



ugr

Universidad  
de Granada

TRABAJO FIN DE GRADO  
INGENIERÍA INFORMÁTICA

# Plataforma para gestionar la asistencia al trabajo

---

**Autor:**

Jose Manuel Martínez Rincón

**Directores:**

Andrés Roldán Aranda



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, 9 de Noviembre de 2024





# Plataforma para gestionar permisos de trabajo

---

Una plataforma de gestión de permisos de trabajo  
desplegada sobre docker basada en un programa ya existente.

## **Autor**

Jose Manuel Martínez Rincón

## **Directores**

Andres María Roldán Aranda



# **Aplicación web para controlar la asistencia de los estudiantes y personal al laboratorio de Granasat. Apoyado en una terminal con un lector de huellas y tarjetas**

Jose Manuel Martínez Rincón

**Palabras clave:** controlar asistencia, centralista lector de huellas, registro de entrada y salida, reneración de reportes, departamentos, aplicación web, servicio de mensajería, gestión de base de datos, turnos de asistencia, recordatorio de asistencia,colecta de información, informes completos, encriptación de datos, seguridad de datos, control de productividad

## **Resumen**

Este proyecto busca la creación de una aplicación web que permita controlar la asistencia al laboratorio de Granasat. Esta aplicación vendrá a mejorar una aplicación con la que ya cuentan en el laboratorio, la cual no se ha actualizado desde la instalación de la centralita lectora de huellas y tarjetas. La aplicación permitirá a los usuarios registrar la entrada y la salida del establecimiento, además de poder generar reportes en los que aparezca información relacionada con el estudiante, horas transcurridas entre varios registros o los departamentos en donde esté registrado actualmente. A parte de eso se ha habilitado un servicio de mensajería entre los usuarios, se podrán mandar mensajes entre los propios usuario y a departamentos enteros. A parte de lo mencionado la plataforma web ofrece una gestión completa de la base de datos, para la creación, modificación y eliminación de todas las posibles entidades de la base de datos. Una de estas entidades son los turnos de asistencia. Horarios en los cuales se espera que los pertenecientes a dicho departamento, en caso de que no se fiche la asistencia completa dentro de los turnos de asistencia se mandará un mensaje al usuario recordando. Se facilita la colecta de información de la base de datos y de los usuarios a través de la generación de archivos con la información de la base de datos. Estos archivos consisten en informes completos de la asistencia de los usuarios ya registrados en la plataforma y generar archivos en donde se volcarán en un formato menos legible para el usuario pero adaptado al programa, ya que podremos obtener la información y llevarla a la base de datos, tanto en nuestro programa como con el programa que ya cuenta el laboratorio. En nuestro backend hemos implementado una lógica robusta y segura para albergar los datos personales de los usuarios registrados. Hemos usado encriptados para cifrar los datos y así viajen encriptados y que no estén expuestos totalmente si acceden a ellos en caso de ataque. Lo que se consigue es un mayor control de la productividad de las operaciones en el laboratorio y un control accesible para los responsables



## **Project Title: Project Subtitle**

First name, Family name (student)

**Keywords:** Attendance control, Fingerprint reader system, Entry and exit log, Report generation, Departments, Web application, Messaging service, Database management, Attendance shifts, Attendance reminder, Data collection, Complete reports, Data encryption, Data security, Productivity control

### **Abstract**

This project aims to create a web application to manage attendance at the Granasat laboratory. This application will improve upon an existing application that the laboratory currently uses, which has not been updated since the installation of the fingerprint and card reader system. The application will allow users to log their entry and exit from the facility, as well as generate reports that include information related to the student, hours elapsed between various records, and the departments in which they are currently registered. In addition, a messaging service has been enabled among users, allowing messages to be sent between individual users and entire departments. Beyond this, the web platform provides complete database management, enabling the creation, modification, and deletion of all possible database entities. One of these entities includes attendance shifts, which are schedules expected for department members. If full attendance is not recorded within these shifts, a reminder message will be sent to the user. The platform facilitates data collection from the database and users by generating files containing database information. These files consist of comprehensive reports of the attendance of users already registered on the platform and files formatted in a way that may be less readable for the user but adapted for the program. This allows the information to be obtained and transferred to the database, both in our program and in the laboratory's existing program. Our backend has been implemented with robust and secure logic to store personal data of registered users. Encryption has been used to secure data, ensuring it is transmitted in an encrypted format and not fully exposed in case of unauthorized access or an attack. This results in better control over the productivity of operations within the laboratory and accessible oversight for those in charge.





---

D. **Andres María Roldán Aranda**, Profesor del Área de electrónica del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *Plataforma para gestionar la asistencia al trabajo*, ha sido realizado bajo su supervisión por **Jose Manuel Martínez Rincón**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a X de Noviembre de 2024 .

**Los directores:**

**Andres María Roldán Aranda**



# Agradecimientos

A mi madre, por enseñarme a nunca rendirme ya pelear por lo que uno quiere hasta conseguirlo.

A mi padre, por enseñarme todo lo que sabe y por ayudarme siempre que me ha hecho falta

A mi hermana, por estar siempre presente cuando hace falta y por ser la mejor acompañante

A mis amigos de la carrera, por vivir conmigo las mejores experiencias de mi vida y por ayudarme siempre que me ha hecho falta

A mis amigos de siempre, por todo lo que me han enseñado y por todas las experiencias que hemos pasado.

A mi tutor del TFG, Andrés, por ayudarme en todo lo que me ha hecho falta y por dirigirme en este proyecto.

A mis profesores, por compartir su conocimiento y su tiempo conmigo.



# Índice general

Índice de Tablas . . . . .	19
Índice de Figuras . . . . .	19
<b>1. Introducción</b>	<b>21</b>
1.1. Motivación . . . . .	21
1.2. Objetivos . . . . .	22
1.2.1. Crear una aplicación en función de una existente . . . . .	22
1.2.2. Crear una aplicación para desplegarla luego con docker	23
1.2.3. Cumplir la legislación de protección de datos . . . . .	23
1.2.4. Usar librerías para mejorar la estética del frontend . . . . .	23
1.3. Descripción del problema . . . . .	24
1.4. Ley de protección de datos . . . . .	25
<b>2. Programa de Asistencia: Attendance Management y Centralita de control de acceso</b>	<b>27</b>
2.1. Programa de Attendance Management . . . . .	27
2.2. Dispositivo lector de huellas . . . . .	30
<b>3. Solución propuesta</b>	<b>39</b>
3.1. Formato de los archivos . . . . .	39
3.2. Requisitos funcionales . . . . .	39
3.3. Esquema de la base de Datos . . . . .	45
3.4. Entidades y Relaciones del Sistema . . . . .	45
3.4.1. Usuario . . . . .	45
3.4.2. Dispositivo . . . . .	46
3.4.3. Regla_Asistencia . . . . .	46
3.4.4. Relación Muchos a Uno: Usuario y Dispositivo . . . . .	47
3.4.5. Registro . . . . .	47
3.4.6. Relación Muchos a Muchos: Departamento . . . . .	47
3.4.7. Mensaje . . . . .	47
3.5. Casos de uso . . . . .	48

<b>4. Implementación</b>	<b>105</b>
4.1. Docker . . . . .	105
4.2. Frontend . . . . .	107
4.2.1. Alternativas . . . . .	107
4.3. Backend . . . . .	113
4.3.1. Alternativas . . . . .	113
<b>5. Presupuesto</b>	<b>119</b>
<b>6. Manual de usuario</b>	<b>121</b>
6.1. Instalación . . . . .	121
6.2. Manual de usuario . . . . .	122
<b>7. Conclusiones</b>	<b>133</b>
7.1. Futuras mejoras . . . . .	134
7.2. Reflexión final . . . . .	135
<b>8. Bibliografía</b>	<b>137</b>

# Índice de tablas

3.1. Requerimientos Funcionales del Sistema . . . . .	40
3.2. Caso de uso: Inicio de Sesión . . . . .	49
3.3. Caso de uso: Registrarse . . . . .	51
3.4. Caso de uso: Cerrar Sesión . . . . .	52
3.5. Caso de uso: Consultar datos de usuario . . . . .	53
3.6. Caso de uso: Cambiar datos del usuario . . . . .	55
3.7. Caso de uso: Consulta de datos de asistencia del usuario . . . . .	56
3.8. Caso de uso: Registro de entrada en uno de los dispositivos . . . . .	57
3.9. Caso de uso: Registro de un nuevo dispositivo . . . . .	59
3.10. Caso de uso: Eliminar un dispositivo . . . . .	61
3.11. Caso de uso: Registro de un nuevo departamento . . . . .	63
3.12. Caso de uso: Modificar el departamento . . . . .	65
3.13. Caso de uso: Dar de baja un usuario . . . . .	67
3.14. Caso de uso: Modificar un dispositivo . . . . .	69
3.15. Caso de uso: Borrar un departamento . . . . .	71
3.16. Caso de uso: Importar datos . . . . .	73
3.17. Caso de uso: Exportar datos . . . . .	75
3.18. Caso de uso: Listar Departamentos . . . . .	76
3.19. Caso de uso: Listar Dispositivos . . . . .	77
3.20. Caso de uso: Listar Usuarios . . . . .	78
3.21. Caso de uso: Conectar con un dispositivo . . . . .	79
3.22. Caso de uso: Desconectar de un dispositivo . . . . .	81
3.23. Caso de uso: Obtener informe de asistencia . . . . .	83
3.24. Caso de uso: Hacer copia de seguridad . . . . .	85
3.25. Caso de uso: Obtener horas de asistencia . . . . .	86
3.26. Caso de uso: Listar operaciones del sistema . . . . .	87
3.27. Caso de uso: Mandar mensaje a un usuario . . . . .	89
3.28. Caso de uso: Mandar mensaje a un departamento . . . . .	91
3.29. Caso de uso: Añadir un usuario a un departamento . . . . .	93
3.30. Caso de uso: Eliminar un usuario de un departamento . . . . .	95
3.31. Caso de uso: Restaurar la información de la base de datos . . . . .	97
3.32. Caso de uso: Eliminar la información de la base de datos . . . . .	99
3.33. Caso de uso: Modificar registro de asistencia . . . . .	101

3.34. Caso de uso: Eliminar un registro de asistencia de la base de datos . . . . .	103
4.1. Comparación de tecnologías . . . . .	111
4.2. Comparación de tecnologías . . . . .	116
5.1. Presupuesto estimado para el desarrollo del software . . . . .	120

# Índice de figuras

2.1. Menu de inicio del programa . . . . .	27
2.2. Menu de departamentos del programa . . . . .	28
2.3. Menu de administradores del programa . . . . .	28
2.4. Menu de gestión de turnos del programa . . . . .	28
2.5. Menu de aplicación de turnos del programa . . . . .	29
2.6. Menu de creación de turnos del programa . . . . .	29
2.7. Menu de lista de empleados del programa . . . . .	29
2.8. Menu de calculo de asistencia de empleados del programa . .	30
2.9. Menu de importación de datos del programa . . . . .	30
2.10. Menu de exportación de datos del programa . . . . .	30
2.11. Menu de inicio de la centralita . . . . .	31
2.12. Menu de inicio de la centralita . . . . .	31
2.13. Interfaz de usuario de la centralita . . . . .	32
2.14. Busqueda de Usuario por el ID . . . . .	32
2.15. Menu de departamento de la centralita . . . . .	33
2.16. Menu de turnos de la centralita . . . . .	33
2.17. Menu de modificación de turnos de la centralita . . . . .	34
2.18. Menu de mensajes de la centralita . . . . .	34
2.19. Menu de envio de mensajes de la centralita . . . . .	35
2.20. Menu de descarga de reportes de la centralita . . . . .	35
2.21. Menu de gestión de USB de la centralita . . . . .	36
2.22. Menu de gestión de la Base de Datos de la centralita . . . . .	36
2.23. Menu de descarga de reportes de la centralita . . . . .	37
2.24. Menu de acceso de la centralita . . . . .	37
3.1. Diagrama entidad-relación de la base de datos . . . . .	45
3.2. Diagrama Interacción CU1: 'Iniciar Sesión' . . . . .	50
3.3. Diagrama Interacción CU2: 'Registro' . . . . .	52
3.4. Diagrama Interacción CU3: 'Cerrar Sesion' . . . . .	53
3.5. Diagrama Interacción CU4: 'Consultar Datos Usuario' . . . . .	54
3.6. Diagrama Interacción CU5: 'Modificar Datos Usuario' . . . . .	56
3.7. Diagrama Interacción CU6: 'Consultar Datos de Asistencia del usuario' . . . . .	57

3.8. Diagrama Interacción CU7: 'Registro de entrada en un dispositivo' . . . . .	58
3.9. Diagrama Interacción CU8: 'Registro de un nuevo dispositivo' . . . . .	60
3.10. Diagrama Interacción CU9: 'Eliminar un Dispositivo' . . . . .	62
3.11. Diagrama Interacción CU10: 'Registrar un departamento nuevo' . . . . .	64
3.12. Diagrama Interacción CU11: 'Modificar un departamento existente' . . . . .	66
3.13. Diagrama Interacción CU12: 'Dar de baja un usuario' . . . . .	68
3.14. Diagrama Interacción CU13: 'Modificar un dispositivo existente' . . . . .	70
3.15. Diagrama Interacción CU14: 'Eliminar un departamento existente' . . . . .	72
3.16. Diagrama Interacción CU15: 'Importar datos' . . . . .	74
3.17. Diagrama Interacción CU16: 'Exportar datos' . . . . .	76
3.18. Diagrama Interacción CU17: 'Listar departmantos' . . . . .	77
3.19. Diagrama Interacción CU18: 'Listar dispositivos' . . . . .	78
3.20. Diagrama Interacción CU19: 'Listar usuarios' . . . . .	79
3.21. Diagrama Interacción CU20: 'Conectar a un dispositivo' . . . . .	80
3.22. Diagrama Interacción CU21: 'Desconectar de un dispositivo' . . . . .	82
3.23. Diagrama Interacción CU22: 'Obtener informe de asistencia' . . . . .	84
3.24. Diagrama Interacción CU23: 'Hacer copia de seguridad' . . . . .	86
3.25. Diagrama Interacción CU24: 'Obtener horas de asistencia' . . . . .	87
3.26. Diagrama Interacción CU25: 'Listar operaciones del sistema' . . . . .	88
3.27. Diagrama Interacción CU26: 'Mandar mensaje a un usuario' . . . . .	90
3.28. Diagrama Interacción CU27: 'Mandar mensaje a un departamento' . . . . .	92
3.29. Diagrama Interacción CU28: 'Añadir un usuario a un departamento' . . . . .	94
3.30. Diagrama Interacción CU29: 'Borrar un usuario a un departamento' . . . . .	96
3.31. Diagrama Interacción CU30: 'Restaurar base de datos' . . . . .	98
3.32. Diagrama Interacción CU31: 'Borrar base de datos' . . . . .	100
3.33. Diagrama Interacción CU32: 'Cambiar registro de asistencia' . . . . .	102
3.34. Diagrama Interacción CU33: 'Eliminar registro de asistencia' . . . . .	104
4.1. Variables de acceso a la base de datos . . . . .	105
4.2. Variables de acceso a la base de datos . . . . .	106
4.3. Variables de acceso a la base de datos . . . . .	106
4.4. Variables de acceso a la base de datos . . . . .	106
4.5. Variables de acceso a la base de datos . . . . .	107
4.6. Diagrama de llamadas en el proceso de ejecución de Docker . . . . .	107
6.1. Captura de PHPMyadmin . . . . .	121
6.2. Captura de PHPMyadmin . . . . .	122
6.3. Inicio de sesión de la aplicación . . . . .	122

---

6.4. Registro en la aplicación . . . . .	122
6.5. Página Principal 1 . . . . .	123
6.6. Página Principal 2 . . . . .	123
6.7. Página cambiar objetos . . . . .	123
6.8. Página de lista de objetos . . . . .	124
6.9. Página cambiar objetos . . . . .	125
6.10. Página lista para cambiar permisos . . . . .	126
6.11. Página cambiar privilegios . . . . .	126
6.12. Página lista de usuarios . . . . .	127
6.13. Página lista de departamentos . . . . .	128
6.14. Página confirmar operación . . . . .	128
6.15. Página generar reporte . . . . .	129
6.16. Página uso buscador . . . . .	129
6.17. Página descargar datos . . . . .	129
6.18. Página subir registros . . . . .	130
6.19. Página generar informe . . . . .	130
6.20. Página notificaciones . . . . .	131
6.21. Página operaciones del sistema . . . . .	131
6.22. Página operaciones del sistema 2 . . . . .	131



# Capítulo 1

## Introducción

Muchas empresas están últimamente pensando en cómo reforzar la seguridad de sus instalaciones para prevenir ataques físicos a sus activos, ya sean digitales o no. Para ello una técnica muy utilizada es el control de acceso a zonas clave. Este control de acceso tiene aplicaciones tanto de prevención como de labor forense en caso de tener que investigar un suceso criminal. Como labor de prevención hace que el atacante deba encontrar unas claves con las que acceder y borrar los registros, para evitar llegar a ser detectado. En cuanto al análisis forense, con estos registros de entrada y salida podemos llegar a detectar quien estaba dentro del lugar cuando ha ocurrido el incidente, para agilizar las investigaciones al saber a quien se le debe preguntar y preguntar el porqué estaba allí o porque no debía estarlo. Es más que necesario en múltiples ámbitos tener este tipo de aplicaciones que realizan estos registros de entrada y salida de una zona en concreto

### 1.1. Motivación

La centralita de control de asistencia ZKTecno K14,[15], es un dispositivo que lee las huellas, tarjetas y chips, obtiene la información del usuario asociado a dichos métodos de indentificación y crea un registro de asistencia el cual refleja la fecha, el usuario y el dispositivo donde se genera dicho registro. Tras generarlo se manda a la base de datos para almacenar dicha información. Con el artilugio venía un programa que proporciona una interfaz para realizar labores de gestión de la información, generación de archivos con la información de la base de datos ya almacenada.

La centralita tiene instalado un programa en sí para hacer una gestión de la información alvegada en la base de datos, la propia base de datos y mandar mensajes a los usuarios. Todo esto en una interfaz con un diseño muy básica pero util. El artilugio esta compuesto a grandes rasgos por los lectores de huellas y tarjetas y una pantalla pequeña. En dicha pantalla se muestra la interfáz antes mencionada. Dicha pantalla no es del todo ope-

rativa para hacer gestiones largas y complejas. Esta pensado para hacer pequeños ajustes tanto en el hardware del lector como en la base de datos.

Volviendo al programa que viene incluido con el aparato debemos mencionar que se nota que está pensado para versiones del sistema operativo Windows bastante antiguas, Windows 7 en este caso, con lo que el programa da una apariencia de obsoleto. Una de los objetivos de este proyecto es actualizar el frontend de este programa para que sea más atractivo para el usuario. Otro problema del programa que tiene actualmente el laboratorio es que las pestañas están muy sobrecargadas tanto de botones como de información que al usuario no le es totalmente útil al usuario para realizar sus tareas o funciones que se repiten tanto en la interfaz de la centralita como en el programa. Intentaremos en nuestro proyecto juntar toda la funcionalidad para acaparar la atención del usuario y dividir las funciones en la parte de las funciones del lector y la parte de gestión para la plataforma web.

Lo que se busca en general con este proyecto será replicar la funcionalidad ya existente en el programa de la centralita y el externo pero dándole un nuevo aspecto a la interfaz de usuario para hacerla más moderna y que invite al usuario a utilizarla. De paso buscaremos facilitar su uso a los usuarios que tengan interés en usarlo.

## 1.2. Objetivos

El objetivo principal de este proyecto lo hemos dejado ver en la sección anterior, pero consiste en la creación de una plataforma web la cual sea capaz de ofrecer las funcionalidades que ya ofrece el programa asociada a la terminal lectora de huellas y la propia terminal. Junto a esto crearemos una nueva interfaz de usuario más actualizada que la actual y que sea más apetecible para el usuario a la hora de utilizar el programa. Por último buscaremos facilitar la labor de usuario haciendo la interfaz clara y lo más userfriendly que podamos para que todo el mundo pueda usarla sin complicaciones.

A raíz de el objetivo global anterior hemos concretado unos objetivos más concisos para definirlos mejor.

### 1.2.1. Crear una aplicación en función de una existente

Lo primero que tenemos que hacer es replicar la funcionalidad que proporciona el software con el que ya cuenta el laboratorio. Para ello empezaremos a hacer un proceso de investigación de estos elementos para primeramente definir que funcionalidad es la que proporcionan estos programas y posteriormente ver como realizan esas funciones. Para esta última tarea obtendremos una copia provisional del programa y utilizaremos procedimientos como la

ingeniería inversa si hace falta para ver como realiza el programa sus funciones.

### 1.2.2. Crear una aplicación para desplegarla luego con docker

El programa asociado a la centralita esta desplegado en un solo equipo de los que entran y salen dentro del laboratorio. Es importante que el programa que resulte de este proyecto tenga un despliegue lo más sencillo posible. En este sentido hemos empleado la tecnología de Docker[2], la cual nos permite desplegar nuestros programas en múltiples plataformas con un solo comando. Para ello crearemos un archivo especial el cuál definirá los servicios que usará nuestra plataforma web para funcionar. A parte de eso definiremos unos scripts con comandos que iniciarán nuestras partes del código para que la aplicación funcione.

### 1.2.3. Cumplir la legislación de protección de datos

En nuestra aplicación se guarda información sensible que sería capaz de afectar a los usuarios registrados en la aplicación en caso de que esta información se filtre o sea comprometida por un ciberataque. Las leyes que velan por proteger la privacidad de los usuarios nos obligan a implementar medidas para dificultar que se de esta situación lo máximo posible. Como se han de proteger estos datos depende del tipo de dato que define en este caso las leyes pertinentes. En nuestro caso utilizaremos cifrados y el uso mínimo de estos datos para dejar los menos puntos posibles por donde puedan los atacantes hacerse con los datos de los usuarios que utilizan la plataforma web.

### 1.2.4. Usar librerías para mejorar la estética del frontend

Una de las partes mas importantes de este proyecto es la parte mas cercana del programa al usuario. Vamos a actualizar la interfaz del programa para que tenga un aspecto renovado y sea mucho más clara de lo que es ahora. Eliminaremos toda la información innecesaria que el frontend contiene ahora mismo para así centrar la atención del usuario en lo que le hace falta para usar la plataforma web comodamente. Para esto y dar estilo a nuestra plataformas utilizaremos una librería deidcada exclusivamente al frontend de la aplicación. Con esta librería conseguiremos que el usuario tenga una experiencia de uso lo más satisfactoria y eficiente posible.

### 1.3. Descripción del problema

El laboratorio GRANASAT es un lugar donde acuden varios estudiantes a realizar varios proyectos tanto de investigación como de final de master o grado. Dicho lugar a instalado una centralita de control de asistencia, la ZKTeco 14, para así intentar controlar la asistencia de los alumnos. Dicha centralita venía con un programa con el cual gestionar los datos guardado en la base de datos y para obtenerlos cuando le haga falta al administrados. Dicho programa no ha sido actualizado en mucho tiempo, con lo que puede presentar muchos problemas, listaremos algunos y los expondremos ahora.

- **Instalación Limitada:** Actualmente el programa esta instalado unicamente en uno de los ordenadores que tiene el laboratorio. Dicho ordenador esta obsoleto al igual que el programa que ejecuta. Este hecho puede llegar a ser un problema en caso de que el ordenador falle, no permitiría acceder a los datos de asistencia fuera de la centralita, la cual no esta pensada para eso. Si el ordenador falla podría crear un cuello de botella en la centralita ya que no se puede registrar la asistencia mientras se está usando la interfaz instalada en el dispositivo. Mejorando el despliegue de la aplicación web podemos solucionar este problema.
- **Diseño obsoleto:** El diseño de la interfaz es completamente arcaico y no funcional. En la pantalla principal se desaprovecha mucho espacio en información que no es util para el usuario y espacios en blanco que no se usan a no ser que se use cierta funcionalidad, lo cual no es eficiente ni util. Estos fallos en el diseño se corresponden en parte por las limitaciones de la época en la que se programó el programa. Otro problema que genera el mal uso del espacio es el tamaño de los botones que activan las funciones del programa, son muy pequeños y dificultan su uso, arreglaremos este problema para que sea mas manejable.
- **Interfaz anticuada:** A raíz del problema anterior a llevado a que la interfaz que se le presente al usuario arrastre los problemas anteriores entre otros. A parte de los problemas mencionados anteriormente podemos añadir que el diseño final no invita a utilizar la aplicación, da la sensación de antiguo y ciertos botones cuesta verlos, lo que puede llegar a frustrar al usuario. En cuanto a la interfaz de la centralita es más adecuada ya que se ven los botones mejor y los iconos son más adecuados. En nuestro proyecto intentaremos unir ambas interfaces para así no repetir funcionalidad y tener una interfaz que inivite al usuario a usarla y lo más atractiva posible
- **Seguridad obsoleta:** Según varios nédios de comunicación e instituciones las amenazas van en aumento y són cada vez mas peligrosas

[1], lo cual obliga a estar lo más actualizado posible para repeler estas nuevas amenazas. Si no actualizamos nuestras defensas estaremos facilitando el trabajo a los atacantes cuando amenazan la ciberseguridad de nuestra aplicación. En este caso el programa no ha recibido actualizaciones en mucho tiempo, con lo que es un punto de acceso por el cual pueden robar la información de los usuarios que están registrados en la plataforma web. Intentaremos arreglar esto todo lo posible añadiendo capas de protección a la aplicación como puede ser el cifrado de los datos.

## 1.4. Ley de protección de datos

Nuestro programa va a guardar datos que son capaces de identificar al usuario. Dichos datos estamos obligados a protegerlos en caso de que ocurra algún tipo de filtración ya sea por parte del código o de la base de datos. Para eso hemos empleado el cifrado AES [3]. Este cifrado nos permitirá añadir una capa de seguridad al tratamiento que hacemos de los datos. En la aplicación se utiliza lo menos posible dichos datos y cuando pasan del frontend al backend o viceversa siempre lo hacen encriptados. Además cuando almacenamos los datos en nuestra base de datos siempre están encriptados, con lo que en caso de recibir un ataque siempre contarán con una capa de protección extra.

Con las medidas antes expuestas buscamos intentar garantizar la integridad de los datos de los usuarios, la disponibilidad y otorgar el máximo grado de seguridad a estos datos. A estas medidas debemos añadir el login encriptado, permitiendo que únicamente cada usuario pueda acceder a sus datos y únicamente los usuarios con suficientes privilegios son los capaces de acceder a los datos del resto de usuario para manejar el estado de la base de datos.

La legislación de la RGPD, Reglamento General de Protección de Datos,[5] y la LOPDGDD, Ley Orgánica de Protección de Datos Personales y Garantías Digitales,[4] nos prohíben utilizar ciertos datos en nuestra plataforma salvo excepciones que no cumplimos, con lo que no los trataremos, dichos datos constituyen datos médicos, datos étnicos, biométricos y otros varios que pueden discriminar al usuario. También nos obliga a utilizar esos datos únicamente para el propósito que se tiene planeado. Cumplimos eso, ya que los datos únicamente se usarán para registrar la asistencia de los usuarios registrado al laboratorio. Desde la propia plataforma no tiene otro fin.

Concluyendo este apartado podemos afirmar que nuestra plataforma es lo más segura y robusta posible, asegurando que los datos de los usuarios son tratados para el propósito dado y de manera transparente. Evitamos el uso de datos prohibidos por la legislación e intentamos dar un grado de

seguridad lo más alto posible.





Figura 2.2: Menu de departamentos del programa

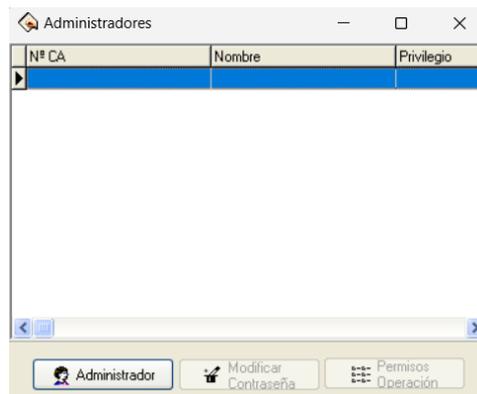


Figura 2.3: Menu de administradores del programa

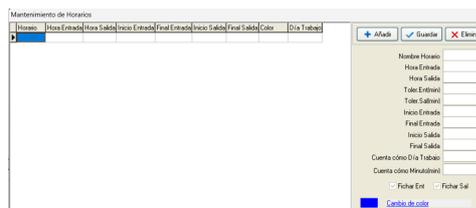


Figura 2.4: Menu de gestión de turnos del programa

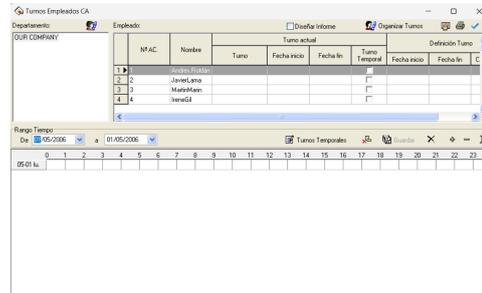


Figura 2.5: Menu de aplicación de turnos del programa

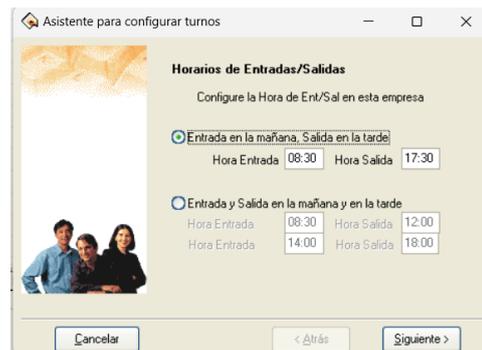


Figura 2.6: Menu de creación de turnos del programa

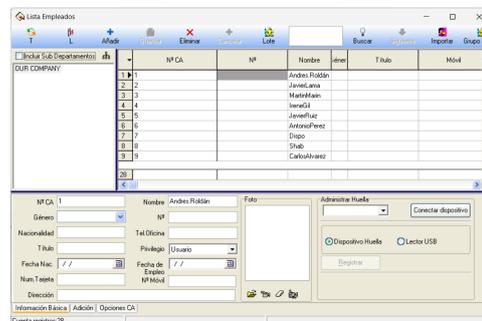


Figura 2.7: Menu de lista de empleados del programa



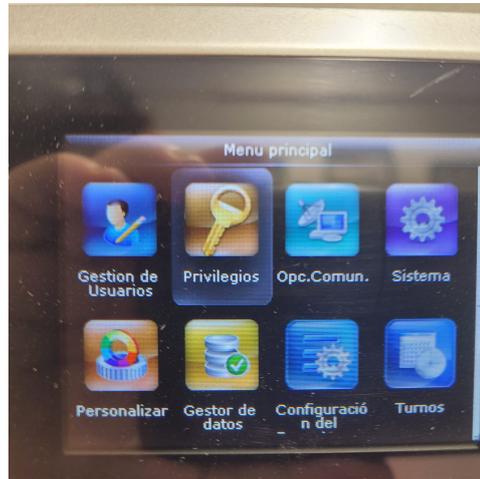


Figura 2.11: Menu de inicio de la centralita

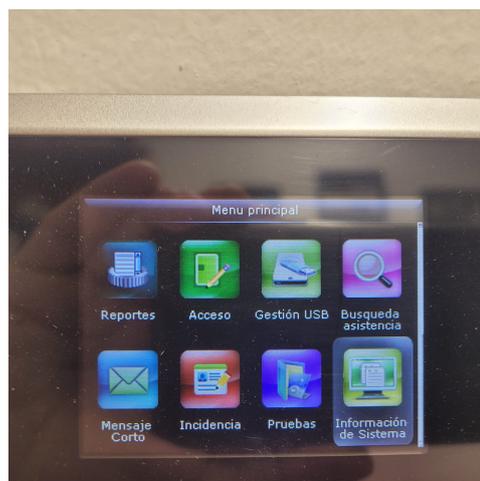


Figura 2.12: Menu de inicio de la centralita

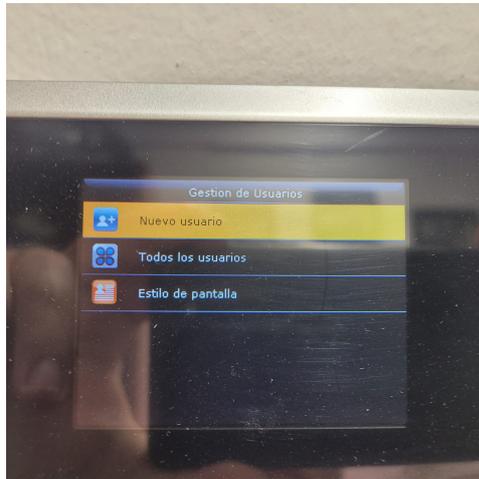


Figura 2.13: Interfaz de usuario de la centralita

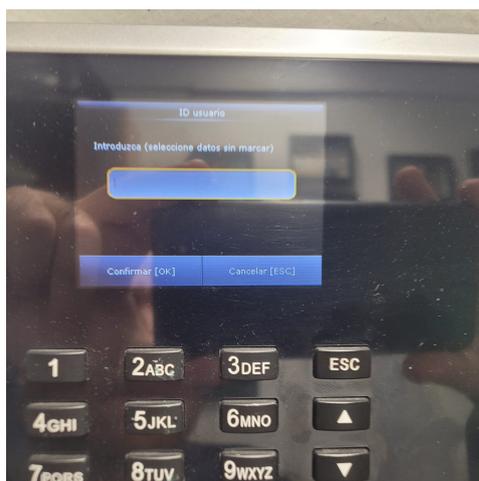


Figura 2.14: Búsqueda de Usuario por el ID

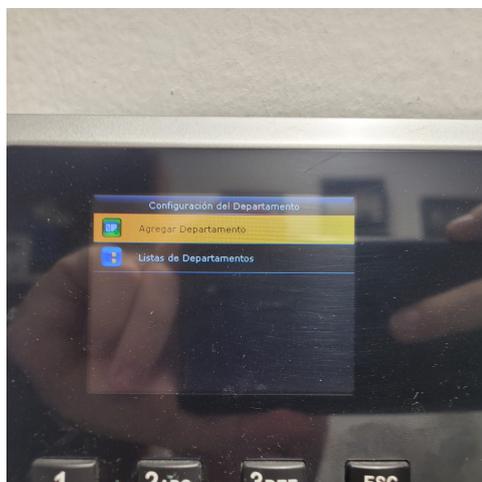


Figura 2.15: Menu de departamento de la centralita

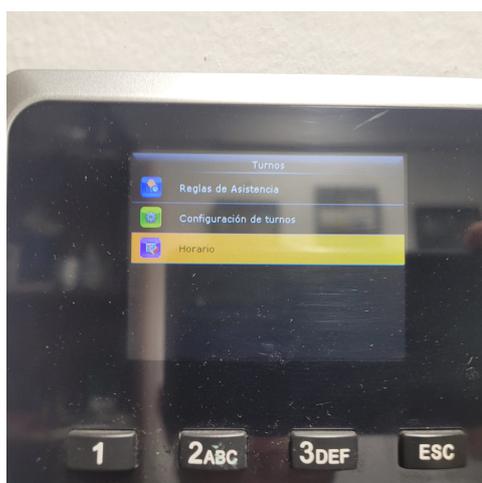


Figura 2.16: Menu de turnos de la centralita

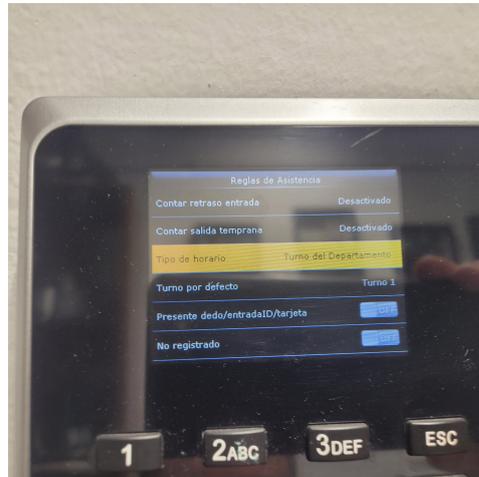


Figura 2.17: Menu de modificación de turnos de la centralita

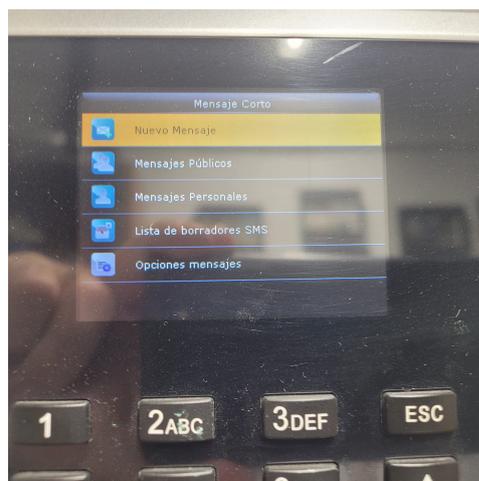


Figura 2.18: Menu de mensajes de la centralita

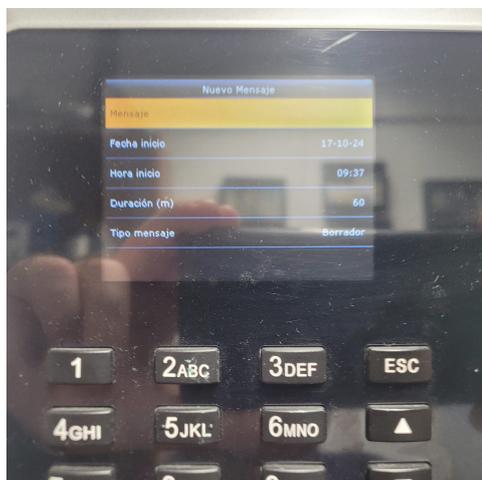


Figura 2.19: Menu de envio de mensajes de la centralita

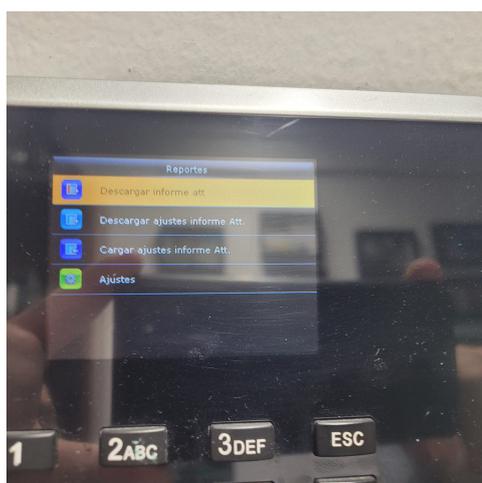


Figura 2.20: Menu de descarga de reportes de la centralita

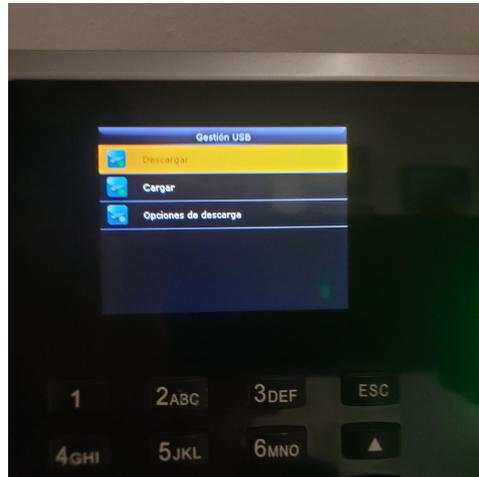


Figura 2.21: Menu de gestión de USB de la centralita

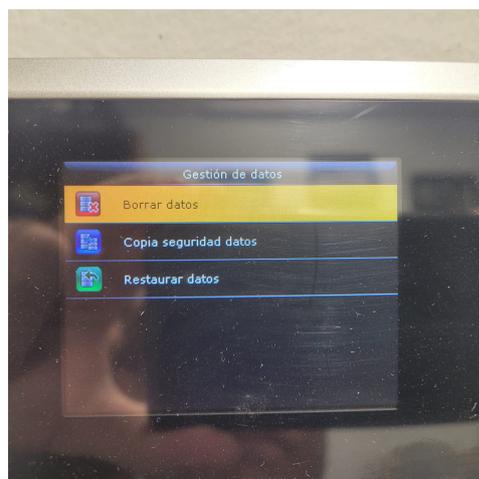


Figura 2.22: Menu de gestión de la Base de Datos de la centralita

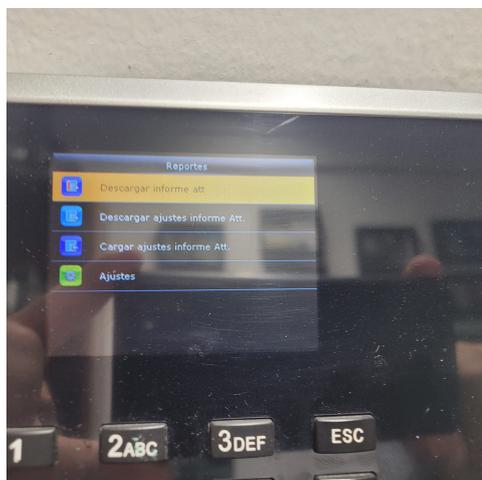


Figura 2.23: Menu de descarga de reportes de la centralita

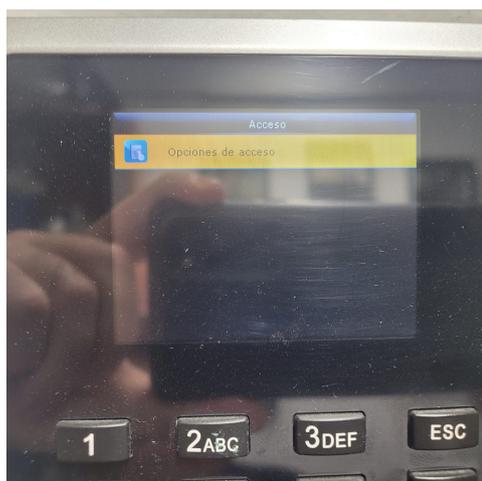


Figura 2.24: Menu de acceso de la centralita



## Capítulo 3

# Solución propuesta

### 3.1. Formato de los archivos

Como hemos mencionado anteriormente nuestro programa va a generar archivos y abrir para obtener los datos que hay en ellos. Vamos a describir a continuación las características de los archivos

- **PDF:** Este formato es usado para presentar la información de asistencia a los usuarios en un formato más legible para los usuarios.
- **CSV:** Este formato será el que utilice la aplicación para pasar los registros entre programas. Se seguirá el formato que ya viene heredado del programa ya instalado.
- **SQL:** Formato usado para las copias de seguridad de la base de datos. Contendrá los comandos SQL necesarios para poder restaurar el estado de la base de de datos.

### 3.2. Requisitos funcionales

La plataforma va a brindar al usuario una gestión básica de la base de datos además de la funcionalidad de los registros de asistencia. En este sentido el usuario podrá ver todos los registros de asistencia, departamentos, turnos de asistencia y usuarios registrados para poder manejarlos a petición y necesidad del administrador, los únicos con privilegios para todas estas acciones. A parte de todo esto todos los usuarios podrán registrar la asistencia al laboratorio, recibir y mandar mensajes a los usuarios que también esten registrados en el sistema o generar reportes para ver sus datos de asistencia. Otras funcionalidades que se le ofrecen a los administradores es la de generar informes completos de todos los usuarios del sistema, manejar la base de datos, realizar copias de seguridad o aplicar turnos de asistencia a los departamentos ya registrados. Con esto han surgido los siguientes requisitos

funcionales:

Cuadro 3.1: Requerimientos Funcionales del Sistema

ID	Nombre	Descripción	Prioridad
FR1	Inicio de sesión	Como usuario quiero poder iniciar sesión con su usuario y contraseña	Alta
FR2	Registrarse	Como usuario quiero poder registrar una cuenta dando los datos necesarios a través de la página de registro	Alta
FR3	Cerrar sesión	Como usuario quiero poder cerrar sesión del sistema	Alta
FR4	Consultar datos de usuario	Como usuario quiero poder ver mis datos personales del sistema	Alta
FR5	Modificar datos de usuario	Como usuario quiero poder modificar mis datos personales registrados en el sistema	Alta
FR6	Consultar datos de asistencia	Como usuario quiero poder consultar los registros de asistencia generados en los dispositivos	Alta
FR7	Registrar asistencia en un dispositivo	Como usuario quiero poder crear un nuevo registro de entrada en un dispositivo	Alta

<b>ID</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad</b>
FR8	Registro de un nuevo dispositivo	Como usuario quiero poder añadir nuevos dispositivos a la base de datos	Alta
FR9	Eliminar un dispositivo	Como usuario quiero poder eliminar dispositivos de registro de la base de datos	Media
FR10	Registro de un nuevo departamento	Como usuario quiero poder registrar un nuevo departamento en la base de datos	Alta
FR11	Modificar un departamento	Como usuario quiero poder modificar departamentos que ya estén registrados previamente en el sistema	Media
FR12	Eliminar un departamento	Como usuario quiero poder borrar departamentos que ya estén registrados en la base de datos previamente	Media
FR13	Dar de baja un usuario	Como usuario quiero poder eliminar cuentas de otros usuarios registrados previamente en la base de datos	Media

<b>ID</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad</b>
FR14	Modificar un dispositivo	Como usuario quiero poder cambiar los parámetros de un dispositivo registrado en el sistema	Media
FR15	Importar datos	Como usuario quiero poder obtener los datos de una hoja excel y poder registrarlos en la base de datos	Media
FR16	Extraer datos	Como usuario quiero poder importar los datos de asistencia de un usuario. En un formato CSV	Baja
FR17	Listar departamentos	Como usuario quiero poder obtener un listado de todos los departamentos registrados en el sistema	Media
FR18	Listar dispositivos	Como usuario quiero poder obtener una lista de los dispositivos registrados en la base de datos	Alta
FR19	Listar usuarios	Como usuario quiero poder obtener una lista de los usuarios registrados en el sistema	Alta

<b>ID</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad</b>
FR20	Conectar con un dispositivo	Como usuario quiero poder conectar con un dispositivo ya registrado previamente en el sistema	Alta
FR21	Desconectarse de un dispositivo	Como usuario quiero poder terminar la conexión con un dispositivo al que ya estoy conectado	Media
FR22	Obtener informe de asistencia	Como usuario quiero poder obtener un informe con los datos de asistencia del usuario registrado	Baja
FR23	Hacer copia de seguridad	Como usuario quiero poder hacer copias de seguridad de la información dentro de la base de datos	Baja
FR24	Calcular las horas de asistencia	Como usuario quiero poder calcular las horas totales entre 2 registros de asistencia de un usuario	Media
FR25	Listar operaciones del sistema	Como usuario quiero poder ver las operaciones que se pueden hacer sobre la base de datos	Media

ID	Nombre	Descripción	Prioridad
FR26	Mandar mensajes a usuarios	Como usuario quiero poder enviar mensajes a otros usuarios	Alta
FR27	Mandar mensajes a departamentos	Como usuario quiero poder enviar mensajes a todos los usuarios de un departamento	Media
FR28	Añadir personal a un departamento	Como usuario quiero añadir personal a un departamento	Alta
FR29	Eliminar personal de un departamento	Como usuario quiero eliminar personal a un departamento	Alta
FR30	Restaurar la información de la base de datos	Como usuario quiero poder restaurar la información de la base de datos desde un archivo	Baja
FR31	Eliminar información de la base de datos	Como usuario quiero poder borrar toda la información de la base de datos	Baja
FR32	Modificar registro de asistencia	Como usuario quiero poder modificar los registros de asistencia creados por el sistema	Baja
FR33	Borrar registro de asistencia	Como usuario quiero poder borrar los registros de asistencia creados por el sistema	Baja



- **Correo:** Correo electrónico del usuario. Es una cadena de texto.
- **ID Usuario:** ID interno del usuario en el sistema, identifica al usuario y es un entero.
- **Huella Digital:** Huella dactilar del usuario guardada como un objeto binario.
- **Usuario:** Cadena de texto que representa al usuario.
- **Contraseña:** Cadena con texto y números que es la contraseña del usuario, será encriptada al comprobarse y al registrarse.
- **Privilegios:** Entero que refleja el rol de usuario en la plataforma.
- **Titulación:** Cadena de texto que referencia a la carrera que estudia el usuario.
- **Foto:** Objeto binario que guarda una foto.
- **Num\_Tarjeta:** Número entero que representa el número de una tarjeta que puede leer la centralita.

### 3.4.2. Dispositivo

Dispositivos donde se registran las asistencias.

- **Nombre:** Cadena de caracteres que guarda el nombre del dispositivo.
- **ID\_Dispositivo:** ID interno del sistema que identifica al dispositivo.
- **Tipo:** Cadena de caracteres que guarda el método de registro de asistencia que ha usado el usuario.

### 3.4.3. Regla Asistencia

Turnos que se le aplican a los usuarios que pertenecen a un departamento.

- **Nombre\_Regla:** Cadena de caracteres que nombra a la regla.
- **ID\_Regla:** ID interno del sistema que distingue a la regla del resto.
- **Fecha\_inicio:** Campo de fecha y hora que guarda el momento en el que se empieza a aplicar la regla.
- **Fecha\_fin:** Campo de fecha y hora que guarda el momento en el que se termina de aplicar la regla.

#### 3.4.4. Relación Muchos a Uno: Usuario y Dispositivo

Entre las entidades **Usuario** y **Dispositivo** existe una relación de muchos a uno, que se comenta a continuación.

#### 3.4.5. Registro

Relación conformada por usuarios y dispositivos. Los dispositivos se pueden ver en varios registros, mientras que por cada registro debe haber únicamente un solo dispositivo. La misma situación ocurre con los usuarios.

- **ID\_Dispositivo:** ID del dispositivo donde se ha registrado la asistencia.
- **ID\_Usuario:** ID del usuario que ha registrado la asistencia en el dispositivo.
- **Fecha:** Fecha y hora que almacena el momento del registro de asistencia.
- **Tipo\_Verificación:** Cadena de caracteres que indica el método como se ha registrado la asistencia.
- **ID\_Registro:** Entero que distingue el registro del resto dentro del sistema.

#### 3.4.6. Relación Muchos a Muchos: Departamento

A los usuarios se les aplican los turnos a través de los departamentos, con lo que nos aparece una nueva relación de muchos a muchos, la cual se denomina **Departamento**.

- **ID\_Regla:** Entero que corresponde a la regla que se le ha aplicado al departamento.
- **Nombre:** Cadena de caracteres que nombra al departamento.
- **ID\_Usuario:** ID entero del usuario que ha sido añadido al departamento.
- **ID\_Departamento:** ID entero que distingue al departamento dentro de la base de datos.

#### 3.4.7. Mensaje

Relación entre usuarios. Un usuario puede ser emisor de varios mensajes y, a su vez, un mensaje puede ser recibido por múltiples receptores.

- **ID\_Mensaje:** Entero que distingue al mensaje dentro de la base de datos.
  
- **Mensaje:** Cadena de caracteres que corresponde al cuerpo del mensaje.
  
- **Fecha:** Fecha y hora del momento en que se mandan los mensajes, en caso de que el usuario la quiera guardar.
  
- **ID\_Emisor:** Entero que corresponde al usuario que ha enviado el mensaje.
  
- **ID\_Receptor:** Entero que corresponde al usuario que ha recibido el mensaje.

### 3.5. Casos de uso

Tras ver los requisitos funcionales que hemos comentado anteriormente vamos a proceder a explicar los Casos de uso que surgen de estos, que definirán la funcionalidad de nuestro programa. Irán acompañadas de Diagrama de Interacción

<b>Código</b>	CU.1
<b>Nombre</b>	Inicio de Sesión
<b>Descripción</b>	Inicio de sesión de un usuario en el sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.2
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El sistema pide las credenciales al usuario</li> <li>2. El usuario proporciona los datos al sistema</li> <li>3. El sistema valida las credenciales</li> <li>4. El usuario se identifica en el sistema</li> </ol>
<b>Flujo Alterno</b>	<ol style="list-style-type: none"> <li>2a. El usuario introduce credenciales incorrectas</li> <li>2b. El sistema avisa al usuario y vuelve a pedir las credenciales</li> <li>3a. El sistema no encuentra las credenciales en la base de datos</li> <li>3b. Se avisará al usuario y se volverá a pedir las credenciales</li> </ol>
<b>Precondiciones</b>	El usuario debe tener cuenta en la base de datos
<b>Poscondiciones</b>	El usuario se identifica y accede al sistema. Crea una entrada en el log correspondiente
<b>Frecuencia de Uso</b>	Muy alta
<b>Comentarios</b>	

Cuadro 3.2: Caso de uso: Inicio de Sesión

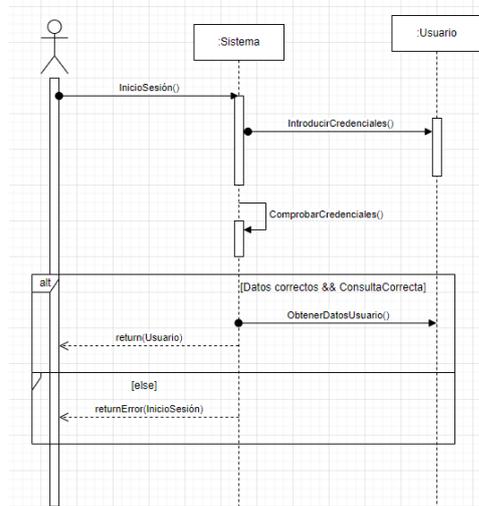


Figura 3.2: Diagrama Interacción CU1: 'Iniciar Sesión'

<b>Código</b>	CU.2
<b>Nombre</b>	Registrarse
<b>Descripción</b>	Registrarse por primera vez en el sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El sistema pide la información a través del formulario</li> <li>2. El usuario proporciona la información al sistema</li> <li>3. El sistema valida la información</li> <li>4. El sistema crea la cuenta del usuario y entra en el sistema</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario da información incorrecta</li> <li>2b. El sistema muestra un aviso y vuelve a pedir las credenciales</li> <li>3a. El sistema no consigue validar la información</li> <li>3b. El sistema lanza un aviso y vuelve a pedir las credenciales</li> </ol>
<b>Precondiciones</b>	
<b>Poscondiciones</b>	El usuario se crea una cuenta con las credenciales que indica
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	

Cuadro 3.3: Caso de uso: Registrarse

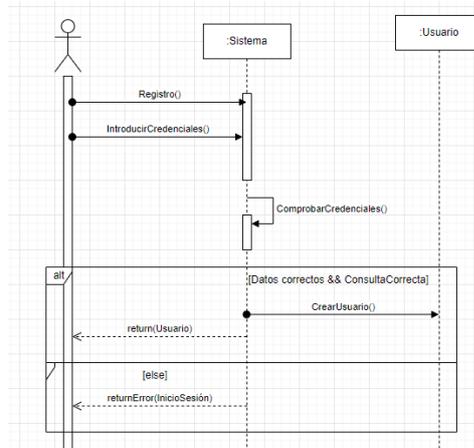


Figura 3.3: Diagrama Interacción CU2: 'Registro'

<b>Código</b>	CU.3
<b>Nombre</b>	Cerrar Sesión
<b>Descripción</b>	Cerrar sesión al salir del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón para cerrar sesión</li> <li>2. El sistema borra la sesión del usuario</li> <li>3. El sistema vuelve a mostrar la pantalla de inicio de sesión</li> </ol>
<b>Flujo Alternativo</b>	
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente
<b>Poscondiciones</b>	El usuario cierra sesión y sale del sistema, creando una entrada nueva en el log
<b>Frecuencia de Uso</b>	Muy alta
<b>Comentarios</b>	

Cuadro 3.4: Caso de uso: Cerrar Sesión

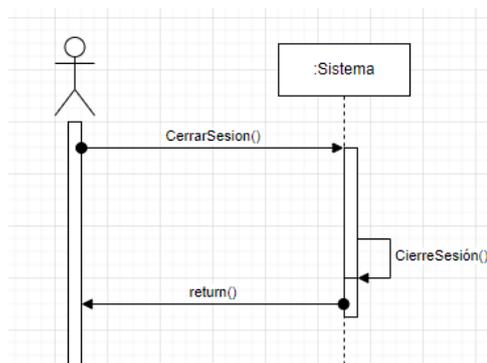


Figura 3.4: Diagrama Interacción CU3: 'Cerrar Sesión'

<b>Código</b>	CU.4
<b>Nombre</b>	Consultar datos de usuario
<b>Descripción</b>	Consulta del usuario que ha iniciado sesión de sus datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de la ventana de "Perfil"</li> <li>2. El sistema muestra una ventana formulario con la información registrada del usuario</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no encuentra la información del usuario</li> <li>2b. El sistema lanza un aviso de error</li> <li>2c. El sistema cerrará la pestaña</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente
<b>Poscondiciones</b>	El usuario consulta sus datos
<b>Frecuencia de Uso</b>	Muy alta
<b>Comentarios</b>	

Cuadro 3.5: Caso de uso: Consultar datos de usuario

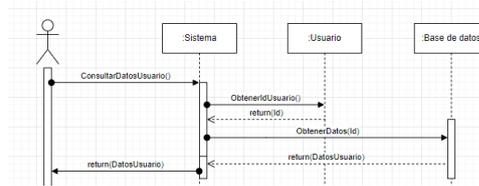


Figura 3.5: Diagrama Interacción CU4: 'Consultar Datos Usuario'

<b>Código</b>	CU.5
<b>Nombre</b>	Cambiar datos del usuario
<b>Descripción</b>	El usuario cambia sus datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pincha el botón de perfil</li> <li>2. El sistema muestra la pantalla formulario con la información del usuario</li> <li>3. El usuario escribe la información y al terminar cierra la pestaña</li> <li>4. El sistema avisa de que se ha confirmado los cambios</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3a. El usuario puso un nombre de usuario ya elegido</li> <li>3b. El sistema lo notificará y se pedirá otra vez la información</li> <li>4a. No se pudo registrar el cambio en la base de datos</li> <li>4b. Se avisa al usuario de que no se ha podido realizar el cambio y se pide al usuario que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente
<b>Poscondiciones</b>	El usuario cambia sus datos y se lleva la modificación a la base de datos
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	Por información errónea entendemos un nombre de usuario ya existente o nulo. Una contraseña nula también sería incorrecta.

Cuadro 3.6: Caso de uso: Cambiar datos del usuario

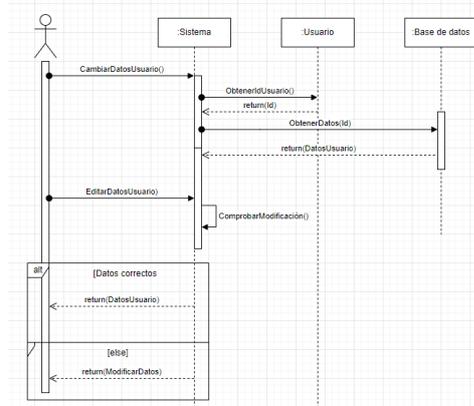


Figura 3.6: Diagrama Interacción CU5: 'Modificar Datos Usuario'

<b>Código</b>	CU.6
<b>Nombre</b>	Consulta de datos de asistencia del usuario
<b>Descripción</b>	El usuario consulta sus datos de asistencia registrados en la aplicación
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Referencias</b>	CU.1
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de Asistencia</li> <li>2. El sistema muestra una pantalla con los datos de asistencia del usuario</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. No se encuentran datos de asistencia que mostrar</li> <li>2b. Se avisará al usuario de que no existen datos que mostrar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente
<b>Poscondiciones</b>	El usuario cambia sus datos y se lleva la modificación a la base de datos
<b>Frecuencia de Uso</b>	Muy alta
<b>Comentarios</b>	

Cuadro 3.7: Caso de uso: Consulta de datos de asistencia del usuario

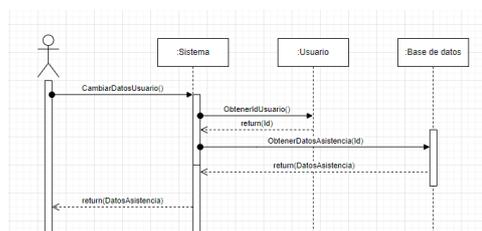


Figura 3.7: Diagrama Interacción CU6: 'Consultar Datos de Asistencia del usuario'

<b>Código</b>	CU.7
<b>Nombre</b>	Registro de entrada en uno de los dispositivos
<b>Descripción</b>	El usuario registra una entrada en la base de datos con los datos de donde y qué usuario ha entrado
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.8
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de registrar</li> <li>2. Se muestra una pantalla de registro exitoso</li> <li>3. Se cierra la pestaña anterior</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. Fallo al registrar la asistencia en el dispositivo</li> <li>2b. Se avisará al usuario y se pedirá que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente y estar conectado a un dispositivo
<b>Poscondiciones</b>	El usuario crea un registro en la base de datos y en los archivos de log con los datos correspondientes
<b>Frecuencia de Uso</b>	Muy alta
<b>Comentarios</b>	

Cuadro 3.8: Caso de uso: Registro de entrada en uno de los dispositivos

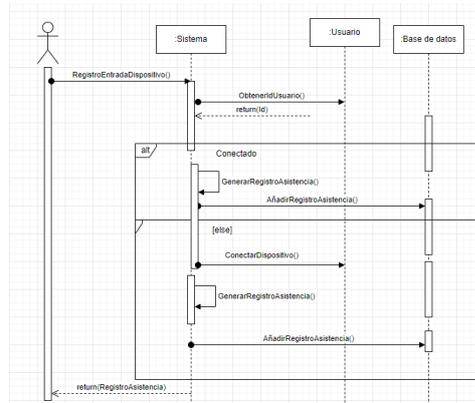


Figura 3.8: Diagrama Interacción CU7: 'Registro de entrada en un dispositivo'

<b>Código</b>	CU.8
<b>Nombre</b>	Registro de un nuevo dispositivo
<b>Descripción</b>	Administrador registra un nuevo dispositivo de entrada en la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de registro de dispositivo</li> <li>2. El sistema muestra un formulario con la información necesaria</li> <li>3. El usuario proporciona la información al sistema a través del formulario</li> <li>4. El sistema cierra la pestaña y registra el nuevo dispositivo</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3a. El usuario proporciona información errónea por el formulario</li> <li>3b. El sistema avisa al usuario y volverá a pedir la información</li> <li>4a. El sistema falla al registrar el dispositivo</li> <li>4b. Se avisará al usuario del error y volverá a pedir que lo intente</li> </ol>
<b>Precondiciones</b>	El administrador debe estar dado de alta y tener privilegios
<b>Poscondiciones</b>	El dispositivo queda registrado en la base de datos
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	Por información errónea nos referimos a información que sea igual a la de otro dispositivo registrado o a campos que son clave dejarlos vacíos, lo cual no será posible

Cuadro 3.9: Caso de uso: Registro de un nuevo dispositivo

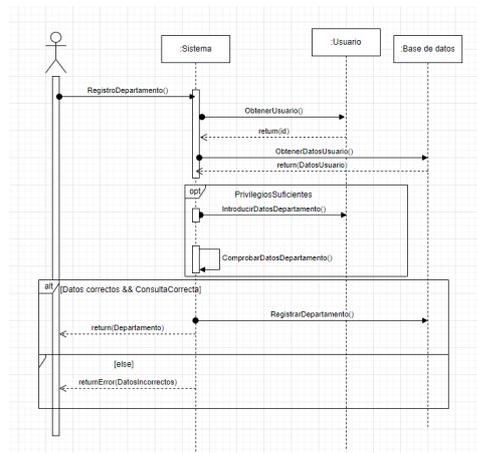


Figura 3.9: Diagrama Interacción CU8: 'Registro de un nuevo dispositivo'

<b>Código</b>	CU.9
<b>Nombre</b>	Eliminar un dispositivo
<b>Descripción</b>	Administrador quita un dispositivo de entrada de la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.8
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de dispositivos</li> <li>2. El sistema muestra una pantalla con todos los dispositivos registrados</li> <li>3. El usuario elige el dispositivo a eliminar</li> <li>4. El sistema elimina el registro del dispositivo</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. El sistema falla al eliminar el dispositivo</li> <li>4b. El sistema avisa del error al usuario y pide que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El administrador debe estar dado de alta y tener privilegios
<b>Poscondiciones</b>	El dispositivo queda eliminado en la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.10: Caso de uso: Eliminar un dispositivo

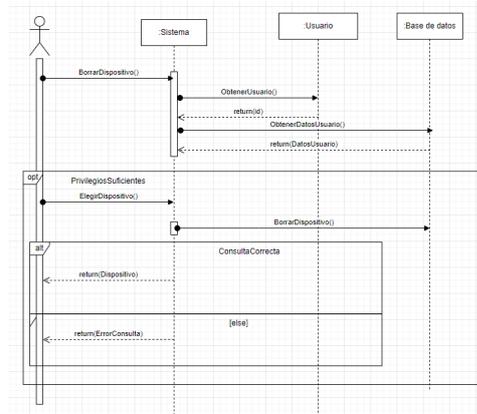


Figura 3.10: Diagrama Interacción CU9: 'Eliminar un Dispositivo'

<b>Código</b>	CU.10
<b>Nombre</b>	Registro de un nuevo departamento
<b>Descripción</b>	Administrador registra un nuevo departamento
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de registrar departamento</li> <li>2. El sistema mostrará una pantalla formulario para pedir la información al usuario</li> <li>3. El usuario introduce la información</li> <li>4. El sistema registra el departamento</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3a. El usuario introduce información incorrecta</li> <li>3b. El sistema lanzará un aviso y volverá a pedir la información al usuario</li> <li>4a. El sistema no puede registrar el departamento</li> <li>4b. Se lanza un aviso al usuario y se pide que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El administrador debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El departamento queda registrado en la base de datos
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	Por información errónea entenderemos información que se repita en otro registro

Cuadro 3.11: Caso de uso: Registro de un nuevo departamento

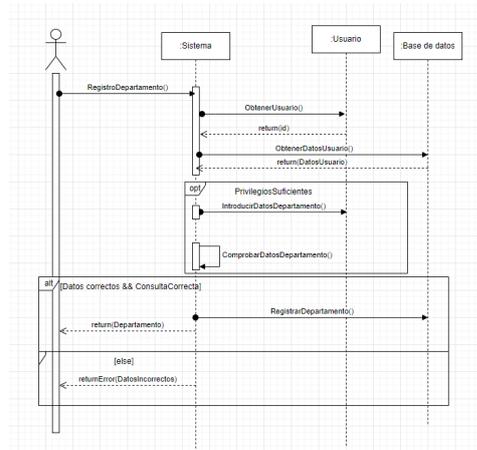


Figura 3.11: Diagrama Interacción CU10: 'Registrar un departamento nuevo'

<b>Código</b>	CU.11
<b>Nombre</b>	Modificar el departamento
<b>Descripción</b>	Administrador modifica un departamento
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Tipo</b>	Secundario
<b>Referencias</b>	CU.1, CU.10
<b>Actores</b>	Usuario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsará el botón de Departamentos</li> <li>2. El sistema mostrará una pantalla con todos los departamentos registrados</li> <li>3. El usuario modifica el departamento</li> <li>4. El sistema registra el cambio en el sistema</li> <li>5. El sistema avisa de que la operación se ha realizado</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. No encuentra departamentos registrados</li> <li>2b. Se avisará al usuario en este caso</li> <li>3a. El usuario introduce información incorrecta</li> <li>3b. Se informará al usuario y se volverá a pedir la información</li> <li>4a. El sistema no consigue hacer la operación</li> <li>4b. Se avisará al usuario y se pedirá que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El administrador debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El departamento queda modificado en la base de datos
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	

Cuadro 3.12: Caso de uso: Modificar el departamento

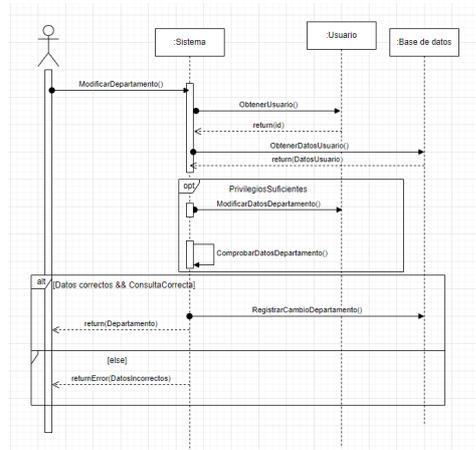


Figura 3.12: Diagrama Interacción CU11: 'Modificar un departamento existente'

<b>Código</b>	CU.12
<b>Nombre</b>	Dar de baja un usuario
<b>Descripción</b>	Eliminar un usuario de la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Tipo</b>	Secundario
<b>Referencias</b>	CU.1, CU.2
<b>Actores</b>	Usuario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de usuarios y se muestra la lista de usuarios registrados</li> <li>2. El usuario elige la cuenta a eliminar</li> <li>3. El sistema elimina la cuenta del usuario</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario elige una cuenta que no puede eliminar</li> <li>2b. El sistema avisará al usuario de que no puede eliminar esa cuenta</li> <li>3a. El sistema no consigue completar el borrado</li> <li>3b. El sistema avisará al usuario y le pedirá que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El administrador debe haber iniciado sesión y tener privilegios.
<b>Poscondiciones</b>	El usuario pierde su cuenta en el sistema
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.13: Caso de uso: Dar de baja un usuario

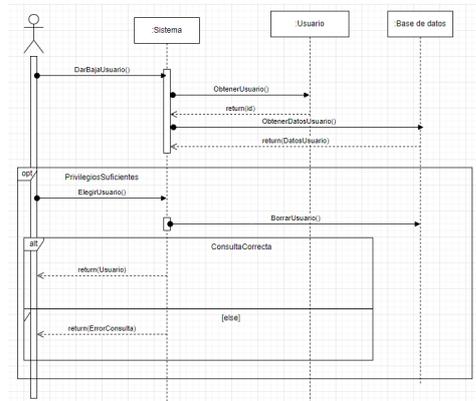


Figura 3.13: Diagrama Interacción CU12: 'Dar de baja un usuario'

<b>Código</b>	CU.13
<b>Nombre</b>	Modificar un dispositivo
<b>Descripción</b>	Modificar datos de un dispositivo
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.8
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón dispositivos y elige el dispositivo a modificar</li> <li>2. El sistema mostrará una pantalla con los dispositivos registrados</li> <li>3. El usuario elige el dispositivo a modificar</li> <li>4. El usuario modifica los datos del dispositivo</li> <li>5. El sistema registra los cambios en el sistema</li> </ol>
<b>Flujo Alterno</b>	<ol style="list-style-type: none"> <li>4a. El usuario introduce información errónea</li> <li>4b. El sistema avisará al usuario y volverá a pedir la información</li> <li>5a. El sistema no consigue registrar el cambio</li> <li>5b. El sistema avisará al usuario y volverá a pedir el cambio</li> </ol>
<b>Precondiciones</b>	El administrador debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El dispositivo queda modificado en la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.14: Caso de uso: Modificar un dispositivo

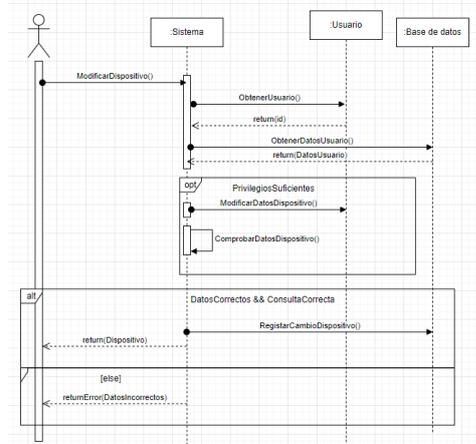


Figura 3.14: Diagrama Interacción CU13: 'Modificar un dispositivo existente'

<b>Código</b>	CU.14
<b>Nombre</b>	Borrar un departamento
<b>Descripción</b>	Eliminar un departamento de la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.10
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de departamentos</li> <li>2. El sistema muestra una pantalla con todos los departamentos registrados</li> <li>3. El usuario elige el departamento</li> <li>4. El sistema elimina el registro de la base de datos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>3a. El sistema no consigue completar el borrado</li> <li>3b. El sistema avisará al usuario y le volverá a pedir la operación</li> </ol>
<b>Precondiciones</b>	El administrador debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El departamento deja de existir en la base de datos y en el sistema, se borrará todo lo relacionado con él
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.15: Caso de uso: Borrar un departamento

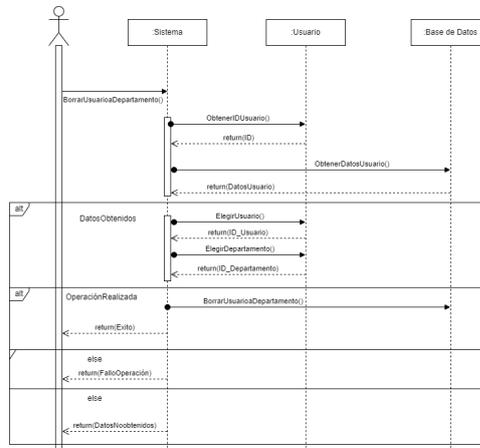


Figura 3.15: Diagrama Interacción CU14: 'Eliminar un departamento existente'

<b>Código</b>	CU.15
<b>Nombre</b>	Importar datos
<b>Descripción</b>	Importar los datos desde la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario proporciona el archivo al sistema</li> <li>2. El sistema comprueba que el archivo sea correcto</li> <li>3. El sistema abre el archivo</li> <li>4. El sistema obtiene los datos del archivo</li> <li>5. El sistema avisa al usuario al terminar la operación</li> </ol>
<b>Flujo Alterno</b>	<ol style="list-style-type: none"> <li>2a. El sistema ve que el archivo no es correcto</li> <li>2b. El sistema volverá a pedir un archivo al usuario</li> <li>3a. El sistema no consigue abrir el archivo</li> <li>3b. El sistema avisará al usuario y volverá a pedir el archivo</li> <li>4a. El sistema no consigue leer los datos del archivo</li> <li>4b. El sistema avisa al usuario y volverá a pedir el archivo</li> </ol>
<b>Precondiciones</b>	Usuario o administrador deben haber iniciado sesión
<b>Poscondiciones</b>	Los datos de asistencia del usuario se copian en un archivo que se puede descargar
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.16: Caso de uso: Importar datos

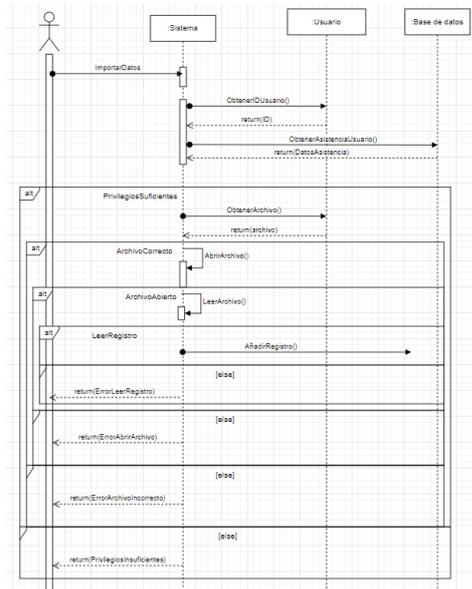


Figura 3.16: Diagrama Interacción CU15: 'Importar datos'

<b>Código</b>	CU.16
<b>Nombre</b>	Exportar datos
<b>Descripción</b>	Exportar los datos desde un archivo
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsará el botón de Exportar</li> <li>2. El sistema cogerá los datos de asistencia del usuario actual</li> <li>3. El sistema pondrá los datos en un archivo formato CSV</li> <li>4. El sistema descarga el documento en la ruta</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario no tiene datos para poner en el documento</li> <li>2b. El sistema avisará de que no hay datos para poner en el documento</li> </ol>
<b>Precondiciones</b>	Usuario debe haber iniciado sesión
<b>Poscondiciones</b>	Los datos del archivo pasarán a la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.17: Caso de uso: Exportar datos

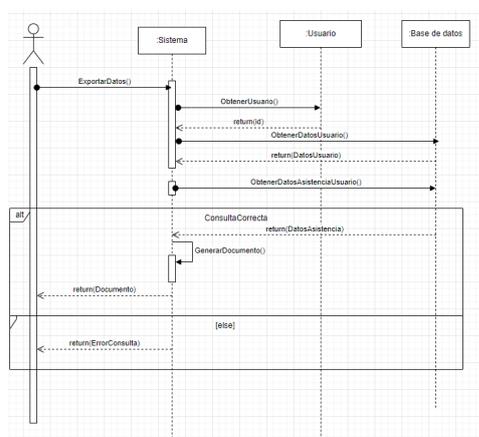


Figura 3.17: Diagrama Interacción CU16: 'Exportar datos'

<b>Código</b>	CU.17
<b>Nombre</b>	Listar Departamentos
<b>Descripción</b>	Obtener una lista de los departamentos registrados en el sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón departamentos</li> <li>2. El sistema mostrará los departamentos registrados en el sistema</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. No hay departamentos registrados en el sistema</li> <li>2b. El sistema mostrará un aviso de que no existen departamentos registrados</li> </ol>
<b>Precondiciones</b>	El usuario ha de haber iniciado sesión
<b>Poscondiciones</b>	El usuario obtiene una lista de los departamentos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.18: Caso de uso: Listar Departamentos

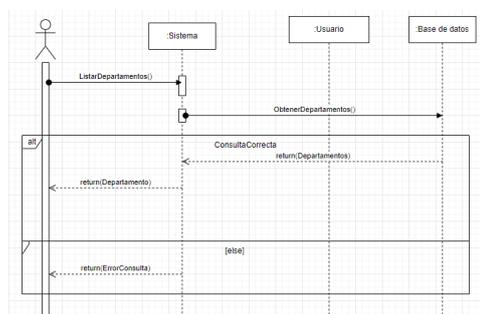


Figura 3.18: Diagrama Interacción CU17: 'Listar departmantos'

<b>Código</b>	CU.18
<b>Nombre</b>	Listar Dispositivos
<b>Descripción</b>	Obtener una lista de los dispositivos registrados en el sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón dispositivos</li> <li>2. El sistema da una pantalla con una lista de dispositivos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no encuentra dispositivos registrados</li> <li>2b. El sistema avisará al usuario de que no existen dispositivos</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión en el sistema
<b>Poscondiciones</b>	El usuario ve una lista de los dispositivos registrados en la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.19: Caso de uso: Listar Dispositivos

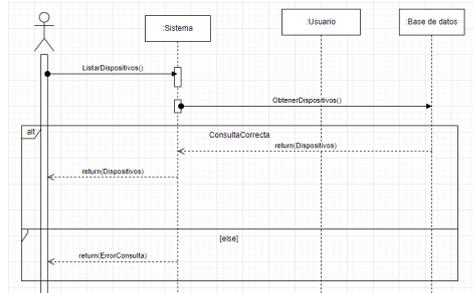


Figura 3.19: Diagrama Interacción CU18: 'Listar dispositivos'

<b>Código</b>	CU.19
<b>Nombre</b>	Listar Usuarios
<b>Descripción</b>	Obtener una lista de los usuarios registrados en el sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pincha el botón de usuarios</li> <li>2. El sistema da una pantalla con la lista de usuarios del sistema</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no encuentra usuarios registrados en el sistema</li> <li>2b. El sistema lanza un aviso al usuario de que no existen registros</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios para acceder a los datos
<b>Poscondiciones</b>	El usuario obtiene una lista de los usuarios registrados en la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.20: Caso de uso: Listar Usuarios

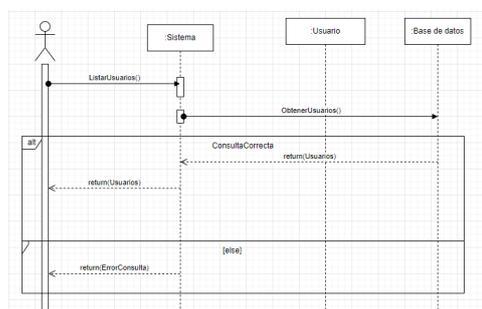


Figura 3.20: Diagrama Interacción CU19: 'Listar usuarios'

<b>Código</b>	CU.20
<b>Nombre</b>	Conectar con un dispositivo
<b>Descripción</b>	El usuario se conecta a un dispositivo
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de dispositivos</li> <li>2. El sistema le ofrece el botón de conectar</li> <li>3. El usuario pulsa el botón</li> <li>4. El sistema se conecta al dispositivo elegido</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. El sistema no consigue conectar con el dispositivo</li> <li>4b. El sistema lanza un aviso y pide al usuario que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión primero
<b>Poscondiciones</b>	El usuario se conecta a un dispositivo
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.21: Caso de uso: Conectar con un dispositivo

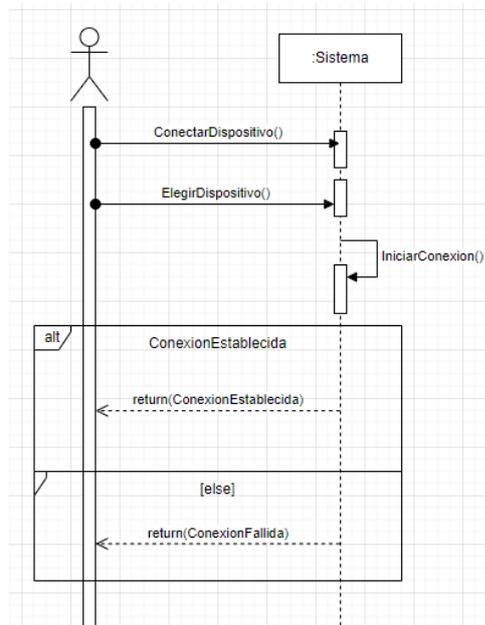


Figura 3.21: Diagrama Interacción CU20: 'Conectar a un dispositivo'

<b>Código</b>	CU.21
<b>Nombre</b>	Desconectar de un dispositivo
<b>Descripción</b>	Finalizar la conexión con un dispositivo ya conectado
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU:20
<b>Actores</b>	Usuario
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de desconectar</li> <li>2. El sistema avisa al usuario de que se ha desconectado del dispositivo</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no consigue desconectarse del dispositivo</li> <li>2b. El sistema avisará al usuario y pedirá que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y haberse conectado a un dispositivo
<b>Poscondiciones</b>	El usuario finaliza la conexión con el dispositivo
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.22: Caso de uso: Desconectar de un dispositivo

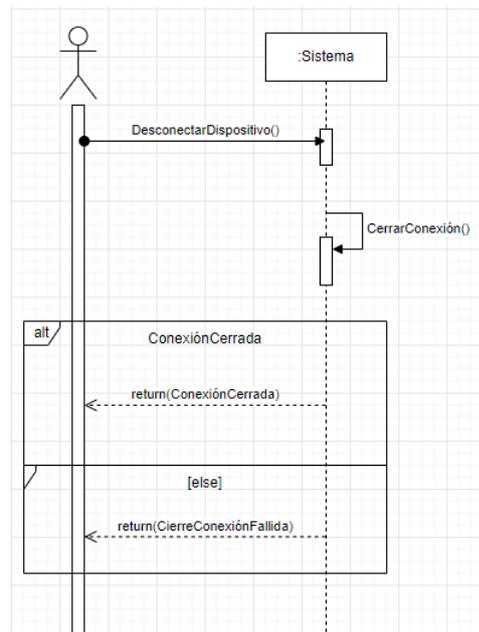


Figura 3.22: Diagrama Interacción CU21: 'Desconectar de un dispositivo'

<b>Código</b>	CU.22
<b>Nombre</b>	Obtener informe de asistencia
<b>Descripción</b>	Obtener un documento con información de asistencia del usuario
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón informe</li> <li>2. El sistema muestra una ventana para que el usuario elija dónde descargar el documento</li> <li>3. El usuario elige la ubicación y descarga el documento</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no puede generar el documento en formato PDF</li> <li>2b. Se avisará al usuario y se pedirá que vuelva a intentar generar el documento</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión
<b>Poscondiciones</b>	El usuario obtiene un informe de su asistencia
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.23: Caso de uso: Obtener informe de asistencia

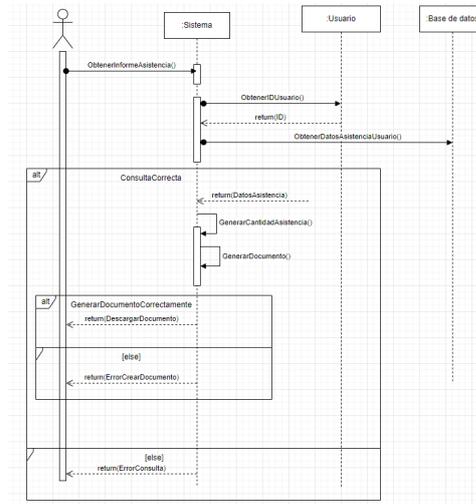


Figura 3.23: Diagrama Interacción CU2: 'Obtener informe de asistencia'

<b>Código</b>	CU.23
<b>Nombre</b>	Hacer copia de seguridad
<b>Descripción</b>	Realizar una copia de la información de la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Administrador
<b>Tipo</b>	Primario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de copia</li> <li>2. El usuario volverá a autenticarse para acceder a las operaciones</li> <li>3. El usuario pulsa el botón de Copia de Seguridad</li> <li>4. El sistema copia todos los datos registrados en la base de datos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario no se autentica correctamente</li> <li>2b. Se mostrará un aviso al usuario para que vuelva a intentarlo</li> <li>4a. El sistema no puede conectarse a la base de datos</li> <li>4b. El sistema avisará al usuario y le pedirá que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios para dicha operación
<b>Poscondiciones</b>	El usuario obtiene una copia de la base de datos
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.24: Caso de uso: Hacer copia de seguridad

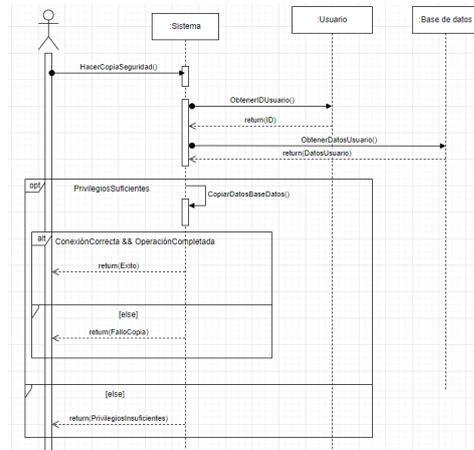


Figura 3.24: Diagrama Interacción CU23: 'Hacer copia de seguridad'

<b>Código</b>	CU.24
<b>Nombre</b>	Obtener horas de asistencia
<b>Descripción</b>	Cálculo del tiempo transcurrido entre 2 registros, sumado todo al final
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de</li> <li>2. El usuario obtiene el tiempo transcurrido entre los 2 últimos registros</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El sistema no puede acceder a la base de datos</li> <li>2b. El sistema avisará al usuario para que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente
<b>Poscondiciones</b>	El usuario obtiene el tiempo transcurrido entre 2 registros
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.25: Caso de uso: Obtener horas de asistencia

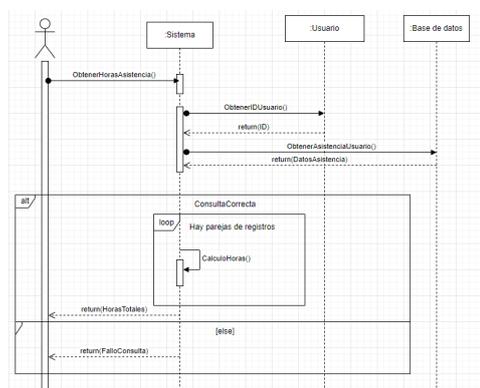


Figura 3.25: Diagrama Interacción CU24: 'Obtener horas de asistencia'

<b>Código</b>	CU.25
<b>Nombre</b>	Listar operaciones del sistema
<b>Descripción</b>	Obtener una lista de operaciones posibles del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de operaciones del sistema</li> <li>2. El usuario se autentica para acceder a las operaciones</li> <li>3. El sistema ofrece una pantalla con las operaciones posibles del sistema</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario no se autentica correctamente</li> <li>2b. El sistema le avisará de que</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El usuario ve una pantalla con las posibles operaciones del sistema
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.26: Caso de uso: Listar operaciones del sistema

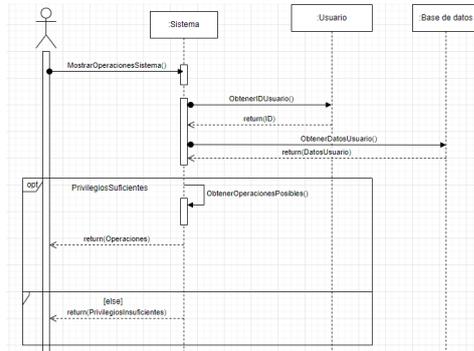


Figura 3.26: Diagrama Interacción CU25: 'Listar operaciones del sistema'

<b>Código</b>	CU.26
<b>Nombre</b>	Mandar mensaje a un usuario
<b>Descripción</b>	Mandar un mensaje a un usuario dentro del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de mensajes</li> <li>2. El sistema ofrece una pantalla con los tipos de mensajes disponibles</li> <li>3. El usuario pulsa en los mensajes relacionados con los usuarios</li> <li>4. El usuario proporciona el cuerpo del mensaje, el usuario al que va dirigido o la fecha, en caso de que se requiera</li> <li>5. El mensaje se envía a los usuarios elegidos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. No se guardan los datos correctamente del mensaje</li> <li>4b. El sistema avisará al usuario para que lo vuelva a intentar</li> <li>5a. Fallo en el envío del mensaje</li> <li>5b. El sistema avisará al usuario para que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión
<b>Poscondiciones</b>	El usuario manda un mensaje a los usuarios elegidos
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	

Cuadro 3.27: Caso de uso: Mandar mensaje a un usuario

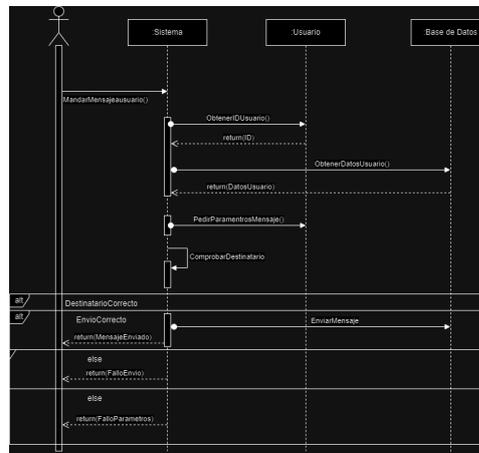


Figura 3.27: Diagrama Interacción CU26: 'Mandar mensaje a un usuario'

<b>Código</b>	CU.27
<b>Nombre</b>	Mandar mensaje a un departamento
<b>Descripción</b>	Mandar un mensaje a todos los usuarios de un departamento dentro del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de mensajes</li> <li>2. El sistema ofrece una pantalla con los tipos de mensajes disponibles</li> <li>3. El usuario pulsa en los mensajes relacionados con los usuarios</li> <li>4. El usuario proporciona el cuerpo del mensaje y el departamento al que va dirigido el mensaje</li> <li>5. El mensaje se envía a los usuarios del departamento elegido</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>4a. La información del mensaje no se guarda correctamente</li> <li>4b. El sistema avisará al usuario para que lo vuelva a intentar</li> <li>5a. El mensaje no se manda correctamente</li> <li>5b. El sistema avisará al usuario para que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión
<b>Poscondiciones</b>	El usuario manda un mensaje a los usuarios pertenecientes al departamento
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	

Cuadro 3.28: Caso de uso: Mandar mensaje a un departamento

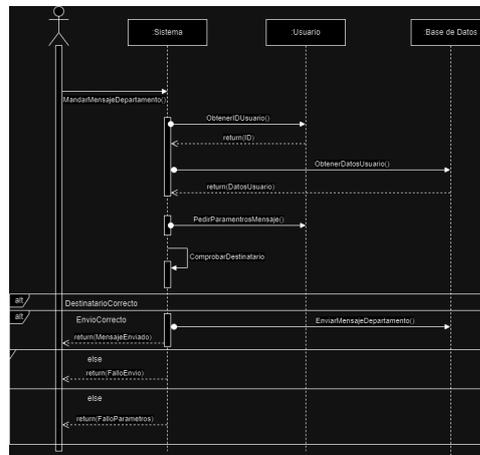


Figura 3.28: Diagrama Interacción CU27: 'Mandar mensaje a un departamento'

<b>Código</b>	CU.28
<b>Nombre</b>	Añadir un usuario a un departamento
<b>Descripción</b>	Relacionar el usuario con el departamento dentro del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El sistema ofrece una lista de usuarios registrados en el sistema</li> <li>2. El usuario pulsa en el usuario a añadir</li> <li>3. El sistema ofrece una lista de los departamentos registrados en el sistema</li> <li>4. El usuario elige el departamento al que añadir el usuario</li> <li>5. El usuario confirma si añade el usuario al departamento</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>1a. El sistema falla al mostrar los usuarios registrados</li> <li>1b. El sistema avisará al usuario del fallo y pedirá recargar la página</li> <li>3a. El sistema falla al mostrar los departamentos registrados</li> <li>3b. El sistema avisará al usuario del fallo y pedirá recargar la página</li> <li>5a. El sistema falla al añadir el usuario al departamento elegido</li> <li>5b. El sistema avisará al usuario y pedirá que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios. El usuario y el departamento han de estar creados previamente
<b>Poscondiciones</b>	El usuario elegido queda unido al departamento
<b>Frecuencia de Uso</b>	Alta
<b>Comentarios</b>	

Cuadro 3.29: Caso de uso: Añadir un usuario a un departamento

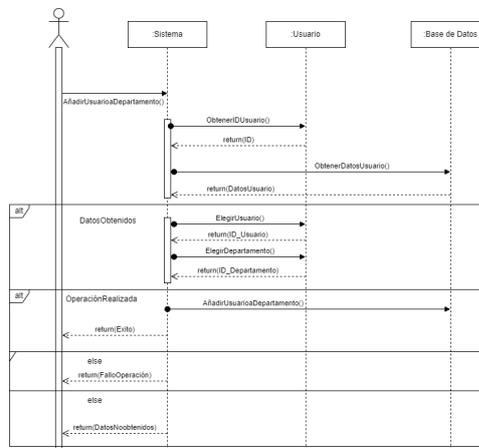


Figura 3.29: Diagrama Interacción CU28: 'Añadir un usuario a un departamento'

<b>Código</b>	CU.29
<b>Nombre</b>	Eliminar un usuario de un departamento
<b>Descripción</b>	Obtener una lista de operaciones posibles del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El sistema ofrece una lista de usuarios registrados en el sistema</li> <li>2. El usuario pulsa en el usuario a añadir</li> <li>3. El sistema ofrece una lista de los departamentos registrados en el sistema</li> <li>4. El usuario elige el departamento al que añadir el usuario</li> <li>5. El usuario confirma si eliminar al usuario del departamento</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>1a. El sistema falla al mostrar los usuarios registrados</li> <li>1b. El sistema avisará al usuario del fallo y pedirá recargar la página</li> <li>3a. El sistema falla al mostrar los departamentos registrados</li> <li>3b. El sistema avisará al usuario del fallo y pedirá recargar la página</li> <li>5a. El sistema falla al eliminar el usuario del departamento elegido</li> <li>5b. El sistema avisará al usuario y pedirá que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios. El usuario debe estar unido al departamento previamente
<b>Poscondiciones</b>	El usuario queda desligado del departamento al que estaba unido
<b>Frecuencia de Uso</b>	Media
<b>Comentarios</b>	

Cuadro 3.30: Caso de uso: Eliminar un usuario de un departamento

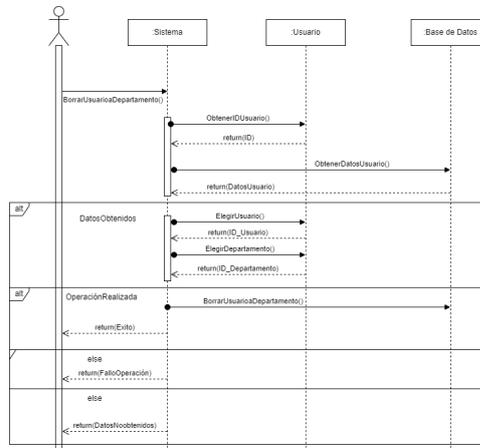


Figura 3.30: Diagrama Interacción CU29: 'Borrar un usuario a un departamento'

<b>Código</b>	CU.30
<b>Nombre</b>	Restaurar la información de la base de datos
<b>Descripción</b>	Mandar un mensaje a todos los usuarios de un departamento dentro del sistema
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de Operaciones de la base de Datos</li> <li>2. El usuario se autentica para acceder a las operaciones</li> <li>3. El usuario pulsa el botón de Restaurar Base de Datos</li> <li>4. El usuario proporciona el archivo correspondiente</li> <li>5. El sistema copia la información del archivo en la base de datos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario falla al autenticarse</li> <li>2b. El sistema avisará al usuario para que lo vuelva a intentar</li> <li>4a. Fallo al obtener el archivo que proporciona el usuario</li> <li>4b. El sistema avisará al usuario y volverá a pedir el archivo</li> <li>5a. El sistema falla al copiar la información del archivo en la base de datos</li> <li>5b. El sistema avisará al usuario para que vuelva a intentarlo</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios. La base de datos ha de haber sido vaciada previamente.
<b>Poscondiciones</b>	El usuario devuelve el estado de la base de datos al estado que corresponde al archivo con el que se ha restaurado.
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	El archivo generado tendrá el formato SQL, con toda la información correspondiente al momento en el cual se hizo la copia de seguridad.

Cuadro 3.31: Caso de uso: Restaurar la información de la base de datos

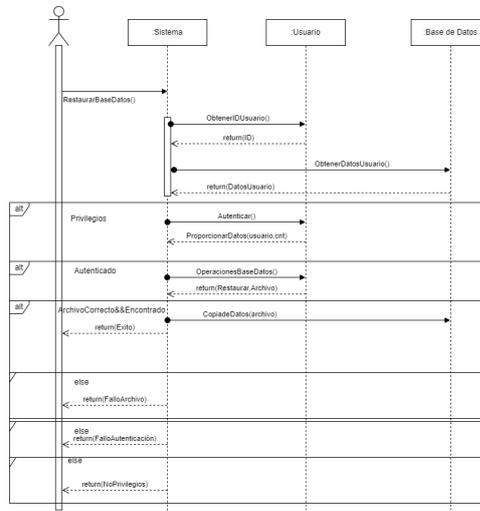


Figura 3.31: Diagrama Interacción CU30: 'Restaurar base de datos'

<b>Código</b>	CU.31
<b>Nombre</b>	Eliminar la información de la base de datos
<b>Descripción</b>	Eliminar todos los datos registrados en la base de datos
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de Operaciones de la base de Datos</li> <li>2. El usuario se autentica para acceder a las operaciones</li> <li>3. El usuario pulsa el botón de Borrar Base de Datos</li> <li>4. El usuario confirma que quiere borrar la base de datos</li> <li>5. El sistema borra toda la información existente en la base de datos</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El usuario falla al autenticarse</li> <li>2b. El sistema avisará al usuario para que lo vuelva a intentar</li> <li>4a. El sistema falla a la hora de borrar el contenido de la base de datos</li> <li>4b. El sistema avisará al usuario para que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El usuario ve una pantalla con las posibles operaciones del sistema
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.32: Caso de uso: Eliminar la información de la base de datos

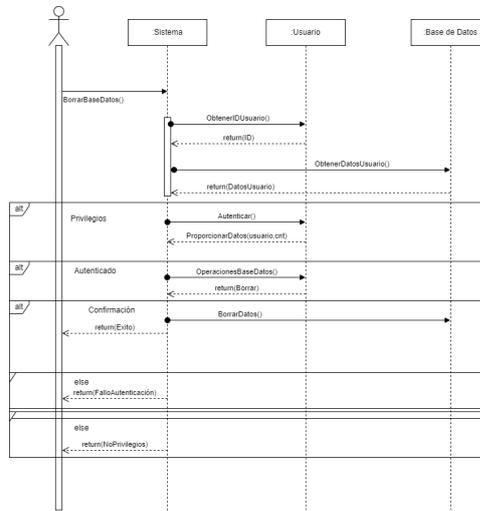


Figura 3.32: Diagrama Interacción CU31: 'Borrar base de datos'

<b>Código</b>	CU.32
<b>Nombre</b>	Modificar registro de asistencia
<b>Descripción</b>	Cambiar los parámetros de un registro de asistencia
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.7
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de registros</li> <li>2. El sistema muestra una pantalla con todos los registros</li> <li>3. El usuario elige el registro a modificar</li> <li>4. El usuario cambia los datos del registro</li> <li>5. El sistema guarda el cambio en el registro</li> </ol>
<b>Flujo Alterno</b>	<ol style="list-style-type: none"> <li>2a. El sistema falla al mostrar los registros</li> <li>2b. El sistema mostrará una pantalla en blanco y avisará al usuario</li> <li>4a. No se registra el cambio de parámetros</li> <li>4b. Se avisará al usuario del error y se pedirá que lo vuelva a hacer</li> <li>5a. El sistema no consigue guardar el cambio de registro</li> <li>5b. El sistema avisará al usuario del error y pedirá que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El usuario cambia los datos de un registro de asistencia
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.33: Caso de uso: Modificar registro de asistencia



<b>Código</b>	CU.33
<b>Nombre</b>	Eliminar un registro de asistencia de la base de datos
<b>Descripción</b>	Eliminar un registro de asistencia de los usuarios
<b>Autor</b>	Jose Manuel Martínez Rincón
<b>Referencias</b>	CU.1, CU.7
<b>Actores</b>	Usuario
<b>Tipo</b>	Secundario
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón de registros</li> <li>2. El sistema muestra una pantalla con todos los registros</li> <li>3. El usuario elige el registro a eliminar</li> <li>4. El sistema borra el registro</li> </ol>
<b>Flujo Alterno</b>	<ol style="list-style-type: none"> <li>2a. El sistema falla al mostrar los registros</li> <li>2b. Se mostrará una pantalla en blanco y se avisará al usuario</li> <li>4a. Error al borrar el registro</li> <li>4b. Se avisará al usuario y se pedirá que lo vuelva a intentar</li> </ol>
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener privilegios
<b>Poscondiciones</b>	El usuario elimina un registro de asistencia de un usuario
<b>Frecuencia de Uso</b>	Baja
<b>Comentarios</b>	

Cuadro 3.34: Caso de uso: Eliminar un registro de asistencia de la base de datos

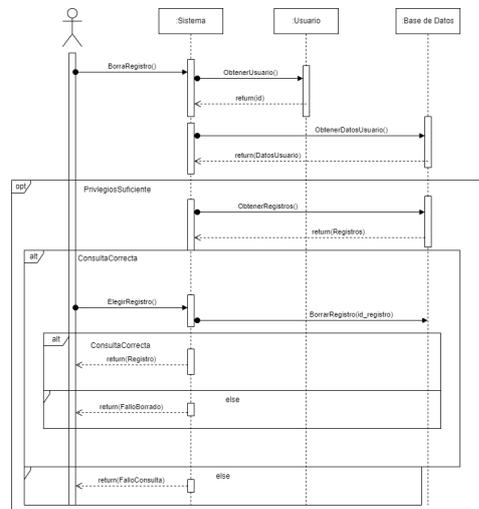


Figura 3.34: Diagrama Interacción CU33: 'Eliminar registro de asistencia'

## Capítulo 4

# Implementación

Vamos a desarrollar ahora como se ha implementado la plataforma web durante el proyecto. Lo primero que se hizo fue considerar todos los lenguajes existentes para decidir cual sería el más adecuado para el proyecto. En nuestro caso elegimos Javascript [8]. Las razones principales para su elección son principalmente la cantidad de imágenes de la tecnología de Docker existente con el lenguaje, la gran variedad de librerías que tiene javascript tanto para desarrollar el frontend como el backend de nuestra aplicación. Además, el lenguaje no tiene una curva de aprendizaje excesivamente alta, con lo que no nos retrasará su aprendizaje a la hora de iniciar el desarrollo. Por último mencionar que vino recomendado el uso de NodeJS [9] para la parte del backend.

### 4.1. Docker

Vamos a comentar primero la tecnología Docker que hemos empleado en nuestro programa. Para ello comenzaremos con nuestro archivo 'docker-compose.yml' Lo primero que aparece en él son las variables necesarias para conectarse al contenedor de MYSQL, que será nuestro sistema de gestión de base de datos.

```
version: "3.4"

x-common-variables: &common-variables
  MYSQL_USER: tfguser
  MYSQL_PASSWORD: tfgpasswd
  MYSQL_DATABASE: sampledb
  REACT_APP_SERVER_PORT: 3001
```

Figura 4.1: Variables de acceso a la base de datos

Ahora mostraremos todos y cada uno de los servicios que componen el archivo y su función **Servicio de MYSQL** Este servicio nos proporciona un

sistema gestor de base de datos para bases de datos relacionales. A través de él realizaremos todas las peticiones necesarias a la base de datos para obtener la información dentro de ella o actualizarla cuando sea necesario.

```
mysql-db:
  image: mysql:5.7
  container_name: mysql_container
  environment:
    <<: "common-variables"
    MYSQL_ROOT_PASSWORD: tfgpasswd
  ports:
    - "3307:3306" # Mapeo correcto: el puerto externo 3307 se mapea al interno 3306 de MySQL
  restart: unless-stopped
  volumes:
    - ./db/sample.sql:/docker-entrypoint-initdb.d/sample.sql
```

Figura 4.2: Variables de acceso a la base de datos

**Servicio de PHPMyAdmin** PHPMyAdmin es una herramienta para facilitar la gestión de la base de datos durante el desarrollo. Proporciona una interfaz fácil de utilizar para no estar forzados a usar la consola de SQL que puede llegar a ser tediosa.

```
phpmyadmin:
  depends_on:
    - mysql-db
  image: phpmyadmin/phpmyadmin
  container_name: phpadmin_container
  environment:
    PMA_HOST: mysql-db # Host apunta al nombre del servicio de MySQL
  links:
    - mysql-db:mysql-db
  ports:
    - "8080:80"
  restart: always
```

Figura 4.3: Variables de acceso a la base de datos

**Servicio de servidor** Este servicio es el que abre el puerto para ejecutar el cliente de la aplicación. Es él que se comunicará con la base de datos y realizará las peticiones para manejar la información, además de ejecutar toda la lógica de la aplicación.

```
server:
  build: ./server
  container_name: node_server_container
  depends_on:
    - mysql-db
  environment:
    <<: "common-variables"
    MYSQL_HOST_IP: mysql-db # Debe ser el nombre del servicio para que se resuelva en Docker
  ports:
    - "3001:3001" # Mapeo del puerto externo 3001 al interno 3001
  volumes:
    - ./server/app
  links:
    - mysql-db
  command: node index.js # Cambiado a "node index.js" si el servidor se inicia con este comando
```

Figura 4.4: Variables de acceso a la base de datos

### Servicio de cliente

Servicio muy parecido al del servidor, habilita el puerto que va a ser usado por el cliente y ejecuta el código del frontend para estar listo cuando le haga falta al usuario

```
client:
  build: ./client
  container_name: client_container
  environment:
    <<: *common-variables
    NODE_PATH: src
  ports:
    - "3000:3000" # Mapeo del puerto externo 3000 al interno 3000
  volumes:
    - ./client/src:/app/src
  links:
    - server
  command: npm start # Comando para iniciar la aplicación React
```

Figura 4.5: Variables de acceso a la base de datos

Para explicar el orden de ejecución del docker vamos a adjuntar un diagrama. Esta pensado para que los servicios se vayan ejecutando en orden secuencial, forzando la espera para que no haya fallos por falta de ejecución de algún servicio. Los dos ultimos servicios llaman a otros dos archivos simples, los Dockefile, que se encargan de la gestión de los paquetes de NodeJS y javascript.

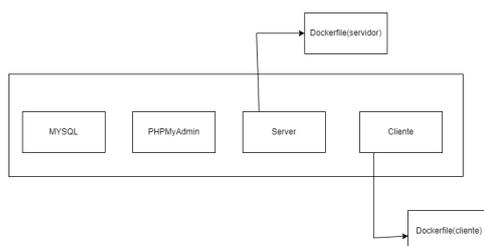


Figura 4.6: Diagrama de llamadas en el proceso de ejecución de Docker

## 4.2. Frontend

Ahora vamos a comentar como se ha implementado cada una de las partes del proyecto y las alternativas que se han tenido en cuenta para su desarrollo.

### 4.2.1. Alternativas

Hay otras múltiples opciones para poder realizar la parte visual de nuestra aplicación, el lenguaje de programación elegido para la aplicación nos brinda múltiples opciones que son muy usadas actualmente. Exploramos también qué opciones nos ofrecen otros lenguajes de programación

Dentro de JavaScript

**Angular:** Angular actualmente es una de las librerías más usadas para programar frontend de la actualidad. Permite desarrollar tanto backend y frontend con una estructura sólida, lo que la hace perfecta para aplicaciones

grandes que buscan uniformidad y un desarrollo parecido en toda la aplicación. En contraposición a esto nos damos cuenta de que es menos flexible que otras tecnologías y que tienen una curva de aprendizaje más difícil de tratar debido a tener que aprender tanto backend como frontend. En el caso concreto de React-Native-Native, ya sea native o no, podemos decir que React-Native-Native es más rápido al tener menos código para su estructura aunque Angular te permite un control más fino de los estados y servicios de tu aplicación.

**Vue.js:** Esta herramienta es muy recomendada entre principiantes en el desarrollo de aplicaciones, debido a su fácil configuración y a lo que contiene integrado el software, como puede ser el router, para las redirecciones y Vuex, para manejar los estados dentro de la aplicación. No es recomendable para proyectos grandes debido al ecosistema limitado, que se define por las bibliotecas y plugins compatibles y que permiten trabajar con la herramienta, de este en comparación a las otras alternativas. SI lo comparamos con la que hemos elegido podemos destacar que Vue es más sencillo de usar y con un aprendizaje más progresivo pero puede llegar a flaquear en proyectos más grandes debido a su falta de ecosistema, en este aspecto React-Native-Native ofrece mejores prestaciones tanto en el desarrollo de la aplicación como en el rendimiento de la misma una vez se ha terminado.

**Gatsby:** Es un framework basado en React-Native-Native que se ha especializado en el desarrollo del contenido estático de las páginas web, siendo este su mayor punto fuerte por la automatización que ofrece el framework, la precarga de rutas para estos contenidos, haciendo la navegación más rápida una vez se han cargado además de cargar únicamente el contenido de la hoja de estilos css que sea necesario para poder ver la parte de la página web que está viendo el usuario. Por último mencionar el extra de seguridad que aporta el centrarse en contenido estático, no crea dependencias con servidores externos o herramientas extra para realizar backed o ataques como la inyección de SQL no serían útiles.

**Lit:** Es la herramienta de desarrollo más parecida a React-Native-Native, ya que consiste en el diseño de componentes para las páginas web que luego el navegador interpreta y añade el código necesario para su visualización. Las ventajas de Lit consisten en ser muy ligero, con lo que no añade excesiva carga a la aplicación web a la hora de ejecutarla y cargar los componentes que se han creado con esta herramienta. Otra ventaja es la sintaxis de la herramienta, moderna y adaptada ayuda mucho al desarrollo de los componentes web. Las desventajas se concentran en su falta de comunidad y en su falta de opciones para crear aplicaciones grandes. Al estar muy centrado en componentes pequeños no está muy desarrollado para componentes más grandes o grandes conjuntos de componentes. Su falta de comunidad puede llegar a ser un problema para futura escalabilidad o para encontrar

soluciones a errores.

**Alpine.js:** Alpine consiste en otro framework de javascript enfocado en añadir comportamiento directamente sobre tu lenguaje de marcado, como puede ser HTML. Es muy ligero y simple de utilizar, sus 2 principales ventajas para elegirlo. Además, tiene un lenguaje muy minimalista y muy directo para añadir la interacción necesaria. Es una muy buena herramienta para manejar estados, como herramienta complementaria, no para definirlos. Los puntos donde flaquea Alpine es en su falta de gestión de enrutamiento o de datos, donde otras herramientas destacan más. No facilita la creación de componentes más avanzados, con lo que para hacer aplicaciones más grandes no es adecuado. Al ser lo más minimalista posible carece de herramientas integradas para hacer test de manera eficiente o no optimiza el rendimiento de los componentes creados, a más complejos su carga de ejecución será mayor que en otros framework.

Fuera de JavaScript

**Python:** Python es conocido por las múltiples aplicaciones que se han ido desarrollando por parte de la comunidad en forma de ya sean en forma de librerías o de frameworks. En cuanto a framework, lo que estamos tratando los más destacados son Flask y Django, dos herramientas muy usadas actualmente y muy potentes para el desarrollo de aplicaciones fullstack. Las principales ventajas de estas herramientas consisten en la integración de herramientas como el ORM, herramienta que nos permite trabajar con las bases de datos usando el enfoque de la programación orientada a objetos que representan las tablas y las relaciones entre objetos. Al ser herramientas fullstack no nos obligan a añadir otras herramientas extra para el desarrollo. Los puntos negativos se centran en lo pesado que puede llegar a convertirse la herramienta a la hora de aplicación y en la falta de opciones para hacer componentes con alto nivel de complejidad, puede llegar a convertirse en una ejecución muy aparatosa. Si lo particularizar para nuestra herramienta, vemos que se ajusta a lo comentado anteriormente

**Java:** Java es un lenguaje que se ha utilizado durante mucho tiempo para el desarrollo tanto de aplicaciones web como de programas de escritorio. Las herramientas que vamos a comparar con la nuestra son Spring y JSP, ambas 2 usadas para el desarrollo de aplicaciones. Los puntos fuertes de esta herramienta son que el lenguaje Java es muy robusto y es muy bueno para implementar la lógica de la aplicación y de los posibles componentes que se creen. Algunas de estas herramientas, Spring, contienen ya implementadas herramientas para implementar las funciones del backend de manera cómoda, en cambio React-Native-Native no. Flaquean las herramientas en cuanto a la interactividad de las aplicaciones que se crean, React-Native-Native es más apto y dedicado para eso. Por último mencionar que con React-Native-Native es más fácil desarrollar el SPA, un tipo de aplicación que consta de

una sola página HTML, cargando todos los elementos de una vez, lo que nos permite dar una navegación más rápida ya solo necesitamos recargar partes de la interfaz en vez de archivos enteros. Añadiremos una tabla de resumen

Tecnología	Ventajas	Desventajas	Ideal para	Comunidad y Ecosistema	Escalabilidad
Angular	Estructura sólida, permite desarrollo fullstack, control preciso de estados y servicios	Menos flexible, curva de aprendizaje difícil	Aplicaciones grandes y uniformes	Amplia y activa	Alta
Vue.js	Fácil configuración, aprendizaje progresivo, router y Vuex integrados	Ecosistema limitado, no ideal para proyectos grandes	Proyectos medianos y pequeños	Moderada a amplia	Media, limitada en grandes proyectos
Gatsby	Especializado en contenido estático, precarga de rutas, mayor seguridad	Limitado a contenido estático, dependencia de React-Native-Native	Páginas web con contenido estático	Activa y creciente	Alta para contenido estático
Lit	Ligero, sintaxis moderna, fácil de usar para componentes pequeños	Falta de comunidad, no adecuado para proyectos grandes	Componentes pequeños y proyectos ligeros	Pequeña	Baja en proyectos grandes
Alpine.js	Minimalista, fácil de añadir interacción a HTML, ligero	Falta de gestión de enrutamiento y datos, no adecuado para proyectos grandes, sin herramientas de test ni optimización de componentes	Complemento para añadir interacción	Pequeña	Baja
Python	Amplio ecosistema de frameworks (Flask, Django), integración con ORM	Menos orientado al desarrollo frontend puro	Aplicaciones fullstack y prototipos rápidos	Muy amplia	Alta en aplicaciones completas

Cuadro 4.1: Comparación de tecnologías

El frontend es una aplicación de React. Esta aplicación consiste en llamadas a los endpoints que están definidos en una API de NodeJS que se encuentra en el backend. Estas llamadas mandan la información necesaria a la API para realizar las operaciones correctamente y estos métodos mandan los resultados, ya sea un error o de éxito al cliente, para informar al usuario de los resultados de las operaciones.

React es una librería de javascript que fue desarrollada por Meta dedicada para crear interfaces de usuario. Es una librería, la cual elegimos por su facilidad para incluirse con otras librerías y herramientas, que se centra en la vista de la aplicación. Su arquitectura y funcionamiento se basa en componentes, los cuales comentaremos los más importantes luego, unidades de código que son capaces de reutilizarse en varios puntos de la aplicación. Permite modificar el renderizado de los componentes en función de los datos que se reciben. Tiene un enfoque unidireccional que se aplica los datos, mejorando el control sobre este y permitiendo conocer el resultado de las operaciones y controlar dichos resultados. Por último mencionar la comunidad tan extendida que tiene y la alta popularidad que posee entre la comunidad de desarrollo web.

Comentaremos ahora algunos de los componentes más significativos, varios de ellos son utilizados en múltiples partes de la aplicación.

**Manejo Usuarios:** Este componente muestra una lista con todos los usuarios que están registrados en el sistema. Cada registro se convertirá en un botón el cual al pulsarlo abrirá otro componente para modificar la información del usuario.

**Cambio Dispositivos:** Componente formulario para modificar los datos del objeto seleccionado anteriormente, utilizado con varias entidades.

**perfil:** Componente para que el usuario cambie sus datos a su gusto cuando le haga falta. Por defecto aparece los que ya tienen registrados.

**registro asistencia:** Componente para generar el registro de asistencia del usuario en el dispositivo conectado.

**Register:** Componente para registrar a un nuevo usuario en la plataforma. Formulario con todos los posibles campos a rellenar, obligatorios son usuario y contraseña.

**Login:** Componente para iniciar sesión en la plataforma web, formulario que pide usuario y contraseña.

**Main Page:** Componente principal de la aplicación web. Contiene todos los botones que dan acceso a las funcionalidades del sistema.

**Poner Reglas:** Componente que muestra la información que pasa a través de otros 2 componentes. Lo muestra en una frase la cual resalta la

información proveniente de los otros componentes

**Buscador:** Buscador para filtrar los registros de asistencia ya sea por dispositivo o el usuario que lo ha generado.

**Subir Datos:** Componente que pide un archivo para subir los datos de este a la base de datos. En caso de error avisa al usuario

**Descargar Datos:** Componente que pide confirmación para descargar los datos de asistencia de la base de datos en formato CSV.

## 4.3. Backend

### 4.3.1. Alternativas

Para el backend vamos a elegir la herramienta más usada, o conjunto de estas herramientas, para los lenguajes más usados para el desarrollo de aplicaciones, ya sean específicas de backend o fullstack. Las compararemos con la que nosotros hemos elegido, NodeJs junto a MySQL.

**Django/PostgreSQL:** Django es una herramienta que usa el lenguaje python combinada con postgres para una gestión más visual y accesible de la base de datos. Destacan mucho su fácil configuración y las convenciones que ya existen para hacerla más amena. Combinadas con las herramientas que trae de por sí ayuda mucho a la producción de la aplicación, para que sea alta. Otro aspecto a destacar es su ORM, de los más intuitivos y fáciles de usar. Por último, mencionar que tanto el lenguaje como la herramienta contienen una comunidad muy extendida, con lo que abundan los recursos por la red. El mayor factor que tiene en su contra es la velocidad en cuanto a la E/S de datos, ya que es más lenta que otras herramientas de interfaces que contrastaremos posteriormente. Además, para novatos en el lenguaje de Python puede llegar a costar comprender y llegar a usar correctamente la herramienta. Por último, mencionar que no está enfocado a microservicios, no es excesivamente escalable.

**Spring Boot/MySQL:** Esta herramienta está desarrollada para Java, lenguaje del cual ya hemos comentado sus cualidades anteriormente. Esta herramienta tiene múltiples ventajas como por ejemplo el multithreading, que favorece la creación y el manejo de hebras que ayudan a la gestión del paralelismo para la lógica de la aplicación, proveniente en gran parte de Java. Las APIs creadas con esta herramienta son muy robustas, son escalables, seguras. Recomendable seguir la arquitectura REST, que es un estilo que define restricciones y prácticas para crear APIs que sean escalables y fáciles de usar. Contiene múltiples herramientas que ayudan a aspectos como pueden ser las pruebas, facilitado por Spring Test, o la creación de microservicios, Spring Cloud es muy útil para esto. Las desventajas que tiene esta

herramienta se centran mucho en su complejidad principalmente. La curva de aprendizaje es muy pronunciada tanto para gente que conoce Java como nuevos en el lenguaje, lo cual no es beneficioso para desarrollos de aplicaciones más pequeñas. En comparación a Nodejs para hacer tareas similares se requiere más código, con lo que puede llegar a ser engorroso y puede llegar a dificultar la gestión y el mantenimiento de la aplicación. Estos factores pueden suponer una ralentización en el desarrollo del proyecto.

**RoR/PostgreSQL:** Ruby on Rails es una tecnología conocida por su versatilidad y por poderse conectar con otras tecnologías que ejerzan de frontend, el lenguaje de ruby no es muy usado últimamente. Esta herramienta tiene buen soporte para varias arquitecturas de APIs, como puede ser la REST o la GraphQL, lenguaje que da más control a los clientes de los datos que piden en las peticiones al servidor y busca solventar las limitaciones del estilo REST. Esta herramienta tiene un tiempo de configuración corto con lo que si tenemos las gemas de ruby preparadas nos será muy simple. Tiene una gran comunidad detrás, además de un ecosistema muy potente que le da acceso y compatibilidad con varias librerías y plugins. Los puntos en contra de RoR son por ejemplo la necesidad de añadir gemas para tareas que otras herramientas ya traen con la instalación, con lo que te obliga a buscar cómo añadirlas y buscar una versión compatible con las que ya tienes instaladas. No es tan escalable a pesar de ser compatible con estilos arquitectónicos y lenguajes que sí lo son, con lo que para aplicaciones grandes se puede llegar a quedar corto. Si lo comparamos con Nodejs se nota que a medida que sube la complejidad de la aplicación se reduce el rendimiento de la API de RoR, cosa que con Node no pasa.

**Flask/SQLite:** Es la herramienta más simple que vamos a tratar, está desarrollada en Python, ya mencionado múltiples veces anteriormente, y destaca principalmente en proyectos de pequeño alcance y poca complejidad. Su mayor ventaja es la velocidad de desarrollo que proporciona la herramienta, por la curva de aprendizaje que es poco pronunciada como en su rapidez a la hora de configurar y preparar el entorno a la hora de empezar el proyecto. Todo esto la convierte en una herramienta muy conveniente para este tipo de proyectos pequeños que no tienen ni excesiva complejidad en cuanto a lógica ni alcance. También destacar la simpleza de la herramienta, aunque esto lleva a uno de sus mayores debilidades, al ser tan simple en cuanto a funcionalidad es lo único que trae, además de que trae la justa y necesaria. Este hecho obliga a tener que buscar herramientas externas para cualquier cosa que esté fuera de la funcionalidad de Flask. Si lo comparamos con la herramienta elegida podemos ver que Flask está más dedicada a proyectos más pequeños y rápidos, mientras que Nodejs tiene la capacidad de poder ser más productiva y útil el proyectos más grandes, además de que incluye más cosas a parte de la lógica, como puede ser el gestor de paquetes NPM, Node Packet Manager, y soporta formatos de archivos como puede ser el

JSON. Añadiremos otra tabla de resumen

Tecnología	Ventajas	Desventajas	Ideal para	Comunidad y Ecosistema	Escalabilidad
Django/PostgreSQL	Fácil configuración, muchas herramientas integradas, ORM intuitivo, gran comunidad	Menor velocidad de E/S de datos, curva de aprendizaje para novatos, no optimizado para microservicios	Proyectos completos de tamaño mediano a grande	Amplia y activa	Media, limitada para microservicios
SpringBoot/MySQL	Multihebra, APIs robustas y seguras, escalabilidad, soporte para microservicios con Spring Cloud	Complejidad alta, curva de aprendizaje pronunciada, requiere más código en comparación a Node.js	Aplicaciones empresariales y escalables	Amplia y profesional	Alta
RoR/PostgreSQL	Versátil, fácil configuración, soporte para REST y GraphQL, gran comunidad y ecosistema	Necesidad de gemas externas, menor rendimiento en aplicaciones grandes	Proyectos de tamaño pequeño a mediano	Activa y con buen soporte	Media, limitada para grandes proyectos
Flask/SQLite	Rápida configuración, curva de aprendizaje baja, ideal para proyectos pequeños	Funcionalidad limitada, dependencia de herramientas externas para funcionalidades adicionales	Prototipos y aplicaciones pequeñas	Amplia (Python)	Baja para proyectos complejos

Cuadro 4.2: Comparación de tecnologías

Esta parte tiene como bibliotecas principales NodeJS, Mysql2 [13] y Express [6]. Node sera nuestro entorno de ejecución por parte del servidor. Express es un framework ligero que ayuda a crear APIS para plataformas web. Mysql2 es un cliente ligero que hará las consultas a la base de datos, ligera y rápida que además hace lo más eficientes posibles las consultas a esta.

Node es muy ligero e instala unicamente las herramientas necesarias para gestionar rutas y solicitudes HTTP, haciendo así que sea ligero ya a la vez completo. Proporciona un control muy directo sobre el servidor

El empleo de Express simplifica la creación de rutas y el manejo de middleware para gestionar las solicitudes, mientras que Node.js, gracias a su naturaleza asíncrona, posibilita el manejo de varias solicitudes al mismo tiempo sin bloqueos, lo que optimiza el rendimiento. Asimismo, se emplea mysql2 como cliente de MySQL para comunicarse con la base de datos. mysql2 es una biblioteca ágil y ligera que facilita la ejecución de consultas SQL de forma eficiente, respaldando promesas y conexiones seguras con la base de datos. Esto facilita la administración de las operaciones de bases de datos, como la inserción, actualización y consulta de datos, al integrarse de manera ideal con el flujo asíncrono de Node.js.

La principal razón para optar por Express y mysql2 es su facilidad de uso, el control exhaustivo sobre la lógica del servidor y el conocimiento previo de JavaScript, lo que permite utilizar un único lenguaje en el frontend y el backend. Asimismo, la comunidad dinámica de Node.js y su extenso ecosistema de módulos ofrecen recursos y herramientas suplementarias que amplían las funcionalidades de la aplicación.

Vamos a comentar ahora algunos endpoints más significativos, el resto o son consultas muy simples o estos representan un endpoint genérico que ya trataremos.

**Login** Endpoint para comprobar que los datos que el usuario proporciona en el registro son correctos, en caso afirmativo avisa a la plataforma para dar acceso a la página principal

**Registro** Endpoint para registrar a un nuevo usuario en la plataforma. Obtiene los datos del componente correspondiente.

**Borrar Objeto** Endpoint genérico para borrar objetos correspondientes a nuestras entidades, recibe en la url la tabla y un ID

**Modificar Objeto** Endpoint genérico para modificar objetos de las tablas. Recibe por la url la tabla y el id, los datos a modificar por el body.

**Mandar Mensaje** Endpoint para registrar mensajes entre usuarios.

**Generar Reporte** Endpoint para generar un reporte de asistencia. Pri-

mero obtiene los datos a través de consultas a la base de datos y luego llama al módulo de generación de pdfs

**Generar Informe** Muy parecido al anterior, cambiando los datos que se van a incluir en el pdf.

**Cargar Datos** Endpoint que usa el módulo de archivos csv para abrir el archivo y posteriormente pasar la información a la base de datos. Comprueba primero el formato del archivo

**Descargar Datos** Endpoint que crea un archivo en formato csv con los datos de asistencia registrados en la plataforma. Usa el módulo antes mencionado

**Copia de Seguridad** Endpoint para generar un archivo que sirva para restaurar la información de la base de datos con los del archivo.

También comentaremos los módulos que utilizaremos para crear los archivos que puede obtener el usuario.

**Archivos PDF** Este módulo consiste en una vez recibida la información la irá colocando en un archivo pdf. Dicha información proviene de consultas a la base de datos. Se utiliza la librería `pdftoolkit`, [12], el cual coloca la información comodamente con funciones que proporciona la librería

**Archivos CSV** Este módulo es parecido al anterior, se utiliza la librería `xlsx` [10] para crear los archivos y para leerlos cuando hace falta. A la hora de llamarlo para abrir los archivos se llama en una llamada recursiva hasta terminar las filas llamadas, tras verificar que la información del archivo es correcta se obtiene la información y se realiza una llamada al cliente de `mysql2`.

## Capítulo 5

# Presupuesto

Para estimar cuanto costaría producir este software vamos a intentar llevarlo a un escenario más profesional. Vamos a empezar por enunciar los recursos que se precisan para completar el desarrollo del programa y cuanto costaría en la vida real.

Comenzaremos con los recursos materiales:

- **Hardware:** Debemos equipar a todos los trabajadores del proyecto con equipos adecuados para el desarrollo del programa. Para ellos debemos comprar por trabajador un ordenador con un microprocesador mínimo un i5 de 10<sup>a</sup> generación o a poder ser más. De memoria RAM deberían ser suficientes 6 GB.
- **Infraestructura y Servidores:** El laboratorio proporciona su propio servidor para ejecutar el servicio, además de todos los certificados necesarios para poder ejecutarlos online de manera segura, con lo que este gasto lo consideraremos ínfimo.

En cuanto a herramientas de desarrollo, todas las librerías utilizadas en el desarrollo, iconos de los botones, etc., son de software libre, con lo que no supone coste. En cuanto a las herramientas de desarrollo, se daría libertad de elección, pero para nuestro caso con Visual Studio Code y una configuración correcta será más que adecuado.

Discutamos ahora los recursos humanos que serán usados:

- **Equipo de desarrollo de Backend:** Serán los encargados de programar la lógica de la aplicación y todo lo relacionado con la base de datos. También se asegurarán de crear y poner en marcha los mecanismos necesarios para proteger la integridad de la base de datos y los datos que se guardarán en ella.
- **Equipo de desarrollo de Frontend:** Crearán una interfaz *user friendly* para que sea lo más amena y fácil de utilizar, uno de los pro-

blemas que presentaba el programa con el que cuenta el laboratorio. Seguirán los diseños creados por el diseñador de interfaces.

- **Diseñador de UX/UI:** Un buen diseño de la interfaz nos agilizará el desarrollo y ayudará a que la aplicación llegue a buen puerto. Buscaremos que sea lo más accesible y cómoda de usar para todo tipo de usuario.
- **Project Manager:** Un buen project manager conseguirá que el desarrollo sea efectivo y eficaz. Coordinará a todos los roles mencionados anteriormente y se encargará de comunicarse con los interesados en el programa, los stakeholders. Ayudará al equipo en todo lo que sea necesario.

En este contexto, el grupo de testers van a ser las personas que acudan al laboratorio a realizar sus proyectos, con lo que no nos supondrá coste adicional. Comunicarán posibles mejoras a los encargados de mantener el programa.

Con toda esta información obtenemos la tabla de gasto de los recursos humanos resultante en función a la información obtenida de [14] y [11]

Estimando que el proyecto requerirá unas 240 horas de programación de backend y de frontend, 100 horas de diseño de interfaz y cerca de 160 horas que empleará el director de proyecto para gestionar todo queda. Multiplicaremos estas horas por los precios obtenidos de las referencias.

Rol	Coste total de cada rol (\$)
Equipo de desarrollo de Backend	[14,400]
Equipo de desarrollo de Frontend	[14,400]
Diseñador de UX/UI	[5,000]
Project Manager	[11,200]
Total	[45,000]

Cuadro 5.1: Presupuesto estimado para el desarrollo del software

# Capítulo 6

## Manual de usuario

Vamos a explicar ahora como vamos podemos instalar y utilizar la funcionalidad básica del producto final de este proyecto. Proporcionaremos una guía clara y concisa para que los usuarios puedan desplegar y utilizar la plataforma web de manera clara. Vamos a exponer la facilidad con la que se acceden a las funcionalidades del programa, uno de los objetivos de este proyecto.

### 6.1. Instalación

Para la instalación debemos mencionar que se requiere la instalación previa de la tecnología de Docker, para poder acceder a su variedad de comandos que ofrece. Para lanzar la aplicación unicamente debemos usar el comando

**docker compose up**

Una vez lo hemos lanzado abriremos en nuestro navegador 'http://localhost:8080' que nos llevará a una página web de soporte para la gestión de la base de datos

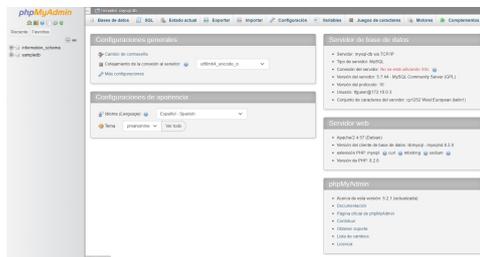


Figura 6.1: Captura de PHPMyadmin

Aqui clicaremos sobre la base de datos 'samplebd'. Tras esto en importar y seleccionaremos el archivo sql proporcionado en el proyecto.

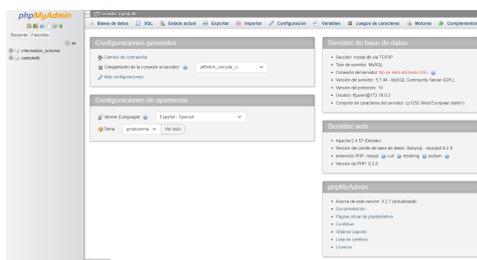


Figura 6.2: Captura de PHPMyadmin

Por ultimo solo debemos abrir en nuestro navegador 'http://localhost:3000' para usar nuestra aplicación.

## 6.2. Manual de usuario

Vamos a incluir ahora una serie de capturas de pantalla de la aplicación para hacer un recorrido por las funciones más significativas de la plataforma.

Todo empieza con el inicio de sesión, tras haber realizado el registro previo

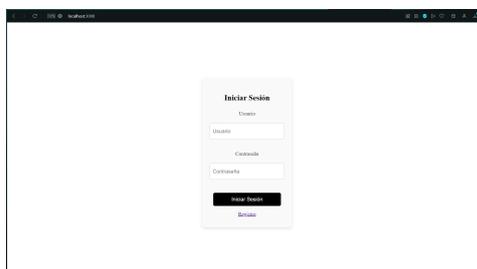


Figura 6.3: Inicio de sesión de la aplicación

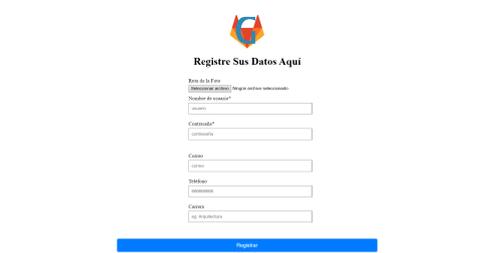


Figura 6.4: Registro en la aplicación

Una vez se haya completado el registro o el login se accede a la página principal

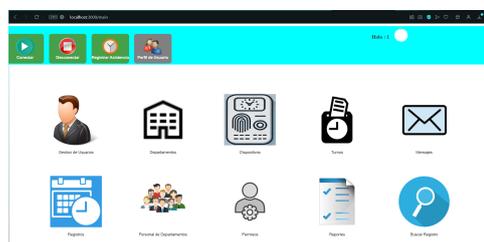


Figura 6.5: Página Principal 1

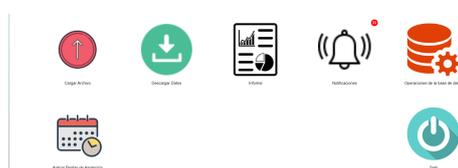


Figura 6.6: Página Principal 2

Desde aquí podemos acceder a todas las funcionalidades. La vista que se ve es la vista de administrador, en caso de tener menos privilegios aparecerán menos funcionalidades disponibles

Aquí podremos ver los datos del propio usuario y podría poder modificarlo

Figura 6.7: Página cambiar objetos

Vamos a ver ahora como podemos registrar un nuevo objeto, en este caso un dispositivo

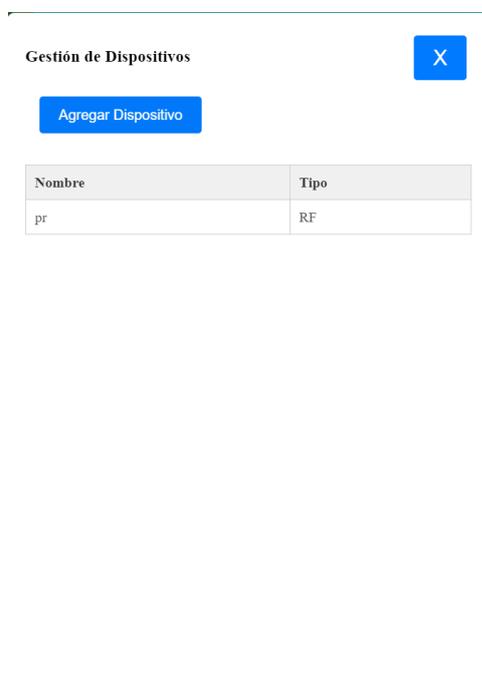
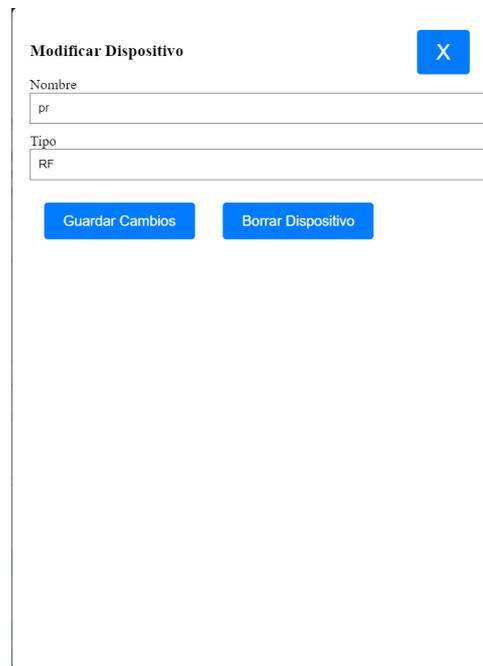


Figura 6.8: Página de lista de objetos

Al pulsar en uno de los botones de la tabla se abre otra pestaña para modificar el objeto o borrarlo



The image shows a web form titled "Modificar Dispositivo" enclosed in a modal window. The form has two input fields: "Nombre" with the value "pr" and "Tipo" with the value "RF". Below the fields are two blue buttons: "Guardar Cambios" and "Borrar Dispositivo". A blue button with a white "X" icon is located in the top right corner of the modal.

Figura 6.9: Página cambiar objetos

Ahora vamos a mostrar como podemos cambiar los privilegios de los usuarios

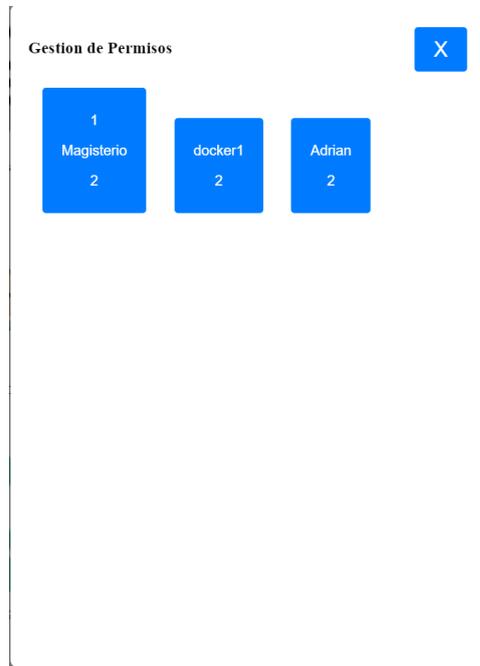


Figura 6.10: Página lista para cambiar permisos

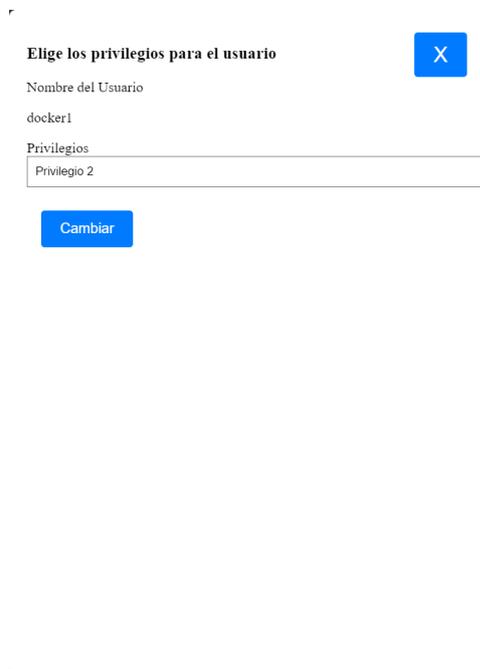
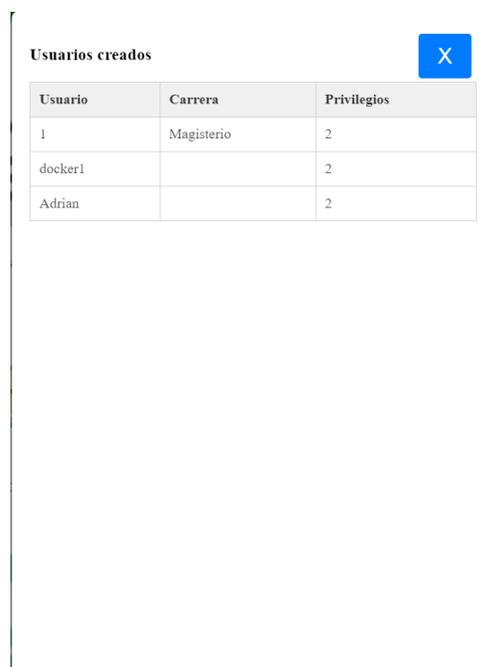


Figura 6.11: Página cambiar privilegios

Veremos como podemos añadir o eliminar un usuario de un departamento



Usuarios creados X

Usuario	Carrera	Privilegios
1	Magisterio	2
docker1		2
Adrian		2

Figura 6.12: Página lista de usuarios

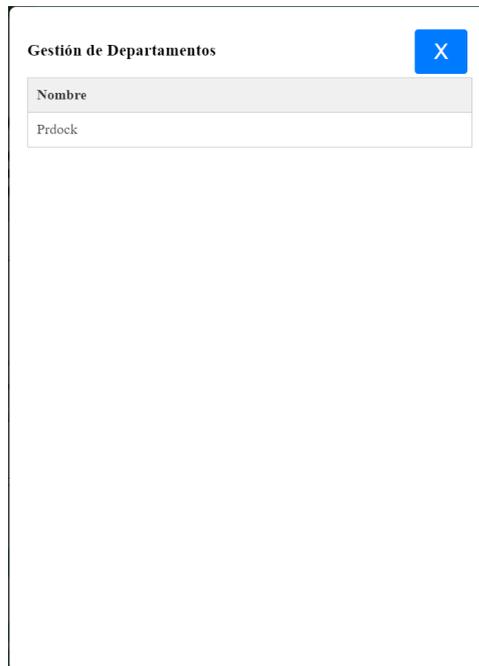


Figura 6.13: Página lista de departamentos

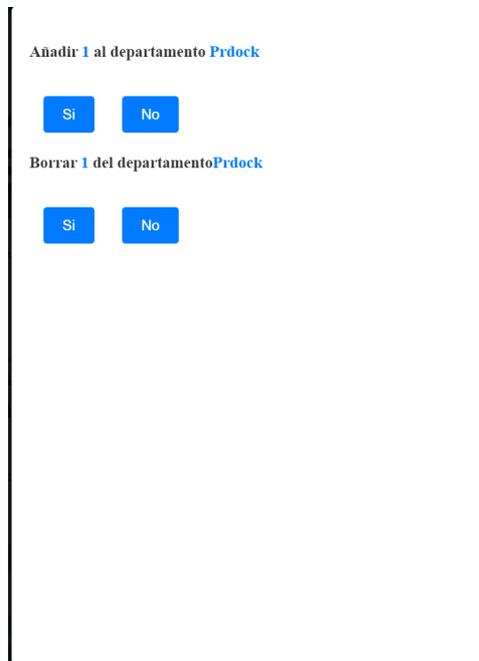


Figura 6.14: Página confirmar operación

Vamos a ver ahora como podemos generar un reporte de asistencia

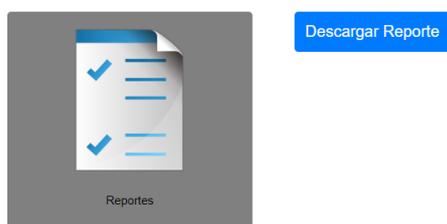


Figura 6.15: Página generar reporte

Veremos ahora como utilizar el buscador, que debemos escribir en el input



Figura 6.16: Página uso buscador

Vamos ahora a ver como podemos descargar los registros de asistencia

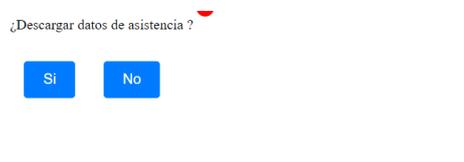
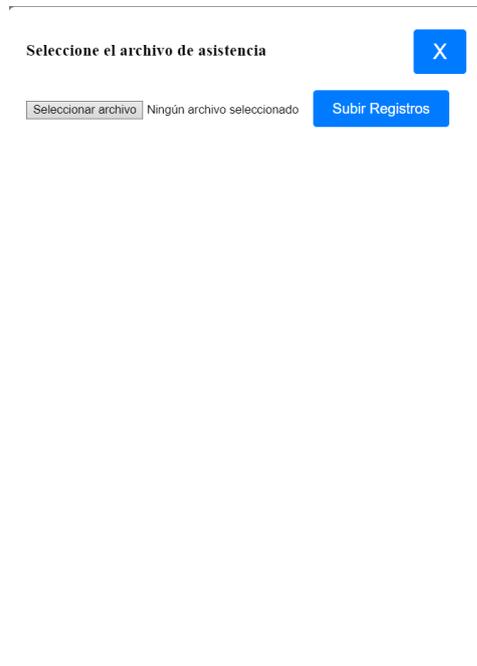


Figura 6.17: Página descargar datos

Veremos ahora como cargar los datos de asistencia

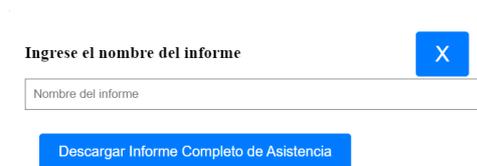


Seleccione el archivo de asistencia

Seleccionar archivo Ningún archivo seleccionado Subir Registros

Figura 6.18: Página subir registros

Veremos ahora como generar informe de asistencia, hay que darle un nombre



Ingrese el nombre del informe

Nombre del informe

Descargar Informe Completo de Asistencia

Figura 6.19: Página generar informe

Mostraremos ahora donde se ven las notificaciones

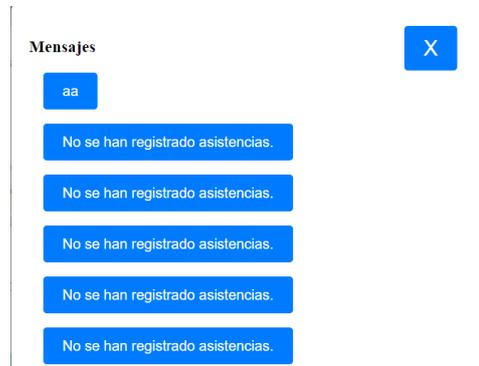


Figura 6.20: Página notificaciones

Mostraremos ahora como abrir y ejecutar las operaciones del sistema

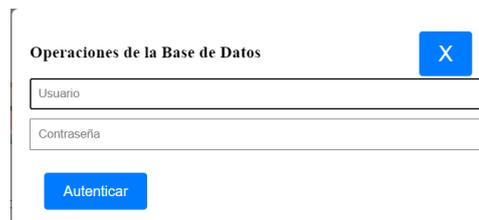


Figura 6.21: Página operaciones del sistema



Figura 6.22: Página operaciones del sistema 2



## Capítulo 7

# Conclusiones

El objetivo que perseguía este proyecto es proporcionar una herramienta que permitiera actualizar una aplicación que ya estaban usando. Las actualizaciones se han centrado mayormente en la parte del frontend, ya que en cuanto a lógica no se ha llegado a variar nada. Esta actualización del frontend permitirá que sea una aplicación más fácil de usar y más intuitiva para los usuarios. Otro aspecto a destacar es la facilidad que se le ha aportado a la aplicación a través del uso de Docker, permitiendo que la instalación del programa sea simple.

Para la mejora del frontend se ha utilizado herramientas más actualizadas para que el diseño sea más atractivo para el usuario, el actual puede dar la sensación de antiguo. Este diseño busca también facilitar la experiencia de uso al usuario, el diseño está centrado en el usuario final que va a usar nuestra aplicación. Esta interfaz quiere facilitar el proceso de generación y consulta de los registros de asistencia que generan los usuarios y facilitar todo lo posible a los administradores el manejo y la gestión de la base de datos, tanto para la seguridad como la integridad de los datos.

El uso de Docker para el despliegue de la aplicación ha permitido que la aplicación se pueda instalar y usar en múltiples plataformas, estas han de ser compatibles con Docker. Así solucionamos uno de los problemas que presenta el programa actual, solo está instalado en un solo dispositivo. Si a esto le añadimos un manual de instalación claro y conciso, cumpliremos con la expectativa.

El proyecto podemos confirmar que ha llegado a buen puerto y ha cumplido con los objetivos que tenía previstos. Hemos conseguido dar una plataforma funcional que sea userfriendly y facil de instalar. Esto ayudará a los futuros estudiantes que realicen sus proyectos en sus instalaciones.

## 7.1. Futuras mejoras

Nuestra aplicación ha completado los objetivos que se le han propuesto, pero esto deja la puerta abierta a posibles mejoras que harían la aplicación mejor y más consistente. Algunas de esas posibilidades se explicarán a continuación.

- **Conectar las bases de datos:** Actualmente la aplicación que actualmente tiene el laboratorio instalada y la que se ha desarrollado en este proyecto están completamente aisladas una de la otra. Con lo que los registros creados en la centralita y los creados desde la aplicación no se escriben en la misma base de datos, con lo que resta realismo e integridad a la base de datos. En caso de querer pasar los datos de una aplicación a otra debemos exportarlos a un archivo en formato csv e importarlos en la otra aplicación, lo cual puede ser aparatoso. Utilizando el SDK correspondiente se conseguiría pasar la información entre aplicaciones, aunque obligaría a tener en cuenta si estamos en la misma red WIFI que la centralita.
- **Creación de gráficos:** Crear gráficos para detectar picos de asistencia en el laboratorio o por ejemplo detectar que método de autenticación se utiliza más. Estos gráficos se pueden llegar a utilizar para planificar visitas al laboratorio o para elegir en caso de mantenimiento o querer mejorar la infraestructura o la plataforma ver donde reportaría mas beneficio centrarse.
- **Añadir más filtros para los archivos generados según los datos de asistencia de los usuarios:** A la hora de generar los archivos se podrían añadir filtros para llegar a ser más específicos con la información que se le proporciona al usuario, para ser más rápidos a la hora de generar el documento y para ser más eficaces a la hora de ayudar al usuario.
- **Añadir factor doble autenticación:** A la hora de entrar en la aplicación se utiliza un inicio de sesión encriptado. Sería interesante reforzar este aspecto añadiendo otro paso de autenticación para hacer la aplicación mas segura y proteger a los usuarios de mejor manera. Con la constante subida de ciberataques, como refleja el ENISA [7], la seguridad que se le pueda aplicar al proyecto siempre es bienvenida.
- **Mejorar la interfaz para personas con discapacidad:** Hay ciertas personas con discapacidad motora las cuales no pueden usar un ratón o un teclado con facilidad. En este caso suelen sustituirlos con pulsadores adaptados para ellos. Esta mejora consistiría en primero conocer si tienen este tipo de limitación y si es el caso facilitarles la experiencia

con algún tipo de método que barra los botones necesarios para que puedan acceder a las funciones de la aplicación de manera más fácil.

## 7.2. Reflexión final

Este proyecto pone de manifiesto la importancia de una buena interfaz en las plataformas web que se van desarrollando en el futuro. Es la primera impresión que tienen los usuarios del producto final, ha de ser fácil de usar y que atraiga al usuario a utilizar la aplicación. Sin una buena interfaz de usuario las probabilidades de fracaso una vez se ha terminado el desarrollo aumentan mucho. Otro aspecto a tener en cuenta es la seguridad, se ha trabajado en este aspecto para cumplir la legislación y para asegurar en la medida de lo posible que la información viaje entre los componentes del programa de manera segura. Por último mencionar la aplicación de los conocimientos adquiridos durante mi estudios de grado en este desarrollo, tanto en el desarrollo del backend para hacerlo lo más eficiente posible como en el del frontend, teniendo en cuenta las necesidades de los usuarios y como crear una interfaz lo más asequible de usar.



## Capítulo 8

# Bibliografía



# Bibliografía

- [1] Cybersecurity and Infrastructure Security Agency (CISA). Cisa annual report on cybersecurity threats and trends 2024, 2024. Accessed: 2024-11-11.
- [2] Docker Inc. Docker, 2024. Último acceso: 10 de noviembre de 2024.
- [3] John Doe and Jane Smith. Implementation of aes encryption for the secure transmission on mysecureweb. *International Journal of Web Security*, 15(3):125–138, June 2024.
- [4] España. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. Boletín Oficial del Estado, 2018. Accedido: 2024-11-11.
- [5] Unión Europea. Reglamento (ue) 2016/679 del parlamento europeo y del consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos. Publicación oficial de la Unión Europea, 2016. Accedido: 2024-11-11.
- [6] ExpressJS. Express - framework web para node.js, 2024. Accedido: 2024-11-11.
- [7] European Union Agency for Cybersecurity (ENISA). Enisa threat landscape 2024, 2024. Accessed: 2024-11-09.
- [8] Mozilla Developer Network. Javascript, 2024. Accedido: 2024-11-11.
- [9] Node.js. Node.js, 2024. Accedido: 2024-11-11.
- [10] Inc. npm. xlsx - biblioteca para manipulación de hojas de cálculo en node.js, 2024. Accedido: 2024-11-11.
- [11] Payscale. Salary data, compensation information and expert network, 2024. Accessed: 2024-11-09.
- [12] PDFKit. Pdfkit - a javascript pdf generation library for node and the browser, 2024. Accedido: 2024-11-11.

- [13] Sidorares. mysql2 - node.js mysql client, 2024. Accedido: 2024-11-11.
- [14] Upwork. Freelance jobs and talent marketplace, 2024. Accessed: 2024-11-09.
- [15] ZKTeco Latinoamérica. K14 - zkteco latinoamérica, 2024. Último acceso: 10 de noviembre de 2024.