for the future internet: state of the art and research directions». En: *Journal of Internet Services and Applications* 2.1 (2011), págs. 23-45.

- [131] Muhammad Jaffar-ur et al. «Testing software components for integration: a survey of issues and techniques». En: *Journal of Software Testing, Verification and Reliability* 17.2 (2007), págs. 95-133.
- [132] Laura Jarvenpaa et al. «Mobile agents for the internet of things». En: *2013 17th International Conference System Theory, Control and Computing (ICSTCC)*. IEEE. 2013, págs. 763-767.
- [133] Simon Jirka, Arne Bröring y Christoph Stasch. «Discovery mechanisms for the sensor web». En: *Sensors* 9.4 (2009), págs. 2661-2681.
- [134] JKennedy John. *An interview with Nikola Tesla*. 1926.
- [135] Nicholas J Kaminski, Maria Murphy y Nicola Marchetti. «Agent-based modeling of an IoT network». En: *2016 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE. 2016, págs. 1-7.
- [136] Takumi Kato, Hideyuki Takahashi y Tetsuo Kinoshita. «Multiagent-based Autonomic and Resilient Service Provisioning Architecture for the Internet of Things». En: *IJCSNS* 17.6 (2017), pág. 36.
- [137] Dimitrios Katsaros, Nikos Dimokas y Leandros Tassiulas. «Social network analysis concepts in the design of wireless ad hoc network protocols». En: *IEEE network* 24.6 (2010).
- [138] Staffs Keele et al. «Guidelines for performing systematic literature reviews in software engineering». En: *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn, 2007.
- [139] Ian Kerr. «The internet of people? reflections on the future regulation of human-implantable radio frequency identification». En: ed. por Oxford University Press. 2013.
- [140] Rafiullah Khan et al. «Future internet: the internet of things architecture, possible applications and key challenges». En: *2012 10th International Conference on Frontiers of Information Technology (FIT)*. IEEE. 2012, págs. 257-260.
- [141] Shah Khusro et al. «Unleashing sensor data on linked open data-the story so far». En: *Life Science Journal* 10.4 (2013), págs. 1766-1785.

- [142] Sara Kiesler et al. «Internet evolution and social impact». En: *It & Society* 1.1 (2002), págs. 120-134.
- [143] Marco Knol et al. «Internet of smart things, a study on embedding agents and information as a service». En: *ICAART Proceedings* 1 (2016), págs. 102-109.
- [144] A Kofod-Petersen. *How to do a structured literature review in computer science, version 0.2*. 2014.
- [145] Naomi Korn y Charles Oppenheim. «Licensing Open Data: a practical guide». En: *Higher Education Funding Council for England*. [Consulta: 14/2/2014]. Disponible en: http://discovery.ac.uk/files/pdf/Licensing_Open_Data_A_Practical_Guide.pdf (2011).
- [146] Matthias Kranz, Paul Holleis y Albrecht Schmidt. «Embedded interaction: Interacting with the internet of things». En: *IEEE internet computing* 14.2 (2010), págs. 46-53.
- [147] Kalliopi Kravari y Nick Bassiliades. «A survey of agent platforms». En: *Journal of Artificial Societies and Social Simulation* 18.1 (2015), pág. 11.
- [148] Alok Kulkarni y Sampada Sathe. «Healthcare applications of the Internet of Things: A Review». En: *International Journal of Computer Science and Information Technologies* 5.5 (2014), págs. 6229-32.
- [149] Alexandros Labrinidis y Hosagrahar V Jagadish. «Challenges and opportunities with big data». En: *Proceedings of the VLDB Endowment* 5.12 (2012), págs. 2032-2033.
- [150] Michal Laclavik et al. «Agentowl: Semantic knowledge model and agent architecture». En: *Computing and Informatics* 25.5 (2012), págs. 421-439.
- [151] Samreen Laghari y Muaz A Niazi. «Modeling the internet of things, self-organizing and other complex adaptive communication networks: a cognitive agent-based computing approach». En: *PloS one* 11.1 (2016), e0146760.
- [152] Giordano Lanzola et al. «Remote blood glucose monitoring in mHealth scenarios: a review». En: *Sensors* 16.12 (2016), pág. 1983.

- [153] Phillip A Laplante y Nancy Laplante. «The internet of things in healthcare: Potential applications and challenges». En: *IT Professional* 18.3 (2016), págs. 2-4.
- [154] Carlos Laufer y Daniel Schwabe. «Semantic Contract Support for E-Business Processes». En: *Web Conference, 2007. LA-WEB 2007. Latin American*. IEEE. 2007, págs. 67-75.
- [155] Jonathan Lee, Shin-Jie Lee y Ping-Feng Wang. «A framework for composing SOAP, non-SOAP and non-web services». En: *IEEE Transactions on Services Computing* 8.2 (2015), págs. 240-250.
- [156] Barry M Leiner et al. «A brief history of the Internet». En: *ACM SIGCOMM Computer Communication Review* 39.5 (2009), págs. 22-31.
- [157] Florin Leon, Marcin Paprzycki y Maria Ganzha. *A Review of Agent Platforms*. Inf. téc. Technical University of Iași, 2015.
- [158] Peter Leong y Liming Lu. «Multiagent web for the internet of things». En: *2014 International Conference on Information Science and Applications (ICISA)*. IEEE. 2014, págs. 1-4.
- [159] Teemu Leppänen et al. «Augmented reality web applications with mobile agents in the internet of things». En: *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST)*. IEEE. 2014, págs. 54-59.
- [160] Teemu Leppänen et al. «Mobile agents-based smart objects for the internet of things». En: *Internet of Things Based on Smart Objects*. Springer, 2014, págs. 29-48.
- [161] Teemu Leppänen et al. «Mobile agents for integration of internet of things and wireless sensor networks». En: *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2013, págs. 14-21.
- [162] Ling Li, Shancang Li y Shanshan Zhao. «QoS-aware scheduling of services-oriented internet of things». En: *IEEE Transactions on Industrial Informatics* 10.2 (2014), págs. 1497-1505.
- [163] Shancang Li, Li Da Xu y Shanshan Zhao. «The internet of things: a survey». En: *Information Systems Frontiers* 17.2 (2015), págs. 243-259.

- [164] Chih-Hao Lin, Pin-Han Ho y Hong-Chuan Lin. «Framework for NFC-based intelligent agents: a context-awareness enabler for social internet of things». En: vol. 10. 2. SAGE Publications Sage UK: London, England, 2014, pág. 978951.
- [165] Chih-Hao Liu y Jason Jen-Yen Chen. «Using Ontology-based BDI Agent to Dynamically Customize Workflow and Bind Semantic Web Service.» En: *Journal of Software* 7.4 (2012), págs. 884-894.
- [166] Ramon Lopes, Flavio Assis y Carlos Montez. «MASPOT: A mobile agent system for Sun SPOT». En: *2011 10th international symposium on Autonomous decentralized systems (ISADS)*. IEEE. 2011, págs. 25-31.
- [167] Tomás Sánchez López et al. «Resource management in the Internet of Things: Clustering, synchronisation and software agents». En: *Uckelmann D., Harrison M., Michahelles F. (eds) Architecting the Internet of Things*. Berlin:Springer, 2011, págs. 159-193.
- [168] Roberto Lucchi, Michel Millot y Christian Elfers. «Resource oriented architecture and REST». En: *Assessment of impact and advantages on INSPIRE, Ispra: European Communities* (2008).
- [169] Zakaria Maamar et al. «Software Agents meet Internet of Things». En: *Internet Technology Letters* (2018).
- [170] Pattie Maes. *Designing autonomous agents: theory and practice from biology to engineering and back*. MIT press, 1990.
- [171] Mirjana Maksimović et al. «Raspberry Pi as Internet of things hardware: performances and constraints». En: *Proceedings of the 1st International Conference on Electrical, Electronic and Computing Engineering*. Vol. 3. USA:IEEE. Vrnjacka Banja, Serbia, 2-5 June 2014, 2014, pág. 8.
- [172] Bogdan Manate, Teodor-Florin Fortis y Viorel Negru. «Infrastructure Management Support in a Multi-Agent Architecture for Internet of Things». En: *2014 European Modelling Symposium (EMS)*. IEEE. 2014, págs. 372-377.
- [173] Mangoh. *Mangoh*. URL: <http://mangoh.io/>.
- [174] D. Martin et al. *OWL-S: Semantic Markup for Web Services*. Accessed 14 December 2016. 2004. URL: <https://www.w3.org/Submission/OWL-S/>.

- [175] Yair Meidan et al. «ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis». En: *Proceedings of the Symposium on Applied Computing*. ACM. 2017, págs. 506-509.
- [176] Bertrand Meyer. «Applying 'design by contract'». En: *Computer* 25.10 (1992), págs. 40-51.
- [177] Julien Mineraud et al. «A gap analysis of Internet-of-Things platforms». En: *Computer Communications* 89 (2016), págs. 5-16.
- [178] Daniele Miorandi et al. «Internet of things: Vision, applications and research challenges». En: *Ad hoc networks* 10.7 (2012), págs. 1497-1516.
- [179] Javier Miranda et al. «From the Internet of Things to the Internet of People». En: *IEEE Internet Computing* 19.2 (2015), págs. 40-47.
- [180] Leo van Moergestel et al. «Internet of smart things, a study on embedding agents and information as a service». En: *ICAART Proceedings* 1 (2016), págs. 102-109.
- [181] KamalEldin Mohamed y Duminda Wijesekera. «Performance analysis of web services on mobile devices». En: *Procedia Computer Science* 10 (2012), págs. 744-751.
- [182] Amir H Moin. «Sense-Deliberate-Act Cognitive Agents for Sense-Compute-Control Applications in the Internet of Things and Services». En: *Internet of Things. User-Centric IoT*. Springer, 2015, págs. 23-28.
- [183] Rafael Moreno-Vozmediano, Rubén S Montero e Ignacio M Llorente. «Key challenges in cloud computing: Enabling the future internet of services». En: *IEEE Internet Computing* 17.4 (2013), págs. 18-25.
- [184] Conor Muldoon et al. «Agent factory micro edition: A framework for ambient applications». En: *International Conference on Computational Science*. Springer. 2006, págs. 727-734.

- [185] Anas M Mzahm, Mohd Sharifuddin Ahmad y Alicia YC Tang. «Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT)». En: *2013 13th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE. 2013, págs. 159-164.
- [186] Anas M Mzahm, Mohd Sharifuddin Ahmad y AY Tang. «Enhancing the internet of things (iot) via the concept of agent of things (aot)». En: *Journal of Network and Innovative Computing* 2.2014 (2014), págs. 101-110.
- [187] Anas M Mzahm et al. «Towards a Design Model for Things in Agents of Things». En: *Proceedings of the International Conference on Internet of things and Cloud Computing*. ACM. 2016, pág. 41.
- [188] Katsuhiko Naito. «A Survey on the Internet-of-Things: Standards, Challenges and Future Prospects». En: *Journal of Information Processing* 25 (2017), págs. 23-31.
- [189] Nathalia Moraes do Nascimento y Carlos José Pereira de Lucena. «Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things». En: *Information Sciences* 378 (2017), págs. 161-176.
- [190] T Nasser y RS Tariq. «Big data challenges». En: *J Comput Eng Inf Technol* 4: 3. doi: <http://dx.doi.org/10.4172/2324-9307.2> (2015).
- [191] Ricardo Neisse et al. «An agent-based framework for informed consent in the internet of things». En: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. Milan, Italy, 14-16 December 2015, 2015, págs. 789-794.
- [192] Ricardo Neisse et al. «Informed consent in Internet of Things: The case study of cooperative intelligent transport systems». En: *Telecommunications (ICT), 2016 23rd International Conference on*. IEEE. 2016, págs. 1-5.
- [193] Juan Carlos Nieves, Daniel Andrade y Esteban Guerrero. «MAIoT-An IoT Architecture with Reasoning and Dialogue Capability». En: *Applications for Future Internet*. Springer, 2017.

- [194] Lyndon Nixon et al. «Future of the internet of services for industry: the ServiceWeb 3.0 roadmap». En: (2009).
- [195] OMG. *OMG Unified Modeling Language TM (OMG UML) V 2.5*. 2015, URL <http://www.omg.org/spec/UML/2.5/PDF/>, (accessed 3.1.17). OMG.
- [196] Mert Ozkaya y Christos Kloukinas. «Towards design-by-contract based software architecture design». En: *2013 IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT)*. IEEE. 2013, págs. 157-164.
- [197] Yvonne O'Connor et al. «Privacy by Design: Informed Consent and Internet of Things for Smart Health». En: *Procedia Computer Science* 113 (2017), págs. 653-658.
- [198] Pico Pablo y Juan A Holgado-Terriza. «A Framework for the Development of Smart Ubiquitous Real-Time Systems Based on the Internet of Agents and Internet of Services Approaches». En: *Intelligent Environments 2016: Workshop Proceedings of the 12th International Conference on Intelligent Environments*. Vol. 21. IOS Press. 2016, pág. 76.
- [199] Sangheon Park et al. «Authentication, Authorization, and Accounting (AAA) Framework in Network Mobility (NEMO) Environments». En: *Handbook of Research on Wireless Security* (2008), págs. 395-411.
- [200] P. Paderewski. «Un modelo arquitectónico evolutivo para sistemas software basados en agentes». Tesis doct. Universidad de Granada, 2013.
- [201] Kripa Shankar Pathak et al. «Making renewable energy SMART using Internet of Things (IOT)». En: *2014 International Conference on Advances in Engineering and Technology Research (ICAETR)*. IEEE. Unnao, India, 1-2 August 2014, 2014, págs. 1-4.
- [202] Juan Pablo Paz, Andrés Castillo Sanz y Rubén González Crespo. «An Evaluation of Integration Technologies to Expose Agent Actions as Web Services». En: *Practical Applications of Intelligent Systems*. Springer, 2014, págs. 259-270.
- [203] C. Perera et al. «A Survey on Internet of Things From Industrial Market Perspective». En: *IEEE Access* 2 (2014), págs. 1660-1679.

- [204] Riccardo Petrolo, Valeria Loscri y Nathalie Mitton. «Cyber-Physical Objects as key elements for a Smart Cyber-City». En: *Management of Cyber Physical Objects in the Future Internet of Things*. Springer, 2016, págs. 31-49.
- [205] P. Pico-Valencia et al. «Towards the Internet of Agents: An Analysis of the Internet of Things from the Intelligence and Autonomy Perspective». En: *Ingeniería e Investigación* 38.1 (2018), págs. 121-129.
- [206] Pablo Pico-Valencia. «Managing the Heterogeneity in Multiagent Systems Based on Service Orientation Technologies». En: *Computación y Sistemas* (2018). (In Review).
- [207] Pablo Pico-Valencia. «Middleware para el desarrollo de arquitectura inteligente orientada a Servicios en Entornos Ubicuos». Tesis de maestría. Universidad de Granada, 2015.
- [208] Pablo Pico-Valencia y Juan A Holgado-Terriza. «ADELE: a middleware for supporting the evolution of multi-agents systems based on a metaprogramming approach». En: *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*. Springer, 2016, págs. 297-310.
- [209] Pablo Pico-Valencia y Juan A Holgado-Terriza. «Semantic agent contracts for Internet of Agents». En: *IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*. IEEE. 2016, págs. 76-79.
- [210] Pablo Pico-Valencia, Juan A. Holgado-Terriza y Senso José. «A Systematic Method for Building Internet of Agents Applications Based on the Linked Open Data Approach». En: *Journal of* (2017).
- [211] Pablo Pico-Valencia, Juan A. Holgado-Terriza y José Senso. «Towards an Internet of Agents Model Based on Linked Open Data Approach». En: *Journal of Autonomous Agents and Multi-Agent Systems* (2017). (En revisión).
- [212] Pablo A Pico-Valencia y Juan A Holgado-Terriza. «An agent middleware for supporting ecosystems of heterogeneous web services». En: *Procedia Computer Science* 94 (2016), págs. 121-128.

- [213] Veselin Pizurica y Piet Vandaele. «A Cloud-Based Bayesian Smart Agent Architecture for Internet-of-Things Applications». En: *Internet of Things. User-Centric IoT*. Springer International Publishing. Rome, 2015, págs. 42-47.
- [214] Alexander Pokahr, Lars Braubach y Winfried Lamersdorf. «Jadex: A BDI reasoning engine». En: vol. 15. Boston: Springer, 2005. Cap. Bordini R.H., Dastani M., Dix J., El Fallah Seghrouchni A. (eds) Multi-Agent Programming. Multiagent Systems, Artificial Societies, and Simulated Organizations, págs. 149-174.
- [215] T. Predrag. «Some Thoughts on Programming Models, Middleware and Self-Healing Capabilities for the Next-Generation Internet-of-Agent». En: *IoA 2017 Internet of Agents*. Brazil, 2017, págs. 74-83.
- [216] Davy Preuveneers et al. «Towards an extensible context ontology for ambient intelligence». En: *European Symposium on Ambient Intelligence*. Springer. 2004, págs. 148-159.
- [217] Arun kishore Ramakrishnan, Davy Preuveneers y Yolande Berbers. «Enabling self-learning in dynamic and open IoT environments». En: *Procedia Computer Science* 32 (2014), págs. 207-214.
- [218] Sarvapali D Ramchurn et al. «A disaster response system based on human-agent collectives». En: *Journal of Artificial Intelligence Research* 57 (2016), págs. 661-708.
- [219] Shuping Ran. «A model for web services discovery with QoS». En: *ACM Sigecom exchanges* 4.1 (2003), págs. 1-10.
- [220] Mohammad Abdur Razzaque et al. «Middleware for internet of things: a survey». En: *IEEE Internet of Things Journal* 3.1 (2016), págs. 70-95.
- [221] Luís Ribeiro, José Barata y Pedro Mendes. «MAS and SOA: complementary automation paradigms». En: *Innovation in manufacturing networks*. Springer, 2008, págs. 259-268.
- [222] S. Rodriguez y J Holgado. «Distributed Service-Based Approach for Sensor Data Fusion in IoT Environments. Sensors, 14(10), 19200–19228.» En: *Sensors* 14.10 (2014), 19200–19228.

- [223] Sandra S Rodríguez y Juan A Holgado. «A home-automation platform towards ubiquitous spaces based on a decentralized p2p architecture». En: *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*. Springer Berlin Heidelberg, 2009, págs. 304-308.
- [224] B. Rodriguez-Castro, L. Torok y M. Hepp. *Computer Vocabulary Language Reference*. Accessed 14 December 2016. 2016. URL: <http://www.ebusiness-unibw.org/ontologies/opdm/computer.html>.
- [225] Sandra Rodriguez-Valenzuela et al. «Distributed service-based approach for sensor data fusion in IoT environments». En: *Sensors* 14.10 (2014), págs. 19200-19228.
- [226] Yefeng Ruan, Arjan Duresi y Lina Alfantoukh. «Trust management framework for internet of things». En: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE. 2016, págs. 1013-1019.
- [227] Stuart Russel y Peter Norvig. «Artificial Intelligence: A Modern Approach». En: *Prentice Hall Press Upper Saddle River* (2003).
- [228] Seref Sagiroglu y Duygu Sinanc. «Big data: A review». En: *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. 2013, págs. 42-47.
- [229] Ola Salman et al. «Edge computing enabling the Internet of Things». En: *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE. 2015, págs. 603-608.
- [230] Fahad Samad y ZA Memon. «The Future of Internet: IPv6 Fulfilling the Routing Needs in Internet of Things». En: *International Journal of Future Generation Communication and Networking* 11.1 (2018).
- [231] Farzad Samie, Lars Bauer y Jörg Henkel. «IoT technologies for embedded computing: A survey». En: *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. ACM. 2016, pág. 8.
- [232] João Santos et al. «Intelligent personal assistants based on internet of things approaches». En: *IEEE Systems Journal* PP.99 (2016), págs. 1-10.

- [233] Claudio Savaglio, Giancarlo Fortino y Mengchu Zhou. «Towards interoperable, cognitive and autonomic IoT systems: an agent-based approach». En: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE. 2016, págs. 58-63.
- [234] Claudio Savaglio et al. «Agent-Based Computing in the Internet of Things: A Survey». En: *International Symposium on Intelligent and Distributed Computing*. Springer. 2017, págs. 307-320.
- [235] Markus Schatten, Jurica Ševa e Igor Tomičić. «A roadmap for scalable agent organizations in the internet of everything». En: *Journal of Systems and Software* 115 (2016), págs. 31-41.
- [236] Eva-Maria Schön, Jörg Thomaschewski y María José Escalona. «Agile requirements engineering: a systematic literature review». En: *Computer Standards & Interfaces* 49 (2017), págs. 79-91.
- [237] Christoph Schroth y Till Janner. «Web 2.0 and SOA: Converging concepts enabling the internet of services». En: *IT professional* 9.3 (2007).
- [238] John Scott. *Social network analysis, third edition*. ISBN-978-1-4462-0903-5. Sage, 2012.
- [239] Jalpa Shah y Biswajit Mishra. «IoT enabled environmental monitoring system for smart cities». En: *International Conference on Internet of Things and Applications (IOTA)*. IEEE. 2016, págs. 383-388.
- [240] Onn Shehory y Arnon Sturm. «The Landscape of Agent-Oriented Methodologies». En: *Agent-oriented software engineering: reflections on architectures, methodologies, languages, and frameworks*. Springer-Verlag Berlin Heidelberg, 2014, págs. 137-154.
- [241] Zhiqi Shen et al. «Dynamic Generation of Internet of Things Organizational Structures Through Evolutionary Computing». En: *IEEE Internet of Things Journal* 5.2 (2018), págs. 943-954.
- [242] Amit Sheth, Cory Henson y Satya S Sahoo. «Semantic sensor web». En: *IEEE Internet computing* 12.4 (2008).
- [243] Weisong Shi et al. «Edge computing: Vision and challenges». En: *IEEE Internet of Things Journal* 3.5 (2016), págs. 637-646.

- [244] Vijayshree Shinde y Seema C Biday. «Comparison of Real Time Task Scheduling Algorithms». En: *International Journal of Computer Applications* 158.6 (2017), págs. 37-41.
- [245] Dhananjay Singh, Gaurav Tripathi y Antonio J Jara. «A survey of Internet-of-Things: Future vision, architecture, challenges and services». En: *2014 IEEE world forum on Internet of things (WF-IoT)*. IEEE. 2014, págs. 287-292.
- [246] Munindar P Singh y Amit K Chopra. «The Internet of Things and Multiagent Systems: Decentralized Intelligence in Distributed Computing». En: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2017, págs. 1738-1747.
- [247] Alexander Smirnov et al. «Application for e-tourism: intelligent mobile tourist guide». En: *2015 IIAI 4th International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE. 2015, págs. 40-45.
- [248] Tom Snyder y Greg Byrd. «The Internet of everything». En: *Computer* 50.6 (2017), págs. 8-9.
- [249] Javier Soriano et al. «Internet of Services». En: Springer Berlin Heidelberg, 2013. Cap. Evolution of Telecommunication Services, 283–325.
- [250] Andreas Spillner, Tilo Linz y Hans Schaefer. *Software testing foundations: a study guide for the certified tester exam, fourth edition*. ISBN-987-1-937538-42-2. Rocky Nook, Inc., 2014.
- [251] D. Sporny M. and Longley et al. *JSON-LD 1.1 A JSON-based Serialization for Linked Data*. <http://json-ld.org/spec/latest/json-ld/>. Accessed 23 January 2017. 2017.
- [252] Biljana L Risteska Stojkoska y Kire V Trivodaliev. «A review of Internet of Things for smart home: Challenges and solutions». En: *Journal of Cleaner Production* 140 (2017), págs. 1454-1464.
- [253] Anand Subramanian et al. «Real-time scheduling of distributed resources». En: *IEEE Transactions on Smart Grid* 4.4 (2013), págs. 2122-2130.

- [254] Lu Tan y Neng Wang. «Future internet: The internet of things». En: *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. Vol. 5. IEEE. 2010, págs. V5-376.
- [255] Dante I Tapia et al. «FUSION@, a SOA-based multi-agent architecture». En: *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*. Springer. 2009, págs. 99-107.
- [256] D Tektonidis et al. «Intuitive user interfaces to help boost adoption of internet-of-things and internet-of-content services for all». En: *Inter-cooperative Collective Intelligence: Techniques and Applications*. Springer, 2014, págs. 93-110.
- [257] Dimitris Tektonidis y Adamantios Koumpis. «Accessible Internet-of-Things and Internet-of-Content Services for All in the Home or on the Move». En: *International Journal of Interactive Mobile Technologies (iJIM)* 6.4 (2012), págs. 25-33.
- [258] Muthuraman Thangaraj et al. «Agent based Semantic Internet of Things (IoT) in Smart Health care». En: *Proceedings of the The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society*. ACM. 2016, pág. 41.
- [259] Roy Thomas. «Architectural styles and the design of network-based software architectures». En: *Irvine: University of California* (2000).
- [260] Johannes Thönes. «Microservices». En: *IEEE Software* 32.1 (2015), págs. 116-116.
- [261] Quynh-Nhu Numi Tran y Graham C Low. «Comparison of ten agent-oriented methodologies». En: *Agent-Oriented Methodologies*. Hershey, PA: IGI Global, 2005, págs. 341-367.
- [262] Michele Tumminelli y Steve Battle. «A Smart Contract Model for Agent Societies». En: Bristol, UK, 2017.
- [263] Pierre-Yves Vandenbussche et al. «Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web». En: *Semantic Web* 8.3 (2017), págs. 437-452.

- [264] Swagath Venkataramani, Kaushik Roy y Anand Raghunathan. «Efficient embedded learning for IoT devices». En: *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE. 2016, págs. 308-311.
- [265] Junping Wang, Qiliang Zhu y Yan Ma. «An agent-based hybrid service delivery for coordinating internet of things and 3rd party service providers». En: *Journal of Network and Computer Applications* 36.6 (2013), págs. 1684-1695.
- [266] Yi Wei y M Brian Blake. «Service-oriented computing and cloud computing: Challenges and opportunities». En: *IEEE Internet Computing* 14.6 (2010), págs. 72-75.
- [267] Mark Weiser. «The computer for the 21st century». En: *Scientific American* (1991).
- [268] Danny Weyns. *Architecture-based design of multi-agent systems*. Springer Science & Business Media, 2010.
- [269] Gary White, Vivek Nallur y Siobhán Clarke. «Quality of service approaches in IoT: A systematic mapping». En: *Journal of Systems and Software* 132 (2017), págs. 186-203.
- [270] Andrew Whitmore, Anurag Agarwal y Li Da Xu. «The Internet of Things—A survey of topics and trends». En: *Information Systems Frontiers* 17.2 (2015), págs. 261-274.
- [271] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [272] Geng Wu et al. «M2M: From mobile to embedded internet». En: *IEEE Communications Magazine* 49.4 (2011).
- [273] Qihui Wu et al. «Cognitive internet of things: a new paradigm beyond connection». En: *IEEE Internet of Things Journal* 1.2 (2014), págs. 129-143.
- [274] Feng Xia et al. «Socially aware networking: A survey». En: *IEEE Systems Journal* 9.3 (2015), págs. 904-921.
- [275] Dandan Xu, Bin Fang y Huijie Li. «An abstract communication service interface of DPWS on embedded device in industrial automation». En: *2016 Chinese Control and Decision Conference (CCDC)*. IEEE. Yinchuan, China, 2016, págs. 4695-4699.

-
- [276] X Xu, Nik Bessis y J Cao. «An autonomic agent trust model for IoT systems». En: *Procedia Computer Science* 21 (2013), págs. 107-113.
- [277] Xiaofei Xu et al. «From big data to big service». En: *Computer* 48.7 (2015), págs. 80-83.
- [278] Han Yu, Zhiqi Shen y Cyril Leung. «From internet of things to internet of agents». En: *IEEE International Conference on Green Computing and Communications (GreenCom), 2013 IEEE Internet of Things (iThings/CPSCoM) and IEEE Cyber, Physical and Social Computing*. IEEE. 2013, págs. 1054-1057.
- [279] Özgür Yürür et al. «Context-awareness for mobile sensing: A survey and future directions». En: *IEEE Communications Surveys & Tutorials* 18.1 (2016), págs. 68-93.
- [280] Andrea Zanella et al. «Internet of things for smart cities». En: *IEEE Internet of Things journal* 1.1 (2014), págs. 22-32.
- [281] Elmar Zeeb et al. «Service-oriented architectures for embedded systems using devices profile for web services». En: *AINAW'07. 21st International Conference on Advanced Information Networking and Applications Workshops*. Vol. 1. IEEE. 2007, págs. 956-963.
- [282] Deze Zeng, Song Guo y Zixue Cheng. «The web of things: A survey». En: *Journal of Communications* 6.6 (2011), págs. 424-438.
- [283] Qi Zhang, Lu Cheng y Raouf Boutaba. «Cloud computing: state-of-the-art and research challenges». En: *Journal of internet services and applications* 1.1 (2010), págs. 7-18.
- [284] Lixin Zhou y Catherine Xiaocui Lou. «Intelligent cargo tracking system based on the internet of things». En: *2012 15th International Conference on Network-Based Information Systems (NBIS)*. IEEE. 2012, págs. 489-493.