



Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Trabajo de
Desarrollo

APLICACIÓN MÓVIL PARA LA GESTIÓN DE ROBOTS CORTACÉSPED

**GRADO EN INGENIERÍA INFORMÁTICA DEL
SOFTWARE**

TRABAJO DE FIN DE GRADO

AUTOR

Álvaro Rodríguez González

TUTOR

Pablo González González

Julio 2024

Este documento ha sido creado basándose en la plantilla elaborada por JOSÉ MANUEL REDONDO LÓPEZ. [1] [2]

Declaración Responsable

El alumno: Álvaro Rodríguez González

Con DNI:

Y UO:

DECLARA

Que esta obra es completamente original y se han citado debidamente las fuentes utilizadas durante la realización de esta.

Y para que conste, lo firma en Oviedo, a 18 de junio de 2024

Firmado: Álvaro Rodríguez González



Agradecimientos

En primer lugar, quiero expresar mi agradecimiento a mi tutor Pablo González González por su seguimiento y apoyo a lo largo del desarrollo del proyecto. Su orientación y la priorización de las funcionalidades han sido esenciales para la finalización de este trabajo.

Estoy profundamente agradecido a toda mi familia, especialmente a mis padres, Juani y Jose, y a mi hermano Miguel, por la confianza y el amor que me han brindado en cada momento de mi carrera, hasta llegar a este punto. Además, ellos me proporcionaron la idea para la aplicación de este proyecto.

Finalmente, quiero dar las gracias a mi novia Sara, por su ánimo y apoyo incondicional durante los últimos meses, lo cual ha sido crucial para completar este proyecto en el presente curso.

En resumen, este proyecto no es solo el resultado del esfuerzo individual, sino también de todas las personas que me han apoyado y confiado en mí a lo largo de todo este tiempo.

Índice de contenido

| | | |
|-------------------|---|-----------|
| Capítulo 1 | ¿Qué es este Trabajo? | 10 |
| 1.1 | Resumen | 10 |
| 1.2 | Palabras clave | 10 |
| 1.3 | Abstract | 11 |
| 1.4 | Keywords | 11 |
| Capítulo 2 | Planificación del Sistema de Información | 13 |
| 2.1 | PSI 1: Inicio del Plan de Sistemas de Información | 14 |
| 2.1.1 | PSI 1.1: Análisis de la Necesidad del PSI | 14 |
| 2.1.2 | PSI 1.2: Identificación del Alcance del PSI | 14 |
| 2.1.3 | PSI 1.3: Determinación de responsables | 15 |
| 2.2 | PSI 2: Definición y Organización del PSI | 16 |
| 2.2.1 | PSI 2.1: Especificación del Ámbito y Alcance | 16 |
| 2.2.2 | PSI 2.2: Organización del PSI | 18 |
| 2.3 | PSI 3: Estudio de la Información Relevante | 19 |
| 2.3.1 | PSI 3.1: Selección y Análisis de Antecedentes | 19 |
| Capítulo 3 | PSI 7: Definición de la Arquitectura Tecnológica | 20 |
| 3.1 | PSI 7.1: Identificación de las Necesidades de Infraestructura Tecnológica | 21 |
| 3.1.1 | Alternativas tecnológicas | 21 |
| 3.1.2 | Comparativa | 27 |
| 3.2 | PSI 7.2: Selección de la Arquitectura Tecnológica | 30 |
| 3.2.1 | Base de datos | 30 |
| 3.2.2 | Servidor | 31 |
| 3.2.3 | Comunicación cliente-servidor | 31 |
| 3.2.4 | Envío de notificaciones | 32 |
| Capítulo 4 | Estudio de Viabilidad del Sistema | 33 |



| | | |
|-------------------|---|-----------|
| 4.1 | EVS 4, 5 y 6: Estudio y Valoración de Alternativas de Solución y Selección de Alternativa Final | 34 |
| 4.1.1 | Sistema 1 – Android y iOS Nativos con Kotlin/Swift y Spring con Java | 34 |
| 4.1.2 | Sistema 2 – Flutter con Dart y Node.js con Typescript | 36 |
| 4.1.3 | Sistema 3 - React Native con JavaScript y Flask con Python. | 37 |
| 4.1.4 | Selección de alternativa final..... | 38 |
| Capítulo 5 | Planificación y Gestión del TFG | 40 |
| 5.1 | Planificación del proyecto | 41 |
| 5.1.1 | Identificación de Interesados | 41 |
| 5.1.2 | OBS y PBS | 41 |
| 5.1.3 | Planificación Inicial. WBS | 43 |
| 5.1.4 | Riesgos..... | 54 |
| 5.1.5 | Presupuesto Inicial..... | 56 |
| 5.2 | Ejecución del Proyecto | 74 |
| 5.2.1 | Bitácora de Incidencias del Proyecto | 74 |
| 5.3 | Cierre del Proyecto | 75 |
| 5.3.1 | Planificación Final | 75 |
| 5.3.2 | Presupuesto Final de Costes..... | 79 |
| Capítulo 6 | Análisis del Sistema de Información..... | 89 |
| 6.1 | ASI 1: Definición del Sistema | 90 |
| 6.1.1 | Determinación del Alcance del Sistema | 90 |
| 6.1.2 | Funcionalidades de alto nivel | 90 |
| 6.2 | ASI 2: Establecimiento de Requisitos | 91 |
| 6.2.1 | Obtención de los Requisitos del Sistema | 91 |
| 6.2.2 | Identificación de Actores del Sistema | 100 |
| 6.2.3 | Especificación de Casos de Uso | 101 |
| 6.3 | ASI 3: Identificación de Subsistemas de Análisis | 106 |
| 6.3.1 | Descripción de los Subsistemas..... | 106 |
| 6.3.2 | Descripción de los Interfaces entre Subsistemas..... | 107 |
| 6.4 | ASI 4: Análisis de los Casos de Uso | 108 |

| | | |
|-------------------|---|------------|
| 6.4.1 | Caso de Uso 1: Registro | 108 |
| 6.4.2 | Caso de Uso 2: Inicio de sesión..... | 109 |
| 6.4.3 | Caso de uso 3: Asignar robot | 110 |
| 6.4.4 | Caso de uso 4: Creación de un empleado | 111 |
| 6.4.5 | Caso de uso 5: Borrado de un cliente..... | 112 |
| 6.5 | ASI 8: Definición de Interfaces de Usuario | 113 |
| 6.5.1 | Descripción de la Interfaz | 113 |
| 6.5.2 | Diagrama de Navegabilidad..... | 124 |
| 6.6 | ASI 10: Especificación del Plan de Pruebas | 125 |
| 6.6.1 | Pruebas automáticas | 125 |
| 6.6.2 | Pruebas manuales..... | 127 |
| Capítulo 7 | Diseño del Sistema de Información | 129 |
| 7.1 | DSI 3: Diseño de Casos de Uso Reales | 130 |
| 7.1.1 | Caso de uso 1: Registro | 130 |
| 7.1.2 | Caso de uso 2: Asignar robot | 131 |
| 7.1.3 | Caso de uso 3: Creación de un empleado | 132 |
| 7.1.4 | Caso de uso 4: Borrado de un cliente..... | 133 |
| 7.1.5 | Caso de uso 5: Generación de incidencia..... | 134 |
| 7.2 | DSI 4: Diseño de Clases | 135 |
| 7.2.1 | REST API – Backend..... | 136 |
| 7.2.2 | Aplicación móvil – Frontend | 138 |
| 7.2.3 | Pantallas | 138 |
| 7.2.4 | Widgets | 138 |
| 7.3 | DSI 5: Diseño de la Arquitectura de Módulos del Sistema | 139 |
| 7.3.1 | DSI 5.1 Diseño de Módulos del Sistema | 139 |
| 7.4 | DSI 6: Diseño Físico de Datos..... | 141 |
| 7.4.1 | Descripción del SGBD Usado | 141 |
| 7.4.2 | Integración del SGBD en Nuestro Sistema | 142 |
| 7.4.3 | Diagrama E-R..... | 143 |
| 7.4.4 | Diagrama relacional | 144 |



| | | |
|--------------------|---|------------|
| 7.5 | DSI 10: Especificación Técnica del Plan de Pruebas..... | 145 |
| 7.5.1 | Pruebas Unitarias..... | 145 |
| 7.5.2 | Pruebas de Seguridad | 156 |
| 7.5.3 | Pruebas de Integración | 157 |
| 7.5.4 | Pruebas E2E y en entornos reales | 159 |
| 7.5.5 | Pruebas de Usabilidad | 161 |
| Capítulo 8 | Construcción del Sistema de Información | 163 |
| 8.1 | CSI 1: Preparación del Entorno de Generación y Construcción | 164 |
| 8.1.1 | Estándares y normas seguidos | 164 |
| 8.1.2 | Lenguajes de programación y herramientas..... | 167 |
| 8.2 | CSI 3: Ejecución de las Pruebas Unitarias..... | 171 |
| 8.3 | CSI 4: Ejecución de las Pruebas de Integración..... | 172 |
| 8.4 | CSI 5: Ejecución de las Pruebas del Sistema..... | 173 |
| 8.4.1 | Pruebas E2E y en entornos reales | 173 |
| 8.4.2 | Prueba de Usabilidad | 175 |
| 8.5 | CSI 6: Elaboración de los Manuales de Usuario | 178 |
| 8.5.1 | Manual de Instalación o Despliegue | 178 |
| 8.5.2 | Manual de Usuario..... | 180 |
| 8.5.3 | Manual del Programador..... | 193 |
| Capítulo 9 | Conclusiones y Ampliaciones | 195 |
| 9.1 | Conclusiones | 196 |
| 9.2 | Ampliaciones..... | 197 |
| Capítulo 10 | Apéndices | 199 |
| 10.1 | Plan de gestión de riesgos..... | 200 |
| 10.1.1 | Valoración de riesgos | 200 |
| 10.1.2 | Gestión de riesgos | 205 |
| 10.2 | Referencias Bibliográficas | 208 |
| 10.3 | Contenido entregado en los anexos | 209 |
| 10.3.1 | Contenidos..... | 209 |
| 10.3.2 | Contenido del archivo comprimido del código..... | 209 |

Capítulo 1 ¿QUÉ ES ESTE TRABAJO?

1.1 RESUMEN

Este trabajo detalla el desarrollo de una aplicación móvil destinada a optimizar la gestión y mantenimiento de robots cortacéspedes. La aplicación, diseñada para ofrecer una solución integral en la gestión de estos robots, se orienta principalmente a empresas dedicadas a la jardinería y mantenimiento de áreas verdes, así como a usuarios que poseen o gestionan múltiples dispositivos de este tipo.

La aplicación permite a las empresas controlar la gestión de robots, clientes y empleados. Además, ofrece recursos como monitorización en tiempo real, alertas de incidencias y control remoto.

Desarrollada para dispositivos móviles, los usuarios pueden acceder a la información de sus robots de manera remota y en cualquier momento. Además, ofrece una gestión de roles, para restringir a determinados usuarios el acceso a funcionalidades o robots específicos.

1.2 PALABRAS CLAVE

Mower Control, Automower, Cortacésped, Jardinería, Aplicación móvil, Husqvarna, Mantenimiento, Gestión



1.3 ABSTRACT

This work details the development of a mobile application aimed at optimizing the management and maintenance of robotic lawn mowers. The application, designed to offer a comprehensive solution for managing these robots, is primarily targeted at companies dedicated to gardening and green area maintenance, as well as users who own or manage multiple devices of this type.

The application allows companies to control the management of mowers, clients, and employees. Additionally, it offers resources such as real-time monitoring, incident alerts, and remote control.

Developed for mobile devices, users can access their robots' information remotely and at any time. Moreover, it features role management to restrict certain users' access to specific functionalities or robots.

1.4 KEYWORDS

Mower Control, Automower, Lawnmower, Gardening, Mobile application, Husqvarna, Maintenance, Management

Capítulo 2 PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI



2.1 PSI 1: INICIO DEL PLAN DE SISTEMAS DE INFORMACIÓN

2.1.1 PSI 1.1: Análisis de la Necesidad del PSI

El Cocañu S.L es una empresa que se dedica a la venta y mantenimiento de robots cortacésped de la marca Husqvarna. Debido al gran número de robots ya instalados y la tendencia de seguir aumentando ese número considerablemente, las alternativas que se proporcionan hoy en día para la gestión de todos estos robots son bastante escasas y carecen de bastantes funciones. Ya que estas aplicaciones están pensadas para el uso particular no para uso profesional y la gestión de un número elevado de robots. Por lo que se vio una oportunidad y un espacio para desarrollar una aplicación enfocada a las empresas.

El producto final esperado es una aplicación móvil completamente funcional, intuitiva y fácil de usar. El objetivo esperado de la aplicación es que permita consultar todos los robots y su estado actual, proporcionando una manera sencilla de buscarlos. Además, una gestión de clientes y empleados, que irán asociados con los robots. Por último, un control de incidencias y un manejo remoto básico de los robots. La aplicación estará disponible para dispositivos iOS y Android.

Aunque la aplicación se enfocará para la empresa ya comentada, estará diseñada y se podrá utilizar por cualquier otra empresa o usuarios que requieran de su uso.

2.1.2 PSI 1.2: Identificación del Alcance del PSI

El proyecto de fin de grado se enfocará en la creación de una aplicación móvil que mejorará la gestión y el manejo de los robots cortacésped distribuidos por una empresa. Las áreas que se verán afectadas y que serán el foco de este proyecto son las siguientes:

- **Visualización de los robots cortacésped:** Actualmente, este proceso con las alternativas hoy en día es difícil, ya que no hay una manera sencilla de buscar. La aplicación proporcionara un buscador sencillo para encontrar de manera más rápida el robot que quiera el usuario.
- **Gestión de clientes y empleados:** El proyecto tiene como objetivo además llevar un control de los clientes, así como de los empleados, para poder relacionar cada robot con el cliente al que se le vendió o el empleado que se hace cargo de su mantenimiento.
- **Control de incidencias:** Un punto muy importante de este proyecto, es que la empresa esté al tanto de las incidencias que tienen los robots, y además tener un historial de estas.



- **Manejo remoto básico de los robots:** La aplicación proporcionara un control básico de los robots, en caso de que se requiere hacer alguna acción a distancia y el cliente no esté disponible.

Los objetivos estratégicos que se deben considerar para que este trabajo de fin de grado se considere un éxito incluyen:

- **Funcionalidad y facilidad de uso:** La aplicación debe ser usable, intuitiva y adaptada para los usuarios que la van a utilizar.
- **Información necesaria y actualizada:** La información que se proporcione de los robots debe ser la necesaria para los usuarios y debe estar actualizada casi en tiempo real.
- **Extensible y adaptable:** El proyecto debe estar pensado para que sea fácilmente extensible con nuevas funcionalidades y adaptable a cambios en el funcionamiento de los robots.

2.1.3 PSI 1.3: Determinación de responsables

- **El proyectante** se encargará del diseño y desarrollo de la aplicación móvil, esto incluye la creación de la interfaz de usuario, así como toda la lógica necesaria para el correcto funcionamiento. Esta responsabilidad no se compartirá con otros proyectantes.
- **La empresa** se encargará de proporcionar todo el material necesario para las pruebas en escenarios reales. Además, estará disponible para resolver dudas que tengan impacto en el desarrollo del proyecto.
- **El tutor del proyecto.** Será el responsable de comprobar que se han conseguido los objetivos intermedios de todas las fases, así como realizar un seguimiento de todo el proyecto.



2.2 PSI 2: DEFINICIÓN Y ORGANIZACIÓN DEL PSI

2.2.1 PSI 2.1: Especificación del Ámbito y Alcance

Fase 1: Desarrollo de API básica con base de datos

Se creará una API REST que llevará acabo toda la lógica del sistema y almacenara de manera centralizada toda la información que requiere la aplicación.

Objetivos de la fase:

- Obtención de la información de los robots de manera automática.
- Registro e identificación de empresas y usuarios.
- Guardado de toda la información (robots, clientes, empleados, etc.).
- Creación de nuevos clientes y empleados.
- Generación automática de incidencias.

Fase 2: Desarrollo de la aplicación móvil.

Se desarrollará una aplicación móvil multiplataforma, intuitiva y fácil de usar, que servirá como la única interfaz para los usuarios finales.

Objetivos de la fase:

- Visualización de toda la información almacenada en la Rest API.
- Búsqueda de los robots.
- Registro e identificación de empresas y usuarios.
- Asignación de robots a cliente y empleados, y la gestión de ambos.

Fase 3: Desarrollo del manejo remoto básico

Se realizará un desarrollo para el manejo remoto básico de los robots directamente desde la aplicación.

Objetivos de la fase:

- Mandar eventos desde la API REST a la API externa,
- Visualización del estado actual del robot.
- Permitir al usuario arrancar o parar el robot desde la aplicación.



Fase 4: Pruebas

Se llevarán a cabo pruebas de toda la funcionalidad incluso en entornos reales, así como pruebas de usabilidad y de seguridad para asegurar el correcto funcionamiento de todo el sistema.

Objetivos de la fase:

- Evaluar la funcionalidad de mediante pruebas unitarias.
- Evaluar la integración de los sistemas mediante pruebas de integración.
- Evaluar el funcionamiento completo con pruebas e2e en entornos reales.
- Evaluar la usabilidad con usuarios reales.

Fase 5: Despliegue y distribución

Se publicará la aplicación para que cualquier empresa pueda usarla.

Objetivos de la fase:

- Se desplegará la API REST en un servidor remoto.
- Se distribuirá la aplicación móvil.



2.2.2 PSI 2.2: Organización del PSI

A continuación, se detallan los equipos de trabajo necesarios para realizar las especificaciones mencionadas anteriormente.

| USUARIO | PERFIL | FUNCIÓN |
|-------------------------------------|----------------|---|
| EQUIPO DE SUPERVISION | | |
| Director del proyecto | Informático | Supervisión y seguimiento del proyecto. |
| EQUIPO DE DESARROLLO | | |
| Alumno | Informática | Planificación, diseño, documentación y desarrollo de las 5 fases descritos |
| EQUIPO DE PRUEBAS DE USUARIO | | |
| Empresa y empleados | No informático | Pruebas en escenarios reales y pruebas de usabilidad y accesibilidad del sistema. |



2.3 PSI 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE

2.3.1 PSI 3.1: Selección y Análisis de Antecedentes

A continuación, se estudiarán productos similares a el sistema a desarrollar junto a sus ventajas e inconvenientes

2.3.1.1 Automower Connect

Automower Connect es una aplicación desarrolla por la propia marca de robots cortacésped Husqvarna. Desde esta aplicación el usuario podrá consultar el estado de su robot, así como realizar múltiples acciones.

Ventajas: Aplicación oficial, múltiples funciones, fácil de usar

Desventajas: Enfocada al cliente final que compra el robot. Manejo de muchos robots muy complicado.

<https://www.husqvarna.com/es/servicios/automower-connect/>

2.3.1.2 Husqvarna Fleet Services

Husqvarna Fleet Services es también una aplicación desarrollada por la propia marca. Desde esta aplicación se puede tener un control de todas las máquinas que dispongas y realizar un seguimiento.

Ventajas: Aplicación oficial, seguimiento de múltiples máquinas

Desventajas: Orientada para clientes profesionales. Difícil de usar. General, para todo tipo de productos.

<https://www.husqvarna.com/es/servicios/fleet-services/>

Capítulo 3 PSI 7: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

FASE DE PLANIFICACIÓN

PSI



3.1 PSI 7.1: IDENTIFICACIÓN DE LAS NECESIDADES DE INFRAESTRUCTURA TECNOLÓGICA

3.1.1 Alternativas tecnológicas

En el ámbito del desarrollo de software, es habitual enfrentarse a la disponibilidad de diversas herramientas y soluciones destinadas a resolver los mismos retos o necesidades. Estas opciones pueden diferenciarse por su funcionalidad, coste, eficiencia, capacidad de ampliación y otros aspectos cruciales. Por lo tanto, es esencial llevar a cabo una evaluación cuidadosa de estas opciones para identificar aquella que mejor se ajuste a los requerimientos del proyecto y cumpla con las expectativas de los clientes.

La evaluación no solo considera la viabilidad técnica, sino que también incluye factores económicos, soporte disponible, la comunidad que respalda la tecnología, su madurez y la capacidad de adaptación a futuro. El objetivo principal de este proceso es ofrecer un panorama de las principales opciones tecnológicas que se ajustan a nuestra infraestructura y necesidades de desarrollo. Para ello, analizaremos los pros y contras, así como la aplicabilidad de cada alternativa en nuestro proyecto específico, lo que nos permitirá tomar decisiones basadas en información detallada y precisa.

Cabe destacar que la "mejor" tecnología no siempre corresponde a la más reciente o la más popular, sino a aquella que cumple de manera más efectiva con los requisitos del proyecto, manteniendo un balance adecuado entre coste, eficiencia, seguridad y facilidad de mantenimiento.

Con base en estos criterios, procederemos a investigar las distintas opciones tecnológicas a nuestra disposición, realizando un análisis exhaustivo de cada una y comparándolas entre sí para identificar la solución más adecuada para nuestro proyecto.



3.1.1.1 Base de datos

En el mundo de la tecnología y la información, las bases de datos desempeñan un papel crucial al ser el núcleo central para el almacenamiento, manejo y organización de datos, facilitando su conexión lógica y accesibilidad. Fundamentalmente, se distinguen dos categorías principales de bases de datos, cada una con sus características y aplicaciones específicas: las bases de datos relacionales y las no relacionales, conocidas también como NoSQL.

3.1.1.1.1 Base de datos relacional

Las bases de datos de tipo relacional se fundamentan en un modelo estructurado donde los datos se agrupan en tablas. Cada tabla consiste en filas y columnas, donde las filas representan registros individuales y las columnas atributos de estos registros. La potencia de este modelo reside en su capacidad para enlazar tablas a través de claves primarias y foráneas, estableciendo así relaciones significativas entre distintos conjuntos de datos. Esta arquitectura no solo asegura la integridad y consistencia de los datos, sino que también posibilita la ejecución de consultas y transacciones complejas mediante SQL (Structured Query Language), un lenguaje de consulta especializado. Entre las plataformas más destacadas que implementan este modelo se encuentran MySQL, conocido por su flexibilidad y compatibilidad con numerosos sistemas operativos; PostgreSQL, que se distingue por su conformidad con estándares y extensibilidad; Oracle Database, reconocido por su robustez y características avanzadas de seguridad; y Microsoft SQL Server, valorado por su integración con herramientas de Microsoft y funcionalidades de análisis.

3.1.1.1.2 Base de datos no relacional

Por otro lado, las bases de datos no relacionales, o NoSQL, se caracterizan por su enfoque flexible hacia la estructuración de datos. A diferencia de las bases relacionales, no se limitan a un esquema fijo ni dependen de tablas interrelacionadas, lo que les permite manejar datos semiestructurados o no estructurados de manera eficiente. Esta flexibilidad se traduce en una mayor escalabilidad y adaptabilidad a diferentes tipos de aplicaciones, desde almacenar documentos hasta gestionar grandes volúmenes de datos en tiempo real o relaciones complejas entre entidades. Entre los ejemplos más notorios de bases de datos NoSQL se encuentran MongoDB, que ofrece un modelo basado en documentos ideal para aplicaciones web modernas, Cassandra, diseñada para manejar grandes cantidades de datos distribuidos a través de muchos servidores con una alta disponibilidad y Neo4j, una base de datos orientada a grafos que sobresale en el manejo de relaciones complejas e interconectadas entre los datos, facilitando la realización de análisis de redes y sistemas complejos.



3.1.1.2 Servidor

Para alojar la REST API y la base de datos, se deben considerar diversas alternativas, evaluando sus características y ventajas según las necesidades específicas del proyecto. A continuación, se describen cuatro opciones distintas:

3.1.1.2.1 Servidor físico

Un servidor físico, también conocido como un servidor dedicado, es una unidad de hardware exclusiva que proporciona servicios y recursos a computadoras en una red local o en internet. Este tipo de servidor se caracteriza por contar con recursos dedicados como procesador, memoria RAM y espacio de almacenamiento, lo que generalmente se traduce en un rendimiento superior al no haber una capa de virtualización que pueda generar latencias. La gestión de estos servidores es directa y no depende de software adicional para la virtualización. Sin embargo, esta exclusividad conlleva mayores costos tanto en la adquisición inicial como en el mantenimiento, y la escalabilidad puede ser limitada, requiriendo intervención física para modificaciones en los recursos.

3.1.1.2.2 Máquina virtual

Las máquinas virtuales son emulaciones de sistemas informáticos que se ejecutan sobre un servidor físico a través de un hipervisor. Ofrecen un aislamiento total, funcionando cada una como si fuera un sistema independiente con su propio sistema operativo y aplicaciones. Esta independencia permite una utilización de recursos más eficiente al permitir la coexistencia de múltiples VMs en un mismo servidor físico. Las VMs destacan por su capacidad para crear copias de seguridad y recuperaciones a través de snapshots y clonaciones, así como por la facilidad para migrar VMs entre servidores sin interrupciones. Sin embargo, el rendimiento puede verse ligeramente afectado por la sobrecarga que introduce el hipervisor.

3.1.1.2.3 Contenedores Docker

Los contenedores ofrecen una forma eficiente de encapsular y ejecutar aplicaciones junto con sus dependencias en un formato estandarizado, asegurando la consistencia en diferentes entornos. Docker es una de las soluciones más populares para contenedores, proporcionando una opción más ligera que las VMs al compartir el mismo sistema operativo base y ofreciendo aislamiento entre contenedores. Esto mejora el rendimiento y reduce los tiempos de inicio de las aplicaciones. La portabilidad es otra ventaja significativa, permitiendo la ejecución en cualquier sistema compatible con Docker. Aunque el aislamiento puede no ser tan fuerte como en las VMs, la seguridad puede ser reforzada con herramientas y prácticas adecuadas, siendo una opción preferente para arquitecturas de microservicios por su capacidad de despliegue y escalado de servicios de forma individual.



3.1.1.2.4 Clúster de Kubernetes

Un clúster de Kubernetes ofrece una plataforma para automatizar la implementación, escalado y operaciones de aplicaciones contenedorizadas. Funciona orquestando contenedores en múltiples hosts, optimizando el uso de recursos y facilitando la administración de aplicaciones a gran escala. Kubernetes no solo proporciona alta disponibilidad y tolerancia a fallos, sino que también automatiza el despliegue de contenedores, la gestión de servicios y la escalabilidad. Esta solución es ideal para entornos de producción complejos donde la demanda de servicios puede variar significativamente, permitiendo una gestión eficiente y dinámica de los recursos.



3.1.1.3 Comunicación cliente-servidor

Para facilitar el intercambio de información entre un cliente y un servidor, se emplean diversos protocolos diseñados para tal fin. A continuación, se describen algunas de las alternativas más destacadas:

3.1.1.3.1 REST API

Esta técnica es ampliamente reconocida por su eficacia y se fundamenta en la utilización de principios arquitectónicos uniformes. Opera mediante el protocolo HTTP, aprovechando verbos HTTP estándar tales como GET, POST, PUT y DELETE para la transmisión de datos. Las respuestas suelen ser en formatos JSON o XML, destacando por su sencillez y la facilidad con la que puede integrarse en diversos sistemas.

3.1.1.3.2 SOAP API

Utiliza XML para el intercambio de mensajes, permitiendo que aplicaciones en distintas plataformas se comuniquen entre sí por medio de HTTP y SMTP. Es conocido por su rigidez y por ofrecer un alto nivel de seguridad y transacciones más estructuradas.

3.1.1.3.3 GraphQL API

Introduce un lenguaje de consulta flexible y un entorno de ejecución para APIs que facilita a los clientes la solicitud de datos específicos. Gracias a la definición de un único punto de acceso para las peticiones, se minimiza el tráfico innecesario, optimizando así la transferencia de datos.

3.1.1.3.4 WebSockets

Añadiendo otra alternativa, los WebSockets proporcionan un canal de comunicación bidireccional y persistente entre el cliente y el servidor. Esto permite una interacción en tiempo real, siendo ideal para aplicaciones web que requieren una rápida actualización de datos. A diferencia de HTTP, que es unidireccional, WebSockets mantiene la conexión abierta, facilitando un flujo continuo de datos sin necesidad de realizar múltiples solicitudes.



3.1.1.4 Envío de notificaciones

El sistema, al ser una aplicación móvil, tendrá notificaciones que avisaran al usuario de determinados eventos que ocurran en el sistema. Actualmente, existen varios servicios que simplifican este proceso, se describen a continuación:

3.1.1.4.1 Firebase Cloud Messaging (FCM)

Firestore Cloud Messaging es un servicio de mensajería en la nube desarrollado por Google, diseñado específicamente para enviar notificaciones y mensajes a aplicaciones Android, iOS y web de manera confiable y escalable. FCM permite a los desarrolladores enviar mensajes a grupos específicos de usuarios, segmentados por comportamiento o atributos, y ofrece capacidades avanzadas como la entrega de notificaciones a dispositivos sin conexión y la gestión de tokens de registro automáticamente. Es una opción popular debido a su integración directa con otros servicios de Firestore y su sólida infraestructura de entrega de mensajes.

3.1.1.4.2 Amazon Simple Notification Service (SNS)

Amazon SNS es un servicio de notificaciones totalmente administrado por Amazon Web Services (AWS) que facilita el envío de mensajes a una variedad de dispositivos móviles, incluidos Android, iOS, FireOS y aplicaciones web. SNS permite enviar notificaciones push personalizadas, así como mensajes de texto SMS, correos electrónicos y mensajes a servicios de mensajería como Amazon SQS y Lambda. Es altamente escalable y flexible, con capacidad para enviar mensajes a millones de usuarios simultáneamente a través de múltiples plataformas.

3.1.1.4.3 OneSignal

OneSignal es una plataforma de mensajería y automatización que permite a los desarrolladores enviar notificaciones push a aplicaciones móviles y sitios web. Con soporte para Android, iOS, Windows y plataformas web, OneSignal se destaca por su facilidad de uso y su capacidad para integrarse rápidamente con aplicaciones existentes mediante SDKs simples. Ofrece características avanzadas como la segmentación basada en comportamientos del usuario, pruebas A/B para optimización de campañas y análisis detallados de la efectividad de las notificaciones entregadas.

3.1.2 Comparativa

3.1.2.1 Base de datos

| Característica | Bases de Datos Relacionales | Bases de Datos No Relacionales |
|-------------------------------|---|--|
| <i>Modelo de datos</i> | Tablas con filas y columnas. Estructura predefinida. | Variable Estructura flexible. |
| <i>Escalabilidad</i> | Vertical | Horizontal |
| <i>Lenguaje de consulta</i> | SQL | Variable según el tipo |
| <i>Esquema</i> | Estricto y definido previamente. | Flexible y dinámico |
| <i>Rendimiento</i> | Optimizado para transacciones y complejas consultas relacionales. | Alta velocidad en operaciones de lectura y escritura |
| <i>Soporte y comunidad</i> | Amplio soporte y grandes comunidades | Fuerte y en crecimiento |
| <i>Integración</i> | Integración sólida | Buen soporte para integración |
| <i>Transacciones</i> | Soporte fuerte | Variable |
| <i>Consistencia</i> | Inmediata | Eventual |
| <i>Seguridad</i> | Características avanzadas | Variable ampliamente |
| <i>Costo y licenciamiento</i> | Muchas opciones gratuitas y de código abierto. | Amplia gama de opciones |

3.1.2.2 Servidor

| Característica | Servidor Físico | Máquina Virtual | Contenedores Docker | Clúster de Kubernetes |
|-------------------------------|------------------------|------------------------|----------------------------|------------------------------|
| <i>Escalabilidad</i> | Limitada | Alta | Muy alta | Extremadamente alta |
| <i>Despliegue</i> | Lento y manual | Relativamente rápido | Rápido | Muy rápido y automatizado |
| <i>Portabilidad</i> | Baja | Alta | Muy alta | Alta |
| <i>Mantenimiento</i> | Alto | Medio | Bajo | Medio a alto |
| <i>Seguridad</i> | Alta | Alta | Media a alta | Alta |
| <i>Aislamiento</i> | Completo | Alto. | Medio | Alto |
| <i>Eficiencia de Recursos</i> | Baja | Media | Alta | Alta |
| <i>Costo</i> | Alto | Medio | Variable | Potencialmente elevado |
| <i>Complejidad</i> | Media a alta | Media | Media | Alta |
| <i>Soporte y Comunidad</i> | Variable | Alta | Muy alta | Muy alta |

3.1.2.3 Comunicación cliente-servidor

| Característica | API REST | API SOAP | API GraphQL | WebSocket |
|---------------------------------|-----------------|-------------------------------|---------------------|------------------------|
| <i>Rendimiento</i> | Alto | Bajo | Alto | Muy |
| <i>Estándares</i> | HTTP/HTTPS | Basado en estándares formales | HTTP/HTTPS | WebSocket |
| <i>Flexibilidad</i> | Alta | Baja | Muy alta | Alta |
| <i>Tipo de Datos</i> | JSON y XML | XML | JSON | Texto y datos binarios |
| <i>Caché</i> | Soportado | Uso complejo | Manual | N/A |
| <i>Estado</i> | Sin estado | Con o sin estado | Sin estado | Con estado |
| <i>Facilidad de Uso</i> | Alta | Media a baja | Alta | Media |
| <i>Seguridad</i> | Implementación | Alta | Implementación | Implementación |
| <i>Madurez de la Tecnología</i> | Muy madura | Muy madura | Relativamente nueva | Madura |
| <i>Comunidad y Soporte</i> | Enorme | Sólida | Crecimiento | Fuerte |
| <i>Escalabilidad</i> | Alta | Baja | Eficiente | Excelente |



3.2 PSI 7.2: SELECCIÓN DE LA ARQUITECTURA TECNOLÓGICA

Una vez analizadas todas las alternativas con sus características, ventajas y desventajas de cada una, en este apartado se explican las opciones elegidas y el porqué de cada una de ellas.

3.2.1 Base de datos

Se ha decidido implementar una base de datos relacional para el proyecto, seleccionando MySQL de entre una gama de opciones como PostgreSQL, MySQL, Oracle, y otras, principalmente debido a su naturaleza gratuita, la amplia disponibilidad de documentación y el soporte extensivo también ofrecido sin costo.

El diseño inherente de las bases de datos relacionales permite el manejo eficiente de relaciones complejas entre distintas entidades o tablas, una capacidad en la que MySQL, como uno de los sistemas de gestión de bases de datos relacionales más establecidos, destaca notablemente. Esto es crucial para sistemas donde las relaciones entre los datos son complejas y fundamentales para su operación correcta.

Además, la elección de MySQL se sustenta en criterios clave que enfatizan la integridad, eficiencia y confiabilidad en el manejo de datos. La garantía de la integridad y la consistencia de la información es una prioridad en cualquier sistema, y MySQL sobresale en ofrecer mecanismos que salvaguardan la precisión y cohesión de los datos. Esto se logra mediante el uso de restricciones, claves primarias y secundarias, y reglas de integridad que verifican que la información introducida y actualizada se adhiera a las especificaciones y políticas de negocio establecidas.

En conclusión, optar por MySQL refleja un compromiso con la confiabilidad, la integridad de los datos y una organización sistemática de la información, estableciendo una base sólida y segura para la gestión de datos.



3.2.2 Servidor

Se ha seleccionado el uso de contenedores Docker para el servidor, basándose en sus numerosas ventajas para el desarrollo y despliegue de aplicaciones. Docker proporciona un ambiente altamente eficiente y móvil, asegurando una operación fluida de la aplicación en distintas plataformas, lo que elimina los contratiempos de implementación a menudo encontrados cuando los sistemas funcionan en entornos de desarrollo, pero no en producción. Esta consistencia es crucial, permitiendo al equipo de desarrollo estar seguro de que lo que se crea y prueba localmente coincidirá con lo que se desplegará en producción, evitando así cualquier discrepancia o incompatibilidad inesperada.

La adopción de Docker también simplifica significativamente la gestión de dependencias, ya que cada contenedor opera de manera independiente con todos los componentes necesarios incluidos en su propia imagen. Esto no solo mejora la reproducibilidad de la aplicación en diferentes entornos, sino que también facilita la actualización y mantenimiento de las aplicaciones sin afectar el funcionamiento de otras en el sistema. Así, Docker se convierte en una herramienta indispensable para equipos que buscan eficiencia, flexibilidad y consistencia en el ciclo de vida del desarrollo de software.

3.2.3 Comunicación cliente-servidor

Se ha seleccionado utilizar una API REST basándose en su simplicidad y capacidad de adaptación. Un beneficio clave de adoptar esta arquitectura es su facilidad de comprensión y ejecución, gracias a su uso de métodos HTTP, fundamentales para el funcionamiento de la web. Esto asegura una uniformidad en las operaciones, simplificando el proceso de desarrollo y mantenimiento.

Otro aspecto favorable de las API REST es su habilidad para soportar múltiples formatos de datos, incluidos JSON, XML, entre otros, sin imponer limitaciones estrictas. Esto ofrece una flexibilidad significativa para cumplir con los requerimientos particulares de cada proyecto. Esta flexibilidad resulta invaluable para la integración con diversas plataformas, ya sea una aplicación móvil o un portal web administrativo. En vez de depender de varias soluciones específicas para cada interfaz, REST permite la implementación de una solución unificada que atiende eficazmente a todas las necesidades implicadas.



3.2.4 Envió de notificaciones

Como servicio para el envío de notificaciones a dispositivos móviles, se selecciona Firebase Cloud Messaging debido a su uso extendido en la industria y su fácil integración con multitud de tecnologías de desarrollo. Firebase Cloud Messaging ofrece una robusta infraestructura que permite enviar mensajes personalizados de manera eficiente a una amplia variedad de dispositivos móviles, incluyendo Android y iOS. Su escalabilidad y confiabilidad son fundamentales para asegurar que las notificaciones lleguen de manera oportuna y efectiva a los usuarios finales, independientemente del tamaño del público objetivo o la complejidad de la aplicación.



Capítulo 4 ESTUDIO DE VIABILIDAD DEL SISTEMA

FASE DE DESARROLLO

EVS



4.1 EVS 4, 5 Y 6: ESTUDIO Y VALORACIÓN DE ALTERNATIVAS DE SOLUCIÓN Y SELECCIÓN DE ALTERNATIVA FINAL

Basándonos en las tecnologías seleccionadas, hemos desarrollado tres sistemas potenciales que satisfacen los requisitos especificados. Cada uno de estos sistemas estará compuesto por un Backend (REST API) y una o dos interfaces de usuario, dependiendo si se desarrolla una aplicación móvil multiplataforma, o aplicaciones para cada sistema operativo (Android y iOS). La base de datos elegida en el apartado anterior es MySQL por lo que todos los sistemas propuestos a continuación son compatibles con dicha tecnología.

4.1.1 Sistema 1 – Android y iOS Nativos con Kotlin/Swift y Spring con Java

4.1.1.1 Descripción

Este sistema utiliza Kotlin para el desarrollo del frontend de Android y Swift para iOS, ambos lenguajes están optimizados para sus respectivas plataformas nativas. El backend se desarrolla utilizando Spring, un robusto framework de Java diseñado para facilitar la creación de aplicaciones en el servidor.

4.1.1.2 Ventajas

- Rendimiento óptimo: Al ser plataformas nativas, tanto Android como iOS ofrecen un rendimiento superior y una mejor integración con el hardware del dispositivo.
- Soporte amplio: Kotlin y Swift están fuertemente respaldados por Google y Apple, respectivamente, asegurando actualizaciones regulares y compatibilidad.
- Ecosistema maduro: Java y Spring proporcionan un ecosistema bien establecido con amplias bibliotecas y una comunidad de soporte grande.
- Acceso completo a APIs nativas: Al usar plataformas nativas, se tiene acceso completo a todas las funciones del sistema operativo y APIs de última generación, lo que permite una integración más profunda y características específicas del sistema.
- Seguridad mejorada: Las aplicaciones nativas pueden ser más seguras debido a la naturaleza cerrada de sus ecosistemas, especialmente en iOS.



4.1.1.3 Desventajas

- Desarrollo duplicado: Requiere mantenimiento y desarrollo paralelo para dos plataformas diferentes, lo que puede incrementar los costos y tiempos de desarrollo.
- Curva de aprendizaje: Java/Spring puede ser complejo para nuevos desarrolladores, comparado con otras opciones más modernas y simplificadas.
- Recursos de desarrollo: Mantener equipos separados para iOS y Android puede ser costoso no solo en términos de desarrollo, sino también en la gestión y formación continua.
- Tiempo de puesta en mercado: El desarrollo en múltiples plataformas nativas puede alargar el tiempo de lanzamiento del producto.



4.1.2 Sistema 2 – Flutter con Dart y Node.js con Typescript

4.1.2.1 Descripción

Flutter es un framework de UI que utiliza Dart para desarrollar aplicaciones móviles que compilan a código nativo para múltiples plataformas desde una única base de código. El backend se desarrolla con Node.js usando TypeScript, un superset de JavaScript que añade tipado estático.

4.1.2.2 Ventajas

- Desarrollo unificado: Un solo código fuente para iOS y Android, lo que puede reducir significativamente el tiempo y el costo de desarrollo.
- Ecosistema moderno y en crecimiento: Tanto Flutter como Node.js son tecnologías modernas con comunidades activas y en crecimiento.
- Facilidad de desarrollo: TypeScript ofrece un desarrollo más predecible y seguro con respecto a JavaScript debido al tipado estático.
- Interfaz de usuario consistente: Flutter permite una experiencia de usuario uniforme en múltiples plataformas, lo que puede ser un desafío en desarrollos nativos donde cada plataforma tiene su propio estilo de UI/UX.
- Desarrollo de backend rápido: Node.js es conocido por su eficiencia en manejar operaciones de I/O asíncronas, lo que lo hace ideal para aplicaciones que requieren alta escalabilidad y manejo de muchas solicitudes simultáneas.

4.1.2.3 Desventajas

- Rendimiento: Aunque Flutter se compila al código nativo, el rendimiento puede no ser siempre equivalente al de las aplicaciones nativas, especialmente en gráficos intensivos u operaciones complejas.
- Madurez: Flutter es relativamente nuevo comparado con otras tecnologías de desarrollo nativo, lo que puede presentar desafíos en términos de estabilidad y soporte de terceros.
- Dependencia del framework: Al depender fuertemente de Flutter, cualquier limitación en el framework puede afectar directamente el proyecto, incluyendo retrasos en la adopción de nuevas características del sistema operativo nativo.
- Curva de aprendizaje para Node.js: A pesar de su popularidad, Node.js puede tener una curva de aprendizaje desafiante debido a su naturaleza asíncrona y no bloqueante, especialmente para aquellos sin experiencia en JavaScript o TypeScript.

4.1.3 Sistema 3 - React Native con JavaScript y Flask con Python.

4.1.3.1 Descripción

React Native permite desarrollar aplicaciones móviles usando JavaScript y React, ofreciendo una experiencia cercana a la nativa desde una base de código compartida. Flask es un microframework de Python que es simple y extensible, ideal para prototipos rápidos y aplicaciones ligeras.

4.1.3.2 Ventajas

- Reutilización de código: Similar a Flutter, permite compartir gran parte del código entre iOS y Android.
- Desarrollo rápido: Flask es conocido por su simplicidad y velocidad de desarrollo, mientras que React Native facilita la transición de desarrolladores web a móviles.
- Gran comunidad: Ambas tecnologías tienen una gran comunidad y un vasto ecosistema de librerías y herramientas.
- Gran soporte para integraciones: Tanto React Native como Flask facilitan la integración con otras herramientas y servicios, como bases de datos, sistemas de autenticación, y más, gracias a su amplia comunidad y soporte.
- Costo eficiente: El uso de tecnologías basadas en JavaScript y Python, lenguajes con muchos desarrolladores disponibles, puede reducir los costos de contratación y formación.

4.1.3.3 Desventajas

- Rendimiento: React Native puede enfrentar problemas de rendimiento para tareas intensivas comparado con código nativo.
- Dependencia de terceros: React Native depende en gran medida de librerías de terceros para muchas funciones, lo que puede llevar a problemas de compatibilidad y mantenimiento.
- Complejidad en el backend: Flask, al ser un microframework, puede requerir la adición de numerosas extensiones o un framework más robusto para aplicaciones más complejas.
- Mantenimiento a largo plazo: Los cambios en las dependencias y librerías de terceros en React Native pueden llevar a un mantenimiento más intensivo y desafiante.
- Escalabilidad del backend: Flask, siendo un microframework, puede requerir más trabajo para escalar en comparación con soluciones más robustas como Django o frameworks basados en Node.js.



4.1.4 Selección de alternativa final

Todos los sistemas descritos en el apartado anterior son perfectamente válidos para desarrollar el sistema que se busca hacer en este proyecto. De entre ellos, se ha escogido el sistema 2 como solución final por las razones que se describen a continuación:

4.1.4.1 Desarrollo unificado y eficiencia en costos

El uso de Flutter permite un desarrollo unificado para iOS y Android, reduciendo significativamente los recursos necesarios en términos de tiempo y costos de desarrollo. A diferencia del sistema 1, que requiere desarrollo paralelo utilizando Kotlin y Swift, y enfrenta duplicación en los esfuerzos de mantenimiento y actualización, Flutter ofrece una sola base de código. Esta característica no solo disminuye la carga operativa y técnica, sino que también facilita la consistencia en la experiencia de usuario entre las diferentes plataformas móviles.

4.1.4.2 Ecosistema moderno con respaldo corporativo

Flutter, apoyado por Google, y Node.js, ampliamente adoptado en la industria tecnológica, ofrecen un ecosistema moderno y en constante evolución. La robusta comunidad detrás de estas tecnologías asegura un flujo constante de actualizaciones, soporte y una vasta biblioteca de plugins y herramientas que pueden acelerar el desarrollo y ofrecer soluciones a problemas comunes de manera eficiente. Además, el respaldo corporativo garantiza que las tecnologías se mantendrán relevantes y soportadas en el futuro previsible.

4.1.4.3 Seguridad y mantenimiento del código con TypeScript

El uso de TypeScript en el desarrollo del backend proporciona ventajas significativas en términos de seguridad y mantenibilidad del código. TypeScript, al ser un superset de JavaScript que incorpora tipado estático, ayuda a detectar errores en tiempo de desarrollo antes que, en producción, lo que mejora la calidad del software y reduce los tiempos de depuración. Esta característica es particularmente valiosa en entornos empresariales donde la robustez y la estabilidad del código son críticas.



4.1.4.4 Experiencia previa y curva de aprendizaje

La decisión de utilizar Node.js y TypeScript también se beneficia de la experiencia previa del equipo de desarrollo. Esta familiaridad con las tecnologías reduce la curva de aprendizaje y permite un ramp-up más rápido en los proyectos, aprovechando las mejores prácticas ya conocidas y la experiencia acumulada para evitar errores comunes y acelerar el ciclo de desarrollo.

4.1.4.5 Adopción de prácticas de desarrollo ágil

Adicionalmente, al optar por Flutter y Node.js, se facilita la adopción de metodologías de desarrollo ágil, dado que ambos entornos soportan ciclos de desarrollo iterativos y rápidos con herramientas que permiten la integración y entrega continuas (CI/CD). Esto es crucial para mantener la competitividad en mercados dinámicos y responder eficazmente a las demandas cambiantes de los usuarios y a las condiciones del mercado.

4.1.4.6 Flexibilidad y escalabilidad

Finalmente, Node.js es particularmente conocido por su escalabilidad en el manejo de aplicaciones que requieren un alto rendimiento en operaciones de entrada/salida y un manejo eficiente de grandes volúmenes de conexiones simultáneas. Esto lo hace ideal para aplicaciones que esperan crecer en términos de usuarios y funcionalidad. Además, la arquitectura de Node.js facilita la integración con tecnologías de bases de datos modernas y servicios de nube, lo que puede ser un factor decisivo para proyectos con visión de futuro y expansión.

Capítulo 5 PLANIFICACIÓN Y GESTIÓN DEL TFG

FASE DE DESARROLLO

5.1 PLANIFICACIÓN DEL PROYECTO

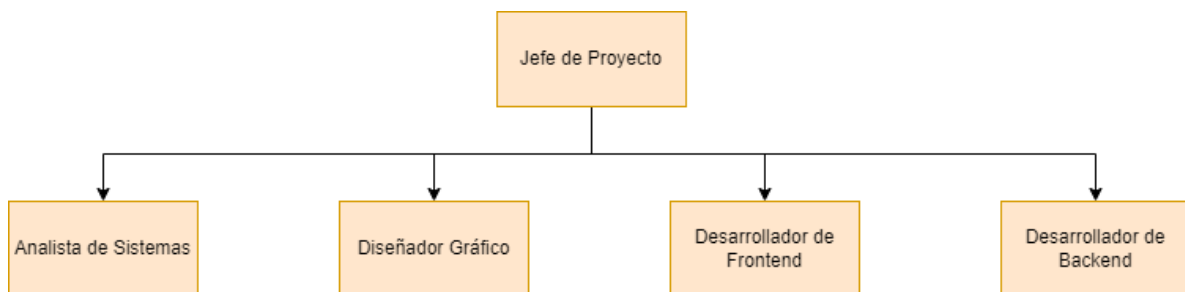
5.1.1 Identificación de Interesados

Los interesados en el proyecto ya se han definido en el apartado [2.1.3 PSI 1.3: Determinación de responsables](#).

5.1.2 OBS y PBS

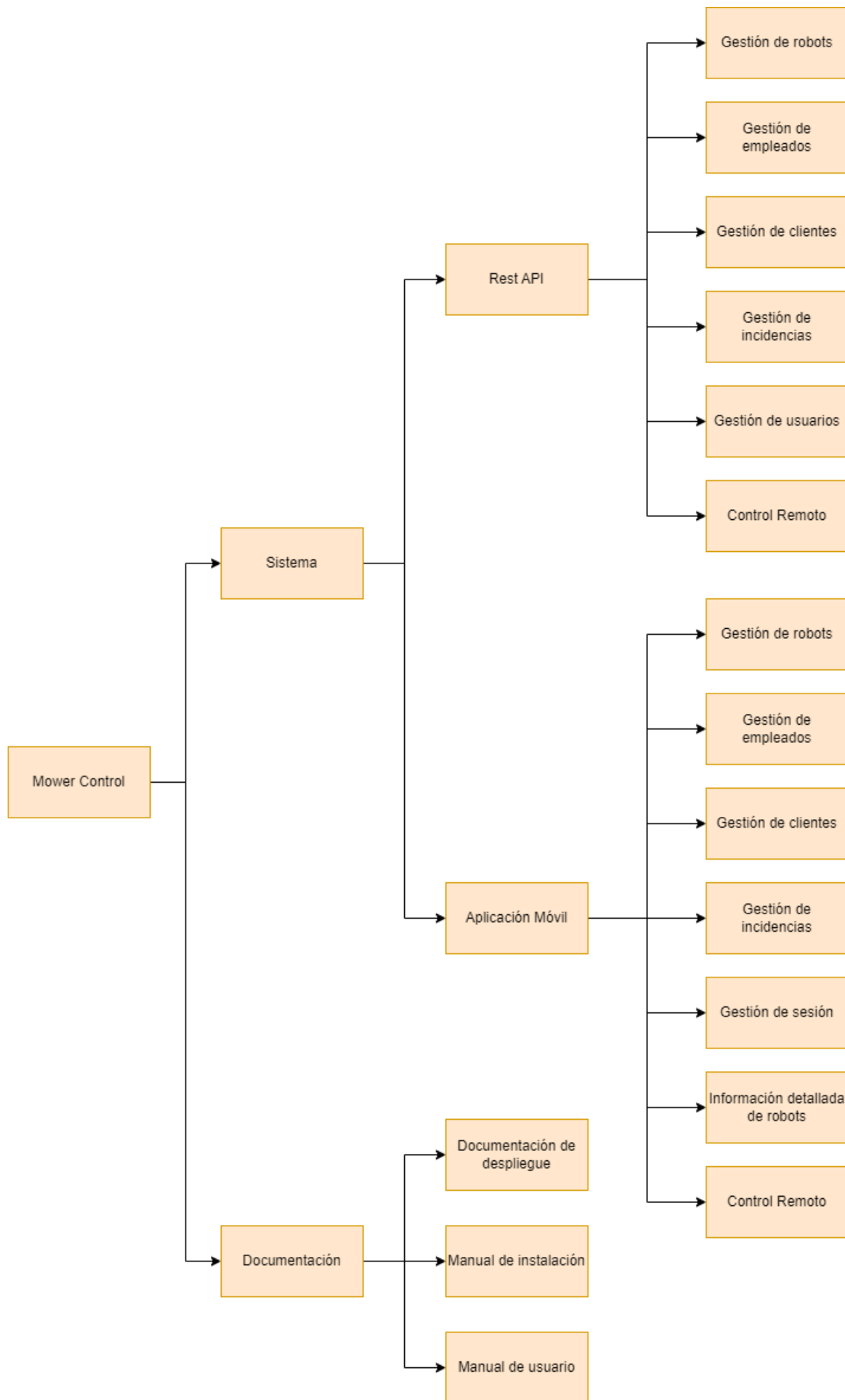
5.1.2.1 OBS

El OBS (Organisational Breakdown Structure o Desglose de la Estructura de Organización) es una representación jerárquica que detalla la estructura organizativa de un proyecto. Este modelo muestra cómo se relacionan los miembros del equipo de proyecto con las unidades organizativas. El OBS sirve como un marco que ayuda en la distribución de tareas y la administración del rendimiento, conectando las responsabilidades y los recursos humanos con las metas y actividades del proyecto. A pesar de que el alumno fue quien desarrolló el proyecto, para su gestión, se creó un escenario en el que desempeñaba distintos roles y funciones dentro de una empresa ficticia.



5.1.2.2 PBS

El PBS (Product Breakdown Structure o Desglose de la Estructura de Producto) es una herramienta esencial en la gestión de proyectos, ya que permite desglosar un producto o proyecto en sus componentes o entregables individuales. Esta estructura facilita la visualización y comprensión de todos los elementos que se requieren para completar un proyecto o para desarrollar un producto. Al descomponer cada parte, el PBS ayuda a identificar los recursos necesarios, planificar las etapas de desarrollo y asignar responsabilidades de manera más efectiva. Además, esta herramienta es crucial para estimar costos, gestionar tiempos y asegurar que cada componente cumpla con los estándares de calidad establecidos.



5.1.3 Planificación Inicial. WBS

El WBS (Desglose de la Estructura de Trabajo) es una metodología que segmenta un proyecto en sus elementos o tareas fundamentales, organizándolos en una estructura jerárquica. Esta herramienta es esencial para organizar y delimitar completamente el alcance del proyecto, mejorando así su gestión y control.

En cuanto a la planificación, se establece que el inicio del proyecto será el 09/01/2023, involucrando a los profesionales mencionados en la sección [5.1.2.2 OBS](#) y siguiendo el calendario establecido.

A continuación, se presenta el diagrama de Gantt del proyecto, que está estructurado en seis niveles principales: análisis y diseño, organización, desarrollo, pruebas, despliegue y distribución y documentación.

5.1.3.1 Análisis y diseño

Esta sección incluye el análisis del sistema y requisitos, así como el diseño de la interfaz y la infraestructura. Se dividirá en tres partes, que se describen a continuación:

5.1.3.1.1 Análisis del proyecto

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F | Nombres de los recursos |
|-----|-----------------|----------|--------------|--------------|---|------------------------------|
| 3 | 1.1.1 | 24 horas | lun 09/01/23 | mié 18/01/23 | | |
| 4 | 1.1.1.1 | 2 horas | lun 09/01/23 | lun 09/01/23 | | Analista de sistemas |
| 5 | 1.1.1.2 | 8 horas | lun 09/01/23 | jue 12/01/23 | 4 | Analista de sistemas |
| 6 | 1.1.1.3 | 2 horas | jue 12/01/23 | jue 12/01/23 | 5 | Analista de sistemas |
| 7 | 1.1.1.4 | 8 horas | vie 13/01/23 | mar 17/01/23 | 6 | Analista de sistemas |
| 8 | 1.1.1.5 | 4 horas | mar 17/01/23 | mié 18/01/23 | 7 | Analista de sistemas;Jefe de |



Tareas:

- Obtención y análisis de stakeholders.
- Obtención inicial de requisitos.
- Validación de requisitos.
- Creación del ERS.
- Validación del ERS.

5.1.3.1.2 Diseño del proyecto

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F | Nombres de los recursos |
|-----|-----------------|----------|--------------|--------------|----|---------------------------|
| 9 | 1.1.2 | 22 horas | jue 19/01/23 | lun 30/01/23 | | |
| 10 | 1.1.2.1 | 6 horas | jue 19/01/23 | vie 20/01/23 | 3 | Diseñador Gráfico |
| 11 | 1.1.2.2 | 2 horas | lun 23/01/23 | lun 23/01/23 | 10 | Diseñador Gráfico |
| 12 | 1.1.2.3 | 12 horas | lun 23/01/23 | vie 27/01/23 | 11 | Diseñador Gráfico |
| 13 | 1.1.2.4 | 2 horas | vie 27/01/23 | lun 30/01/23 | 12 | Diseñador Gráfico;Jefe de |

Tareas:

- Prototipos de pantalla iniciales.
- Validación y refinamiento de los prototipos.
- Wireframe de la aplicación.
- Validación del wireframe.

5.1.3.1.3 Análisis y Diseño de la infraestructura

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F | Nombres de los recursos |
|-----|-----------------|----------|--------------|--------------|----|--------------------------|
| 14 | 1.1.3 | 20 horas | lun 30/01/23 | mar 07/02/23 | | |
| 15 | 1.1.3.1 | 2 horas | lun 30/01/23 | lun 30/01/23 | 9 | Analista de sistemas |
| 16 | 1.1.3.2 | 4 horas | mar 31/01/23 | mié 01/02/23 | 15 | Analista de sistemas |
| 17 | 1.1.3.3 | 4 horas | mié 01/02/23 | jue 02/02/23 | 16 | Analista de sistemas |
| 18 | 1.1.3.4 | 4 horas | jue 02/02/23 | vie 03/02/23 | 17 | Analista de sistemas |
| 19 | 1.1.3.5 | 6 horas | lun 06/02/23 | mar 07/02/23 | | |
| 20 | 1.1.3.5.1 | 2 horas | lun 06/02/23 | lun 06/02/23 | 18 | Analista de sistemas;Jef |
| 21 | 1.1.3.5.2 | 4 horas | lun 06/02/23 | mar 07/02/23 | 20 | Analista de sistemas |

Tareas:

- Análisis de sistemas similares.
- Análisis de alternativas de infraestructura.
- Esquema inicial de la arquitectura del sistema.
- Análisis de alternativas tecnológicas.
- Diseño de la solución final.
- Selección de alternativas tecnológicas.
- Esquema final de la arquitectura del sistema.

5.1.3.2 Organización

Esta sección recoge las reuniones que se harán a lo largo del proyecto, tanto con el tutor del proyecto, como con la empresa para ir validando la aplicación.

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f | Nombres de los recursos |
|----|-------|--|-----------|--------------|--------------|---|-------------------------|
| 22 | 1.2 | Organización | 372 horas | lun 09/01/23 | jue 29/06/23 | | |
| 23 | 1.2.1 | Reunión recurrente con tutor del TFG | 16 horas | lun 09/01/23 | lun 16/01/23 | | Jefe de Proyecto |
| 24 | 1.2.2 | Reunión recurrente con stakeholders externos | 8 horas | mar 27/06/23 | jue 29/06/23 | | Jefe de Proyecto |

Tareas:

- Reunión recurrente con tutor del TFG.
- Reunión recurrente con stakeholders externos.

5.1.3.3 Desarrollo

Esta sección engloba todas las tareas necesarias para implementar todas las funcionalidades del sistema. Se divide en tres subapartados, desarrollo de la API, desarrollo de la aplicación móvil y desarrollo del manejo remoto básico.

5.1.3.3.1 Desarrollo de la API

5.1.3.3.1.1 Desarrollo inicial

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|----|---------|---|-----------|--------------|--------------|----|
| 25 | 1.3 | Desarrollo | 190 horas | mié 08/02/23 | lun 08/05/23 | |
| 26 | 1.3.1 | Desarrollo de la API | 96 horas | mié 08/02/23 | jue 23/03/23 | |
| 27 | 1.3.1.1 | Diseño inicial del modelo de base de datos | 6 horas | mié 08/02/23 | jue 09/02/23 | 2 |
| 28 | 1.3.1.2 | Diseño inicial de la especificación de la API | 4 horas | vie 10/02/23 | lun 13/02/23 | 27 |

Tareas:

- Diseño inicial del modelo de base de datos.
- Diseño inicial de la especificación de la API.

5.1.3.3.1.2 Gestión de empleados

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|------------------------|----------|--------------|--------------|----|
| 29 | 1.3.1.3 | ➤ Gestión de empleados | 11 horas | lun 13/02/23 | jue 16/02/23 | |
| 30 | 1.3.1.3.1 | Creación de empleados | 6 horas | lun 13/02/23 | mié 15/02/23 | 28 |
| 31 | 1.3.1.3.2 | Consulta de empleados | 2 horas | mié 15/02/23 | mié 15/02/23 | 30 |
| 32 | 1.3.1.3.3 | Borrado de empleados | 3 horas | jue 16/02/23 | jue 16/02/23 | 31 |

Tareas:

- Creación de empleados.
- Consulta de empleados.
- Borrado de empleados.

5.1.3.3.1.3 Gestión de clientes

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|-----------------------|----------|--------------|--------------|----|
| 33 | 1.3.1.4 | ➤ Gestión de clientes | 11 horas | vie 17/02/23 | mié 22/02/23 | |
| 34 | 1.3.1.4.1 | Creación de clientes | 6 horas | vie 17/02/23 | lun 20/02/23 | 29 |
| 35 | 1.3.1.4.2 | Consulta de clientes | 2 horas | mar 21/02/23 | mar 21/02/23 | 34 |
| 36 | 1.3.1.4.3 | Borrado de clientes | 3 horas | mar 21/02/23 | mié 22/02/23 | 35 |

Tareas:

- Creación de clientes.
- Consulta de clientes.
- Borrado de clientes.

5.1.3.3.1.4 Gestión de usuarios

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|--|----------|--------------|--------------|----|
| 37 | 1.3.1.5 | ➤ Gestión de usuarios | 14 horas | mié 22/02/23 | mié 01/03/23 | |
| 38 | 1.3.1.5.1 | Registro | 8 horas | mié 22/02/23 | lun 27/02/23 | 33 |
| 39 | 1.3.1.5.2 | Login | 4 horas | lun 27/02/23 | mar 28/02/23 | 38 |
| 40 | 1.3.1.5.3 | Generación automática de credenciales para empleados | 2 horas | mar 28/02/23 | mié 01/03/23 | 39 |

Tareas:

- Registro.
- Login.
- Generación automática de credenciales para empleados.

5.1.3.3.1.5 Gestión de robots

| EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|-----|-----------------|----------|--------------|--------------|----|
| 41 | 1.3.1.6 | 20 horas | mié 01/03/23 | jue 09/03/23 | |
| 42 | 1.3.1.6.1 | 12 horas | mié 01/03/23 | mar 07/03/23 | 37 |
| 43 | 1.3.1.6.2 | 4 horas | mar 07/03/23 | mié 08/03/23 | 42 |
| 44 | 1.3.1.6.3 | 4 horas | mié 08/03/23 | jue 09/03/23 | 43 |

Tareas:

- Obtención y guardado de robots desde un proveedor externo.
- Consulta de robots.
- Asignación de robots.

5.1.3.3.1.6 Gestión de incidencias

| EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|-----|-----------------|----------|--------------|--------------|----|
| 45 | 1.3.1.7 | 22 horas | vie 10/03/23 | mar 21/03/23 | |
| 46 | 1.3.1.7.1 | 4 horas | vie 10/03/23 | lun 13/03/23 | 41 |
| 47 | 1.3.1.7.2 | 2 horas | lun 13/03/23 | lun 13/03/23 | 46 |
| 48 | 1.3.1.7.3 | 2 horas | mar 14/03/23 | mar 14/03/23 | 47 |
| 49 | 1.3.1.7.4 | 2 horas | mar 14/03/23 | mié 15/03/23 | 48 |
| 50 | 1.3.1.7.5 | 12 horas | mié 15/03/23 | mar 21/03/23 | 49 |

Tareas:

- Generación de incidencias.
- Consulta de incidencias.
- Edición de incidencias.
- Actualización periódica del estado de los robots.
- Envío de notificaciones de incidencias.

5.1.3.3.1.7 Gestión de roles

| EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|-----|-----------------|----------|--------------|--------------|----|
| 51 | 1.3.1.8 | 8 horas | mar 21/03/23 | jue 23/03/23 | 50 |

Tareas:

- Gestión de roles.

5.1.3.3.2 Desarrollo de la aplicación móvil

5.1.3.3.2.1 Gestión de robots

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|---|----------|--------------|--------------|----|
| 52 | 1.3.2 | Desarrollo de la aplicación móvil | 80 horas | vie 24/03/23 | lun 01/05/23 | |
| 53 | 1.3.2.1 | Gestión de robots | 12 horas | vie 24/03/23 | mié 29/03/23 | |
| 54 | 1.3.2.1.1 | Visualización información de los robots | 8 horas | vie 24/03/23 | mar 28/03/23 | 26 |
| 55 | 1.3.2.1.2 | Visualización estado de los robots | 4 horas | mar 28/03/23 | mié 29/03/23 | 54 |

Tareas:

- Visualización información de los robots.
- Visualización estado de los robots.

5.1.3.3.2.2 Gestión de clientes

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|---------------------------|----------|--------------|--------------|----|
| 56 | 1.3.2.2 | Gestión de clientes | 12 horas | jue 30/03/23 | mar 04/04/23 | |
| 57 | 1.3.2.2.1 | Visualización de clientes | 4 horas | jue 30/03/23 | vie 31/03/23 | 53 |
| 58 | 1.3.2.2.2 | Creación de cliente | 6 horas | vie 31/03/23 | mar 04/04/23 | 57 |
| 59 | 1.3.2.2.3 | Borrado de cliente | 2 horas | mar 04/04/23 | mar 04/04/23 | 58 |

Tareas:

- Visualización de clientes.
- Creación de cliente.
- Borrado de cliente.

5.1.3.3.2.3 Gestión de empleados

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|----|-----------|----------------------------|----------|--------------|--------------|----|
| 60 | 1.3.2.3 | Gestión de empleados | 12 horas | mié 05/04/23 | lun 10/04/23 | |
| 61 | 1.3.2.3.1 | Visualización de empleados | 4 horas | mié 05/04/23 | jue 06/04/23 | 56 |
| 62 | 1.3.2.3.2 | Creación de empleado | 6 horas | jue 06/04/23 | lun 10/04/23 | 61 |
| 63 | 1.3.2.3.3 | Borrado de empleado | 2 horas | lun 10/04/23 | lun 10/04/23 | 62 |

Tareas:

- Visualización de empleados.
- Creación de empleado.
- Borrado de empleado.

5.1.3.3.2.4 Asignación de robots

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|----|-----------|-------------------------------------|----------|--------------|--------------|----|
| 64 | 1.3.2.4 | Asignación de robots | 10 horas | mar 11/04/23 | vie 14/04/23 | |
| 65 | 1.3.2.4.1 | Visualización de robots sin asignar | 4 horas | mar 11/04/23 | mié 12/04/23 | 60 |
| 66 | 1.3.2.4.2 | Asignación de robot | 6 horas | mié 12/04/23 | vie 14/04/23 | 65 |

Tareas:

- Visualización de robots sin asignar.
- Asignación de robot.

5.1.3.3.2.5 Gestión de incidencias

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|----|-----------|------------------------------|----------|--------------|--------------|----|
| 67 | 1.3.2.5 | Gestión de incidencias | 6 horas | vie 14/04/23 | mar 18/04/23 | |
| 68 | 1.3.2.5.1 | Visualización de incidencias | 4 horas | vie 14/04/23 | lun 17/04/23 | 64 |
| 69 | 1.3.2.5.2 | Edición de incidencias | 2 horas | lun 17/04/23 | mar 18/04/23 | 68 |

Tareas:

- Visualización de incidencias.
- Edición de incidencias.

5.1.3.3.2.6 Gestión de sesión

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|----|-----------|-----------------------------------|----------|--------------|--------------|----|
| 70 | 1.3.2.6 | Gestión de sesión | 14 horas | mar 18/04/23 | lun 24/04/23 | |
| 71 | 1.3.2.6.1 | Login | 4 horas | mar 18/04/23 | mié 19/04/23 | 67 |
| 72 | 1.3.2.6.2 | Registro | 8 horas | mié 19/04/23 | lun 24/04/23 | |
| 73 | 1.3.2.6.2 | Registro básico | 4 horas | mié 19/04/23 | jue 20/04/23 | 71 |
| 74 | 1.3.2.6.2 | Registro contra proveedor externo | 4 horas | vie 21/04/23 | lun 24/04/23 | 73 |
| 75 | 1.3.2.6.3 | Cierre de sesión | 2 horas | lun 24/04/23 | lun 24/04/23 | 72 |

Tareas:

- Login.
- Registro básico.
- Registro contra proveedor externo.
- Cierre de sesión.

5.1.3.3.2.7 Información detallada de robots

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F | |
|-----|-----------------|--|----------|--------------|--------------|----|
| 76 | 1.3.2.7 | Información detallada de robots | 14 horas | mar 25/04/23 | lun 01/05/23 | |
| 77 | 1.3.2.7.1 | Visualización información completa del robot | 4 horas | mar 25/04/23 | mié 26/04/23 | 70 |
| 78 | 1.3.2.7.2 | Visualización cliente asignado | 2 horas | mié 26/04/23 | mié 26/04/23 | 77 |
| 79 | 1.3.2.7.3 | Visualización empleado asignado | 2 horas | jue 27/04/23 | jue 27/04/23 | 78 |
| 80 | 1.3.2.7.4 | Visualización historico de incidencias | 4 horas | jue 27/04/23 | vie 28/04/23 | 79 |
| 81 | 1.3.2.7.5 | Preparación de botones para control remoto | 2 horas | lun 01/05/23 | lun 01/05/23 | 80 |

Tareas:

- Visualización información completa del robot.
- Visualización cliente asignado.
- Visualización empleado asignado.
- Visualización histórico de incidencias.
- Preparación de botones para control remoto.

5.1.3.3.3 Desarrollo del manejo remoto básico

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F | |
|-----|-----------------|--|----------|--------------|--------------|----|
| 82 | 1.3.3 | Desarrollo del manejo remoto básico | 14 horas | lun 01/05/23 | lun 08/05/23 | |
| 83 | 1.3.3.1 | Análisis y decisión de controles básicos | 2 horas | lun 01/05/23 | mar 02/05/23 | 52 |
| 84 | 1.3.3.2 | Desarrollo de controles básicos en la API | 8 horas | mar 02/05/23 | jue 04/05/23 | |
| 85 | 1.3.3.2.1 | Aparcar | 4 horas | mar 02/05/23 | mié 03/05/23 | 83 |
| 86 | 1.3.3.2.2 | Pausar | 2 horas | mié 03/05/23 | jue 04/05/23 | 85 |
| 87 | 1.3.3.2.3 | Segar | 2 horas | jue 04/05/23 | jue 04/05/23 | 86 |
| 88 | 1.3.3.3 | Añadir controles en la pantalla de información detallada | 4 horas | vie 05/05/23 | lun 08/05/23 | 84 |

Tareas:

- Análisis y decisión de controles básicos.
- Desarrollo de controles básicos en la API.
 - Aparcar.
 - Pausar.
 - Segar.
- Añadir controles en la pantalla de información detallada.

5.1.3.4 Pruebas

La siguiente sección engloba todas las tareas relacionadas con las pruebas, desde pruebas automáticas, hasta pruebas manuales en entornos reales. Se divide en tres partes, pruebas de la API, pruebas de la aplicación móvil y pruebas del sistema completo.

5.1.3.4.1 Pruebas de la API

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|-----|--------------------------------|----------|--------------|--------------|----|
| 89 | 1.4 Pruebas | 76 horas | lun 08/05/23 | lun 12/06/23 | |
| 90 | 1.4.1 Pruebas de la API | 36 horas | lun 08/05/23 | mié 24/05/23 | |
| 91 | 1.4.1.1 Pruebas unitarias | 24 horas | lun 08/05/23 | jue 18/05/23 | 25 |
| 92 | 1.4.1.2 Pruebas de integración | 8 horas | jue 18/05/23 | lun 22/05/23 | 91 |
| 93 | 1.4.1.3 Pruebas de seguridad | 4 horas | mar 23/05/23 | mié 24/05/23 | 92 |

Tareas:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de seguridad.

5.1.3.4.2 Pruebas de la aplicación móvil

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|-----|---|----------|--------------|--------------|----|
| 94 | 1.4.2 Pruebas de la aplicación móvil | 24 horas | mié 24/05/23 | lun 05/06/23 | |
| 95 | 1.4.2.1 Pruebas unitarias | 16 horas | mié 24/05/23 | mié 31/05/23 | 90 |
| 96 | 1.4.2.2 Pruebas de integración | 4 horas | mié 31/05/23 | jue 01/06/23 | 95 |
| 97 | 1.4.2.3 Pruebas de usabilidad y accesibilidad | 4 horas | vie 02/06/23 | lun 05/06/23 | 96 |

Tareas:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de usabilidad.

5.1.3.4.3 Pruebas del sistema completo

| EDT | Nombre de tarea | Duración | Comienzo | Fin | F |
|-----|-------------------------------|----------|--------------|--------------|----|
| 98 | 1.4.3 Pruebas E2E | 8 horas | lun 05/06/23 | mié 07/06/23 | 97 |
| 99 | 1.4.4 Pruebas en entorno real | 8 horas | jue 08/06/23 | lun 12/06/23 | 98 |

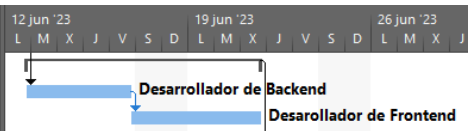
Tareas:

- Pruebas E2E.
- Pruebas en entorno real.

5.1.3.5 Despliegue y distribución

Este apartado incluye las tareas necesarias para el despliegue y distribución del sistema, tanto la API como la aplicación móvil.

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|-----|-------|--------------------------------------|----------|--------------|--------------|-----|
| 100 | 1.5 | Despliegue y distribución | 20 horas | lun 12/06/23 | mié 21/06/23 | |
| 101 | 1.5.1 | Despliegue de la API y base de datos | 12 horas | lun 12/06/23 | vie 16/06/23 | 89 |
| 102 | 1.5.2 | Distribución de la aplicación móvil | 8 horas | vie 16/06/23 | mié 21/06/23 | 101 |



The Gantt chart shows the timeline for tasks 100, 101, and 102. Task 100 (Deployment and distribution) spans from Monday, June 12, 2023, to Wednesday, June 21, 2023. Task 101 (API and database deployment) spans from Monday, June 12, 2023, to Friday, June 16, 2023. Task 102 (Mobile application distribution) spans from Friday, June 16, 2023, to Wednesday, June 21, 2023. The chart also indicates the involvement of 'Desarrollador de Backend' and 'Desarrollador de Frontend'.

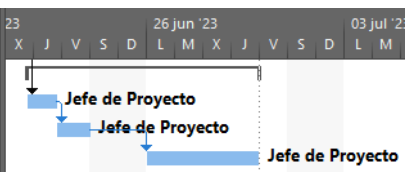
Tareas:

- Despliegue de la API y base de datos.
- Distribución de la aplicación móvil.

5.1.3.6 Documentación del proyecto

La última sección trata de todo el trabajo para la generación de toda la documentación del proyecto.

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | f |
|-----|-------|-----------------------------|----------|--------------|--------------|-----|
| 103 | 1.6 | Documentación del proyecto | 20 horas | mié 21/06/23 | jue 29/06/23 | |
| 104 | 1.6.1 | Documentación de despliegue | 4 horas | mié 21/06/23 | jue 22/06/23 | 100 |
| 105 | 1.6.2 | Manual de instalación | 4 horas | jue 22/06/23 | vie 23/06/23 | 104 |
| 106 | 1.6.3 | Manual de usuario | 12 horas | lun 26/06/23 | jue 29/06/23 | 105 |



The Gantt chart shows the timeline for tasks 103, 104, 105, and 106. Task 103 (Project documentation) spans from Wednesday, June 21, 2023, to Thursday, June 29, 2023. Task 104 (Deployment documentation) spans from Wednesday, June 21, 2023, to Thursday, June 22, 2023. Task 105 (Installation manual) spans from Thursday, June 22, 2023, to Friday, June 23, 2023. Task 106 (User manual) spans from Monday, June 26, 2023, to Thursday, June 29, 2023. The chart also indicates the involvement of 'Jefe de Proyecto'.

Tareas:

- Documentación de despliegue.
- Manual de instalación.
- Manual de usuario.



5.1.3.7 Resumen

A continuación, se muestra un resumen de la planificación explicada en los apartados anteriores.

El proyecto constará de un total de 372 horas de trabajo, que darán comienzo el 9 de enero de 2023 y terminará el 29 de junio de 2023.

RESUMEN DEL WBS

372 horas

| | |
|-----------------------------------|-----------|
| Análisis y Diseño | 66 horas |
| Organización | 24 horas |
| Desarrollo | 190 horas |
| Pruebas | 76 horas |
| Despliegue y distribución | 20 horas |
| Documentación del proyecto | 20 horas |



5.1.4 Riesgos

5.1.4.1 Plan de Gestión de Riesgos

El Plan de Gestión de Riesgos se encuentra especificado en el apartado [10.1 Plan de gestión de riesgos](#).

5.1.4.2 Identificación de Riesgos

Con el fin de prever y reconocer los posibles obstáculos y desafíos que podrían influir en la ejecución y finalización exitosa del proyecto, se llevó a cabo un estudio en el que se identificaron y clasificaron los riesgos más significativos para el sistema a desarrollar, tomando como referencia el plan de gestión de riesgos establecido en el apartado [10.1.1 Valoración de riesgos](#).

5.1.4.2.1 Categoría Técnica

- **Ambigüedad en los requisitos:** Si los requisitos del sistema no están claramente definidos o son cambiantes, aumentará la complejidad o riesgo del proyecto, ya que puede llevar a interpretaciones incorrectas.
- **Inmadurez tecnológica:** Flutter es una tecnología relativamente nueva, esto podría llevar a problemas no anticipados, funcionalidades que no se puedan desarrollar o un rendimiento ineficiente.
- **Brecha de seguridad:** Si la aplicación tiene alguna vulnerabilidad o fallo en el desarrollo, podrían producirse brechas de seguridad que comprometan los datos sensibles de robots, empresas, clientes y empleados.
- **Mala integración de sistemas:** Si los diferentes componentes del sistema, no se integran bien entre ellos, puede que su utilidad se vea limitada, debido a los fallos que pueda causar.

5.1.4.2.2 Categoría Externa

- **Cambio en el software de terceros:** La aplicación depende en gran medida de la API de robots externa. Cualquier cambio o cese de funcionamiento de esta API podría afectar negativamente a la aplicación.
- **Adopción baja:** Si las empresas que puedan usar la aplicación no la adoptan, al no encontrarla atractiva o útil, no se obtendrá el retorno de inversión esperado.
- **Nuevos competidores:** Ahora mismo es una aplicación única en el mercado, pero podrían aparecer nuevas aplicaciones similares, que reduzcan la cuota de mercado de la aplicación.
- **Falta de capacitación:** Si los usuarios finales de las empresas no reciben una capacitación y formación adecuada, no sabrán aprovechar todas las funcionalidades, pudiendo así reducir el uso de la aplicación.
- **Cambios regulatorios:** Si se producen cambios en la normativa, leyes o regulaciones relevantes para el proyecto, podrían obligar a modificar la aplicación para cumplir con los nuevos cambios.

5.1.4.2.3 Categoría de Gestión del Proyecto

- **Estimación inexacta:** Si se subestima el esfuerzo, tiempo, costos o recursos necesarios para desarrollar ciertas funcionalidades, podría causar retrasos y sobrecostos.

5.1.4.3 Registro de Riesgos

El registro de riesgos se encuentra en el apéndice [10.1.2 Gestión de riesgos](#), más concretamente la priorización en el apartado [10.1.2.1 Priorización de Riesgos](#) y la resolución en el apartado [10.1.2.2 Resolución de Riesgos](#).

5.1.5 Presupuesto Inicial

5.1.5.1 Definición de la empresa

En este apartado se mostrará la situación actual de nuestra empresa, definiendo y mostrando los rangos salariales de los empleados, los costes directos e indirectos, la productividad y la facturación, entre otros datos de interés.

5.1.5.1.1 Personal

La empresa es formada por 5 empleados, los cuales están definidos en el apartado [5.1.2.1 OBS](#). A continuación, puede verse su salario anual, así como el coste salarial y el coste total que suponen los empleados.

PERSONAL

| PERSONAL | NUM | SUELDO BRUTO ANUAL | COSTE SALARIAL ANUAL | TOTAL |
|---------------------------|----------|--------------------|----------------------|---------------------|
| Jefe de proyecto | 1 | 50.900,00 € | 67.849,70 € | 67.849,70 € |
| Analista de Sistemas | 1 | 44.100,00 € | 58.785,30 € | 58.785,30 € |
| Diseñador Gráfico | 1 | 25.000,00 € | 33.325,00 € | 33.325,00 € |
| Desarrollador de Frontend | 1 | 28.000,00 € | 37.324,00 € | 37.324,00 € |
| Desarrollador de Backend | 1 | 28.000,00 € | 37.324,00 € | 37.324,00 € |
| Total | 5 | | | 234.608,00 € |

5.1.5.1.2 Productividad del personal

No todos los empleados tienen el mismo grado de productividad o participación en el producto final, por tanto, se calculará el gasto directo e indirecto que supondrá cada tipo de empleado en función de su productividad.

PRODUCTIVIDAD DEL PERSONAL

| Personal | Total | Prod (%) | Coste directo | Ci (%) | Coste Indirecto |
|---------------------------|---------------------|----------|---------------------|--------|--------------------|
| Jefe de proyecto | 67.849,70 € | 30% | 20.354,91 € | 70% | 47.494,79 € |
| Analista de sistemas | 58.785,30 € | 75% | 44.088,98 € | 25% | 14.696,33 € |
| Diseñador gráfico | 33.325,00 € | 85% | 28.326,25 € | 15% | 4.998,75 € |
| Desarrollador de Frontend | 37.324,00 € | 85% | 31.725,40 € | 15% | 5.598,60 € |
| Desarrollador de Backend | 37.324,00 € | 85% | 31.725,40 € | 15% | 5.598,60 € |
| Total | 234.608,00 € | | 156.220,94 € | | 78.387,07 € |



5.1.5.1.3 Costes indirectos

No todos los costes de una empresa vienen representados por los sueldos, hay también otra serie de costes derivados de la propia actividad empresarial.

COSTES INDIRECTOS

| Servicio | Coste Mes | Coste Año |
|---|-------------------|--------------------|
| Asesoría Legal | 350,00 € | 4.200,00 € |
| Seguros (responsabilidad civil, propiedad, etc.) | 400,00 € | 4.800,00 € |
| Limpieza | 150,00 € | 1.800,00 € |
| Luz | 100,00 € | 1.200,00 € |
| Agua | 30,00 € | 360,00 € |
| Calefacción | 50,00 € | 600,00 € |
| Tributos | 200,00 € | 2.400,00 € |
| Material de oficina | 40,00 € | 480,00 € |
| Servicios de seguridad y vigilancia | 200,00 € | 2.400,00 € |
| Asesoría en contabilidad | 300,00 € | 3.600,00 € |
| Capacitación de personal | 150,00 € | 1.800,00 € |
| Proveedores de telefonía e internet | 200,00 € | 2.400,00 € |
| Total | 2.170,00 € | 26.040,00 € |

5.1.5.1.4 Medios de producción

Para que los empleados puedan ejercer las tareas de la empresa, deben de tener a su disposición el material necesario para llevarlas a cabo, ya sea material de oficina, ordenadores, etc. Los costes de estos medios quedan reflejados a continuación.

COSTES DE LOS MEDIOS DE PRODUCCIÓN

| Equipo/Licencia | Uni | Precio | Coste total | Coste Año | Tipo | Plazo |
|---|-----|-------------|-------------|-------------------|--------------|-------|
| Rack de servidor | 1 | 12.000,00 € | 12.000,00 € | 1.500,00 € | Amortización | 8 |
| Equipos de propósito general (Portátiles HP 15s-fq5020ns) | 2 | 549,00 € | 1.098,00 € | 274,50 € | Amortización | 4 |
| Equipos de desarrollo (Portátiles ASUS vivobook S15) | 3 | 998,00 € | 2.994,00 € | 598,80 € | Amortización | 5 |
| Licencias de desarrollo (IntelliJ idea) | 2 | 724,00 € | 1.448,00 € | 1.448,00 € | Alquiler | |
| Monitores 27 pulgadas | 5 | 107,00 € | 535,00 € | 133,75 € | Amortización | 4 |
| Teclado y ratón inalámbrico | 5 | 16,50 € | 82,50 € | 20,63 € | Amortización | 4 |
| iPhone 12 | 2 | 809,00 € | 1.618,00 € | 202,25 € | Amortización | 8 |
| Total | | | | 4.177,93 € | | |



5.1.5.1.5 Beneficios

Teniendo en cuenta los datos de los 4 apartados anteriores, y que esperamos obtener un 22% de beneficio anual, podemos mostrar los objetivos y necesidades de la empresa, visibles en la tabla siguiente.

| | |
|-------------------------------------|--------------|
| Costes directos e indirectos | 264.825,93 € |
| Beneficios deseados (22%) | 58.261,70 € |
| Necesidades de facturación | 323.087,63 € |

5.1.5.1.6 Horas productivas y no productivas

Considerando los porcentajes de productividad vistos en el apartado Productividad del personal, y estimando 2008 horas laborables al año, se calculan las horas productivas de los empleados.

HORAS PRODUCTIVAS Y NO PRODUCTIVAS

| Personal | Product. (%) | Horas/Año | Horas productivas / año (por persona) | Horas productivas (total empresa) |
|---------------------------|--------------|-----------|--|---|
| Jefe de proyecto | 30% | 2008 | 602,4 | 602,4 |
| Analista de sistemas | 75% | 2008 | 1506 | 1506 |
| Diseñador Gráfico | 85% | 2008 | 1706,8 | 1706,8 |
| Desarrollador de Frontend | 85% | 2008 | 1706,8 | 1706,8 |
| Desarrollador de Backend | 85% | 2008 | 1706,8 | 1706,8 |
| Total | | | | 7228,8 |

5.1.5.1.7 Precio por hora

Para calcular el precio por hora de cada empleado se tiene en cuenta, no solo su salario anual, pero también los costes indirectos.

Los precios por hora, sin esperar beneficios, de los empleados son los siguientes:

PRECIO / HORA (SIN BENEFICIOS)

| | |
|----------------------------------|--------|
| Jefe de proyecto | 60,45€ |
| Analista de sistemas | 39,60€ |
| Diseñador gráfico | 22,10€ |
| Desarrollador de frontend | 24,39€ |
| Desarrollador de backend | 24,39€ |

Por otro lado, se aumentan los precios/hora en un 57% para alcanzar la facturación necesaria para cumplir los objetivos de beneficios:

PRECIO HORA (COSTE Y VENTA)

| Personal | Precio/hora | Horas productivas (Total empresa) | Facturación |
|----------------------------------|--------------------|--|---------------------|
| Jefe de proyecto | 94,91 € | 602,4 | 57.173,78 € |
| Analista de sistemas | 62,18 € | 1506 | 93.640,93 € |
| Diseñador Gráfico | 34,69 € | 1706,8 | 59.208,21 € |
| Desarrollador de Frontend | 38,29 € | 1706,8 | 65.345,37 € |
| Desarrollador de Backend | 38,29 € | 1706,8 | 65.345,37 € |
| Total | | | 340.713,67 € |



5.1.5.1.8 Resumen

Todo lo referente a la definición de la empresa queda reflejado y resumido en la siguiente tabla:

RESUMEN

| N.º | Concepto | Importe |
|-----|--|--------------|
| 1 | Todos los costes directos | 156.220,94 € |
| 2 | Total de los costes indirectos | 108.604,99 € |
| 3 | Suma de los costes indirectos e indirectos | 264.825,93 € |
| 4 | Beneficios deseados | 58.261,70 € |
| 5 | Coste total (suma directos, indirectos y beneficios) | 323.087,63 € |
| 6 | Facturación posible en función de las horas de producción y de los precios por hora calculados | 340.713,67 € |
| 7 | Margen entre el coste total y la facturación (entre 5 y 6) | 5,17% |

Como podemos ver en la fila 7, habría un margen del 5.17% entre coste total y la facturación prevista. Este margen se mantendrá en todos los proyectos y funcionará como reserva o contingencia frente a imprevistos presupuestarios.

5.1.5.2 Presupuesto de Costes

En esta sección se detallarán las partidas de costes identificadas en el proyecto. Las partidas que se describen a continuación son:

1. Análisis y Diseño
2. Organización
3. Desarrollo
4. Pruebas
5. Despliegue y Distribución
6. Documentación

5.1.5.2.1 Partida 1: Análisis y Diseño

ANALISIS Y DISEÑO

| Descripción | Cantidad | Unidades | Precio | Sub (3) | Sub (2) | Total |
|--------------------------------------|----------|----------|---------|----------|----------|-----------------|
| Análisis del Proyecto | | | | | | 992,20 € |
| Obtención y análisis de stakeholders | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Obtención inicial de requisitos | | | | | 316,83 € | |
| <i>Analista de sistemas</i> | 8 | horas | 39,60 € | 316,83 € | | |
| Validación de Requisitos | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Creación del ERS | | | | | 316,83 € | |
| <i>Analista de sistemas</i> | 8 | horas | 39,60 € | 316,83 € | | |
| Validación del ERS | | | | | 200,11 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |



| | | | | | | |
|--|----|-------|---------|----------|----------|-----------------|
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| Diseño del Proyecto | | | | | | 607,00 € |
| Prototipos de pantallas iniciales | | | | | 132,57 € | |
| <i>Diseñador gráfico</i> | 6 | horas | 22,10 € | 132,57 € | | |
| Validación y refinamiento de los prototipos | | | | | 44,19 € | |
| <i>Diseñador gráfico</i> | 2 | horas | 22,10 € | 44,19 € | | |
| Wireframe de la aplicación | | | | | 265,14 € | |
| <i>Diseñador gráfico</i> | 12 | horas | 22,10 € | 265,14 € | | |
| Validación del wireframe | | | | | 165,10 € | |
| <i>Diseñador gráfico</i> | 2 | horas | 22,10 € | 44,19 € | | |
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| Análisis y Diseño de la infraestructura | | | | | | 912,99 € |
| Análisis de sistemas similares | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Análisis de alternativas de infraestructura | | | | | 158,42 € | |
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |
| Esquema inicial de la arquitectura del sistema | | | | | 158,42 € | |
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |



| | | | | | | |
|---------------------------------------|---|-------|---------|----------|----------|-------------------|
| Análisis de alternativas tecnológicas | | | | | 158,42 € | |
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |
| Diseño de la solución final | | | | | 358,53 € | |
| <i>Analista de sistemas</i> | 6 | horas | 39,60 € | 237,63 € | | |
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| TOTAL DE ANALISIS Y DISEÑO: | | | | | | 2.512,19 € |

5.1.5.2.2 Partida 2: Organización

ORGANIZACIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|---|----------|----------|---------|----------|-----------------|
| Reunión recurrente con el tutor | | | | | 483,62 € |
| <i>Jefe de Proyecto</i> | 8 | horas | 60,45 € | 483,62 € | |
| <i>Reunión recurrente con stakeholders externos</i> | | | | | 483,62 € |
| <i>Jefe de Proyecto</i> | 8 | horas | 60,45 € | 483,62 € | |
| TOTAL DE ORGANIZACIÓN | | | | | 967,24 € |



5.1.5.2.3 Partida 3: Desarrollo

DESARROLLO

| Descripción | Cant. | Unidades | Precio | Sub (3) | Sub (2) | Total |
|---|-------|----------|---------|----------|----------|-------------------|
| Desarrollo de la API | | | | | | 2.584,87 € |
| Diseño inicial del modelo de base de datos | | | | | 292,63 € | |
| <i>Desarrollador de Backend</i> | 6 | horas | 24,39 € | 146,31 € | | |
| <i>Desarrollador de Frontend</i> | 6 | horas | 24,39 € | 146,31 € | | |
| Diseño inicial de la especificación de la API | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Gestión de empleados | | | | | 268,24 € | |
| <i>Desarrollador de Backend</i> | 11 | horas | 24,39 € | 268,24 € | | |
| Gestión de clientes | | | | | 268,24 € | |
| <i>Desarrollador de Backend</i> | 11 | horas | 24,39 € | 268,24 € | | |
| Gestión de usuarios | | | | | 341,40 € | |
| <i>Desarrollador de Backend</i> | 14 | horas | 24,39 € | 341,40 € | | |



| | | | | | | |
|--|----|-------|---------|----------|----------|-------------------|
| Gestión de robots | | | | | 487,71 € | |
| <i>Desarrollador de Backend</i> | 20 | horas | 24,39 € | 487,71 € | | |
| Gestión de incidencias | | | | | 536,48 € | |
| <i>Desarrollador de Backend</i> | 22 | horas | 24,39 € | 536,48 € | | |
| Gestión de roles | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | 195,08 € | | |
| Desarrollo de la Aplicación Móvil | | | | | | 1.950,84 € |
| Gestión de robots | | | | | 292,63 € | |
| <i>Desarrollador de Frontend</i> | 12 | horas | 24,39 € | 292,63 € | | |
| Gestión de clientes | | | | | 292,63 € | |
| <i>Desarrollador de Frontend</i> | 12 | horas | 24,39 € | 292,63 € | | |
| Gestión de empleados | | | | | 292,63 € | |
| <i>Desarrollador de Frontend</i> | 12 | horas | 24,39 € | 292,63 € | | |
| Asignación de robots | | | | | 243,86 € | |
| <i>Desarrollador de Frontend</i> | 10 | horas | 24,39 € | 243,86 € | | |
| Gestión de incidencias | | | | | 146,31 € | |



| | | | | | | |
|---|----|-------|---------|----------|----------|-------------------|
| <i>Desarrollador de Frontend</i> | 6 | horas | 24,39 € | 146,31 € | | |
| Gestión de sesión | | | | | 341,40 € | |
| <i>Desarrollador de Frontend</i> | 14 | horas | 24,39 € | 341,40 € | | |
| Información detallada de robots | | | | | 341,40 € | |
| <i>Desarrollador de Frontend</i> | 14 | horas | 24,39 € | 341,40 € | | |
| Desarrollo del manejo remoto básico | | | | | | 371,83 € |
| Análisis y decisión de controles básicos | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Desarrollo de controles básicos en la API | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | 195,08 € | | |
| Controles en la pantalla de información detallada | | | | | 97,54 € | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| TOTAL DE DESARROLLO: | | | | | | 4.907,55 € |

5.1.5.2.4 Partida 4: Pruebas

PRUEBAS

| Descripción | Cant. | Unidades | Precio | Sub (3) | Sub (2) | Total |
|---------------------------------------|-------|----------|---------|----------|----------|-----------------|
| Pruebas de la API | | | | | | 877,88 € |
| Pruebas unitarias | | | | | 585,25 € | |
| <i>Desarrollador de Backend</i> | 24 | horas | 24,39 € | 585,25 € | | |
| Pruebas de integración | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | 195,08 € | | |
| Pruebas de seguridad | | | | | 97,54 € | |
| <i>Desarrollador de Backend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Pruebas de la Aplicación Móvil | | | | | | 585,25 € |
| Pruebas unitarias | | | | | 390,17 € | |
| <i>Desarrollador de Frontend</i> | 16 | horas | 24,39 € | 390,17 € | | |
| Pruebas de integración | | | | | 97,54 € | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Pruebas de usabilidad y accesibilidad | | | | | 97,54 € | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |



| | | | | | | |
|----------------------------------|---|-------|---------|--|----------|-------------------|
| Pruebas E2E | | | | | | 390,17 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| Pruebas en entorno real | | | | | | 631,98 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Jefe de Proyecto</i> | 4 | horas | 60,45 € | | 241,81 € | |
| TOTAL DE PRUEBAS: | | | | | | 2.485,28 € |

5.1.5.2.5 Partida 5: Despliegue y Distribución

DESPLIEGUE Y DISTRIBUCIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|--|----------|----------|---------|----------|-----------------|
| Despliegue de la API y base de datos | | | | | 292,63 € |
| <i>Desarrollador de Backend</i> | 12 | horas | 24,39 € | 292,63 € | |
| Distribución de la aplicación móvil | | | | | 195,08 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | 195,08 € | |
| TOTAL DE DESPLIEGUE Y DISTRIBUCIÓN: | | | | | 487,71 € |

5.1.5.2.6 Partida 6: Documentación

DOCUMENTACIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|--------------------------------|----------|----------|---------|----------|-----------------|
| Documentación de despliegue | | | | | 241,81 € |
| <i>Jefe de Proyecto</i> | 4 | horas | 60,45 € | 241,81 € | |
| Manual de instalación | | | | | 241,81 € |
| <i>Jefe de Proyecto</i> | 4 | horas | 60,45 € | 241,81 € | |
| Manual de usuario | | | | | 483,62 € |
| <i>Jefe de Proyecto</i> | 8 | horas | 60,45 € | 483,62 € | |
| TOTAL DE DOCUMENTACIÓN: | | | | | 967,24 € |

5.1.5.2.7 Resumen y ponderación de presupuesto de costes

PRESUPUESTO DE COSTES RESUMIDO

| Cod. | Partida | Total |
|------|-------------------|--------------------|
| 01 | Análisis y Diseño | 2.512,19 € |
| 02 | Organización | 967,24 € |
| 03 | Desarrollo | 4.907,55 € |
| 04 | Pruebas | 2.485,28 € |
| 05 | Despliegue | 487,71 € |
| 06 | Documentación | 967,24 € |
| | TOTAL | 11.359,96 € |

Si se aplica el 22% de beneficio que se espera obtener, el coste total y final del proyecto es de **13.859,15 €**. En esta cantidad, se mantendría el margen de 5,17% especificado en el apartado [5.1.5.1.8 Resumen](#), lo que vendría a suponer una cantidad de 787,70 € como contingencia frente a gastos imprevistos o retrasos.

Ahora, se debe redistribuir los beneficios esperados (2.499,19 €) y la partida de ‘Organización’ (967,24 €) en el resto de las tareas del proyecto para que no queden reflejados en el presupuesto final de cliente. Con este fin se calcula un porcentaje de ponderación a aplicar al resto de tareas, el cual resulta ser del 33%.

Con este porcentaje de ponderación, se aumenta el precio a facturar de todas las partidas. Este proceso puede verse a continuación, donde hay un aumento del 33% entre la columna “Coste” y “Facturación a cliente”.

PONDERACIÓN DE LOS COSTES PARA CLIENTE

| Partida | Partida | Coste | A facturar al cliente |
|-----------------------|---------------------------|------------|-----------------------|
| 1 | Análisis y Diseño | 2.512,19 € | 3.350,11 € |
| 3 | Desarrollo | 4.907,55 € | 6.544,43 € |
| 4 | Pruebas | 2.485,28 € | 3.314,23 € |
| 5 | Despliegue y Distribución | 487,71 € | 650,38 € |
| 6 | Documentación | 967,24 € | 1.289,85 € |
| TOTAL CLIENTE: | | | 15.149,00 € |



5.1.5.3 Presupuesto de Cliente

Teniendo en cuenta todo lo expuesto en el apartado [5.1.5.2 Presupuesto de costes](#), se presenta aquí el presupuesto final entregable al cliente, visible en la tabla a continuación.

PRESUPUESTO DE CLIENTE

| Cod | Partida | Total |
|----------------------|---------------------------|--------------------|
| 1 | Análisis y Diseño | 3.350,11 € |
| 2 | Desarrollo | 6.544,43 € |
| 3 | Pruebas | 3.314,23 € |
| 4 | Despliegue y Distribución | 650,38 € |
| 5 | Documentación | 1.289,85 € |
| TOTAL CLIENTE | | 15.149,00 € |



5.2 EJECUCIÓN DEL PROYECTO

5.2.1 Bitácora de Incidencias del Proyecto

El proyecto arranca según lo esperado, siguiendo la planificación inicial. La parte de Análisis y Diseño se completó sin incidencias y en los tiempos acordados, así como el desarrollo de la API.

El primer problema apareció cuando se empezó con el desarrollo de la aplicación móvil, debido a la falta de conocimiento en la tecnología Flutter, se decidió retrasar el inicio del desarrollo de la aplicación para formarse en dicha tecnología antes de empezar con el desarrollo. Una vez empezado el desarrollo, debido también a la falta de experiencia con Flutter, el trabajo acordado para casi todas las tareas de desarrollo aumentó en algunas horas. No hubo ninguna gran incidencia más, y el desarrollo de todo el sistema se completó satisfactoriamente.

La fase de pruebas se tuvo que retrasar, ya que no se disponían de los recursos necesarios y se tardó en acordar una fecha con la empresa, para realizar estas pruebas en entornos reales.

Por último, llegada la fase de documentación, se cambiaron los diferentes documentos que se iban a realizar, ya que no tenía sentido hacer un manual de instalación y otro de despliegue por separado. Además se añadió un manual de programador, necesario para que en caso de que futuros programadores ajenos al desarrollo inicial, sean capaces de contribuir al sistema.

5.3 CIERRE DEL PROYECTO

5.3.1 Planificación Final

En este apartado, se muestra la planificación final del proyecto, la cual inicia el 9 de enero de 2023 y termina el 5 de junio de 2024, debido a los retrasos ya comentados en el apartado anterior.

| | EDT | Nombre de tarea | Duración | Comienzo | Fin | Predeces | Nombres de los recursos |
|----|----------------|--|-------------------|---------------------|---------------------|----------|---|
| 1 | 1 | ▸ TFG - Mower Control APP | 1103 horas | lun 09/01/23 | mié 05/06/24 | | |
| 2 | 1.1 | ▸ Análisis y Diseño | 66 horas | lun 09/01/23 | mar 07/02/23 | | |
| 3 | 1.1.1 | ▸ Análisis del proyecto | 24 horas | lun 09/01/23 | mié 18/01/23 | | |
| 9 | 1.1.2 | ▸ Diseño del proyecto | 22 horas | jue 19/01/23 | lun 30/01/23 | | |
| 10 | 1.1.2.1 | Prototipos de pantalla iniciales | 6 horas | jue 19/01/23 | vie 20/01/23 | 3 | Diseñador Gráfico |
| 11 | 1.1.2.2 | Validación y refinamiento de los prototipos | 2 horas | lun 23/01/23 | lun 23/01/23 | 10 | Diseñador Gráfico |
| 12 | 1.1.2.3 | Wireframe de la aplicación | 12 horas | lun 23/01/23 | vie 27/01/23 | 11 | Diseñador Gráfico |
| 13 | 1.1.2.4 | Validación del wireframe | 2 horas | vie 27/01/23 | lun 30/01/23 | 12 | Diseñador Gráfico; Jefe de Proyecto |
| 14 | 1.1.3 | ▸ Análisis y Diseño de la Infraestructura | 20 horas | lun 30/01/23 | mar 07/02/23 | | |
| 15 | 1.1.3.1 | Análisis de sistemas similares | 2 horas | lun 30/01/23 | lun 30/01/23 | 9 | Analista de sistemas |
| 16 | 1.1.3.2 | Análisis de alternativas de infraestructura | 4 horas | mar 31/01/23 | mié 01/02/23 | 15 | Analista de sistemas |
| 17 | 1.1.3.3 | Esquema inicial de la arquitectura del sistema | 4 horas | mié 01/02/23 | jue 02/02/23 | 16 | Analista de sistemas |
| 18 | 1.1.3.4 | Análisis de alternativas tecnológicas | 4 horas | jue 02/02/23 | vie 03/02/23 | 17 | Analista de sistemas |
| 19 | 1.1.3.5 | ▸ Diseño de la solución final | 6 horas | lun 06/02/23 | mar 07/02/23 | | |
| 20 | 1.1.3.5.1 | Selección de alternativas tecnológicas | 2 horas | lun 06/02/23 | lun 06/02/23 | 18 | Analista de sistemas; Jefe de Proyecto |
| 21 | 1.1.3.5.2 | Esquema final de la arquitectura del sistema | 4 horas | lun 06/02/23 | mar 07/02/23 | 20 | Analista de sistemas |
| 22 | 1.2 | ▸ Organización | 372 horas | lun 09/01/23 | jue 29/06/23 | | |
| 23 | 1.2.1 | Reunión recurrente con tutor del TFG | 16 horas | lun 09/01/23 | lun 16/01/23 | | Jefe de Proyecto |
| 24 | 1.2.2 | Reunión recurrente con stakeholders externos | 8 horas | mar 27/06/23 | jue 29/06/23 | | Jefe de Proyecto |
| 25 | 1.3 | ▸ Desarrollo | 852 horas | mié 08/02/23 | lun 11/03/24 | | |
| 26 | 1.3.1 | ▸ Desarrollo de la API | 96 horas | mié 08/02/23 | jue 23/03/23 | | |
| 27 | 1.3.1.1 | Diseño inicial del modelo de base de datos | 6 horas | mié 08/02/23 | jue 09/02/23 | 2 | Desarrollador de Backend; Desarrollador de Frontend |
| 28 | 1.3.1.2 | Diseño inicial de la especificación de la API | 4 horas | vie 10/02/23 | lun 13/02/23 | 27 | Desarrollador de Backend; Desarrollador de Frontend |
| 29 | 1.3.1.3 | ▸ Gestión de empleados | 11 horas | lun 13/02/23 | jue 16/02/23 | | |
| 30 | 1.3.1.3.1 | Creación de empleados | 6 horas | lun 13/02/23 | mié 15/02/23 | 28 | Desarrollador de Backend |
| 31 | 1.3.1.3.2 | Consulta de empleados | 2 horas | mié 15/02/23 | mié 15/02/23 | 30 | Desarrollador de Backend |
| 32 | 1.3.1.3.3 | Borrado de empleados | 3 horas | jue 16/02/23 | jue 16/02/23 | 31 | Desarrollador de Backend |
| 33 | 1.3.1.4 | ▸ Gestión de clientes | 11 horas | vie 17/02/23 | mié 22/02/23 | | |
| 34 | 1.3.1.4.1 | Creación de clientes | 6 horas | vie 17/02/23 | lun 20/02/23 | 29 | Desarrollador de Backend |
| 35 | 1.3.1.4.2 | Consulta de clientes | 2 horas | mar 21/02/23 | mar 21/02/23 | 34 | Desarrollador de Backend |
| 36 | 1.3.1.4.3 | Borrado de clientes | 3 horas | mar 21/02/23 | mié 22/02/23 | 35 | Desarrollador de Backend |
| 37 | 1.3.1.5 | ▸ Gestión de usuarios | 14 horas | mié 22/02/23 | mié 01/03/23 | | |
| 38 | 1.3.1.5.1 | Registro | 8 horas | mié 22/02/23 | lun 27/02/23 | 33 | Desarrollador de Backend |
| 39 | 1.3.1.5.2 | Login | 4 horas | lun 27/02/23 | mar 28/02/23 | 38 | Desarrollador de Backend |
| 40 | 1.3.1.5.3 | Generación automática de credenciales para empleados | 2 horas | mar 28/02/23 | mié 01/03/23 | 39 | Desarrollador de Backend |



| EDT | Nombre de tarea | Duración | Comienzo | Fin | Predeces | Nombres de los recursos |
|-----|--|------------------|---------------------|---------------------|----------|---------------------------|
| 41 | 1.3.1.6 ▲ Gestión de robots | 20 horas | mié 01/03/23 | jue 09/03/23 | | |
| 42 | 1.3.1.6.1 Obtención y guardado de robots desde un proveedor externo | 12 horas | mié 01/03/23 | mar 07/03/23 | 37 | Desarrollador de Backend |
| 43 | 1.3.1.6.2 Consulta de robots | 4 horas | mar 07/03/23 | mié 08/03/23 | 42 | Desarrollador de Backend |
| 44 | 1.3.1.6.3 Asignación de robots | 4 horas | mié 08/03/23 | jue 09/03/23 | 43 | Desarrollador de Backend |
| 45 | 1.3.1.7 ▲ Gestión de incidencias | 22 horas | vie 10/03/23 | mar 21/03/23 | | |
| 46 | 1.3.1.7.1 Generación de incidencias | 4 horas | vie 10/03/23 | lun 13/03/23 | 41 | Desarrollador de Backend |
| 47 | 1.3.1.7.2 Consulta de incidencias | 2 horas | lun 13/03/23 | lun 13/03/23 | 46 | Desarrollador de Backend |
| 48 | 1.3.1.7.3 Edición de incidencias | 2 horas | mar 14/03/23 | mar 14/03/23 | 47 | Desarrollador de Backend |
| 49 | 1.3.1.7.4 Actualización periódica del estado de los robots | 2 horas | mar 14/03/23 | mié 15/03/23 | 48 | Desarrollador de Backend |
| 50 | 1.3.1.7.5 Envío de notificaciones de incidencias | 12 horas | mié 15/03/23 | mar 21/03/23 | 49 | Desarrollador de Backend |
| 51 | 1.3.1.8 Gestión de roles | 8 horas | mar 21/03/23 | jue 23/03/23 | 50 | Desarrollador de Backend |
| 52 | 1.3.2 ▲ Desarrollo de la aplicación móvil | 174 horas | mié 01/11/23 | vie 19/01/24 | | |
| 53 | 1.3.2.1 ▲ Gestión de robots | 20 horas | mié 01/11/23 | jue 09/11/23 | | |
| 54 | 1.3.2.1.1 Visualización información de los robots | 12 horas | mié 01/11/23 | lun 06/11/23 | 26 | Desarrollador de Frontend |
| 55 | 1.3.2.1.2 Visualización estado de los robots | 8 horas | mar 07/11/23 | jue 09/11/23 | 54 | Desarrollador de Frontend |
| 56 | 1.3.2.2 ▲ Gestión de clientes | 16 horas | jue 09/11/23 | jue 16/11/23 | | |
| 57 | 1.3.2.2.1 Visualización de clientes | 4 horas | jue 09/11/23 | vie 10/11/23 | 53 | Desarrollador de Frontend |
| 58 | 1.3.2.2.2 Creación de cliente | 8 horas | lun 13/11/23 | mié 15/11/23 | 57 | Desarrollador de Frontend |
| 59 | 1.3.2.2.3 Borrado de cliente | 4 horas | mié 15/11/23 | jue 16/11/23 | 58 | Desarrollador de Frontend |
| 60 | 1.3.2.3 ▲ Gestión de empleados | 12 horas | vie 17/11/23 | mié 22/11/23 | | |
| 61 | 1.3.2.3.1 Visualización de empleados | 4 horas | vie 17/11/23 | lun 20/11/23 | 56 | Desarrollador de Frontend |
| 62 | 1.3.2.3.2 Creación de empleado | 6 horas | lun 20/11/23 | mié 22/11/23 | 61 | Desarrollador de Frontend |
| 63 | 1.3.2.3.3 Borrado de empleado | 2 horas | mié 22/11/23 | mié 22/11/23 | 62 | Desarrollador de Frontend |
| 64 | 1.3.2.4 ▲ Asignación de robots | 14 horas | jue 23/11/23 | mié 29/11/23 | | |
| 65 | 1.3.2.4.1 Visualización de robots sin asignar | 4 horas | jue 23/11/23 | vie 24/11/23 | 60 | Desarrollador de Frontend |
| 66 | 1.3.2.4.2 Asignación de robot | 10 horas | vie 24/11/23 | mié 29/11/23 | 65 | Desarrollador de Frontend |
| 67 | 1.3.2.5 ▲ Gestión de incidencias | 10 horas | mié 29/11/23 | lun 04/12/23 | | |
| 68 | 1.3.2.5.1 Visualización de incidencias | 6 horas | mié 29/11/23 | vie 01/12/23 | 64 | Desarrollador de Frontend |
| 69 | 1.3.2.5.2 Edición de incidencias | 4 horas | vie 01/12/23 | lun 04/12/23 | 68 | Desarrollador de Frontend |
| 70 | 1.3.2.6 ▲ Gestión de sesión | 24 horas | mar 05/12/23 | jue 14/12/23 | | |
| 71 | 1.3.2.6.1 Login | 8 horas | mar 05/12/23 | jue 07/12/23 | 67 | Desarrollador de Frontend |
| 72 | 1.3.2.6.2 ▲ Registro | 14 horas | jue 07/12/23 | jue 14/12/23 | | |
| 73 | 1.3.2.6.2 Registro básico | 6 horas | jue 07/12/23 | lun 11/12/23 | 71 | Desarrollador de Frontend |
| 74 | 1.3.2.6.2 Registro contra proveedor externo | 8 horas | lun 11/12/23 | jue 14/12/23 | 73 | Desarrollador de Frontend |
| 75 | 1.3.2.6.3 Cierre de sesión | 2 horas | jue 14/12/23 | jue 14/12/23 | 72 | Desarrollador de Frontend |



| EDT | Nombre de tarea | Duración | Comienzo | Fin | Predeces | Nombres de los recursos | |
|-----|-----------------|--|-----------------|---------------------|---------------------|-------------------------|--------------------------------------|
| 76 | 1.3.2.7 | Información detallada de robots | 14 horas | lun 08/01/24 | vie 12/01/24 | | |
| 77 | 1.3.2.7.1 | Visualización información completa del robot | 4 horas | lun 08/01/24 | mar 09/01/24 | 70 | Desarrollador de Frontend |
| 78 | 1.3.2.7.2 | Visualización cliente asignado | 2 horas | mar 09/01/24 | mar 09/01/24 | 77 | Desarrollador de Frontend |
| 79 | 1.3.2.7.3 | Visualización empleado asignado | 2 horas | mié 10/01/24 | mié 10/01/24 | 78 | Desarrollador de Frontend |
| 80 | 1.3.2.7.4 | Visualización historico de incidencias | 4 horas | mié 10/01/24 | jue 11/01/24 | 79 | Desarrollador de Frontend |
| 81 | 1.3.2.7.5 | Preparación de botones para control remoto | 2 horas | vie 12/01/24 | vie 12/01/24 | 80 | Desarrollador de Frontend |
| 82 | 1.3.3 | Desarrollo del manejo remoto básico | 21 horas | vie 01/03/24 | lun 11/03/24 | | |
| 83 | 1.3.3.1 | Análisis y decisión de controles básicos | 2 horas | vie 01/03/24 | vie 01/03/24 | 52 | Analista de sistemas |
| 84 | 1.3.3.2 | Desarrollo de controles básicos en la API | 14 horas | vie 01/03/24 | vie 08/03/24 | | |
| 85 | 1.3.3.2.1 | Aparcar | 6 horas | vie 01/03/24 | mar 05/03/24 | 83 | Desarrollador de Backend |
| 86 | 1.3.3.2.2 | Pausar | 4 horas | mar 05/03/24 | mié 06/03/24 | 85 | Desarrollador de Backend |
| 87 | 1.3.3.2.3 | Segar | 4 horas | jue 07/03/24 | vie 08/03/24 | 86 | Desarrollador de Backend |
| 88 | 1.3.3.3 | Añadir controles en la pantalla de información detallada | 4 horas | vie 08/03/24 | lun 11/03/24 | 84 | Desarrollador de Frontend |
| 89 | 1.4 | Pruebas | 78 horas | lun 08/04/24 | lun 13/05/24 | | |
| 90 | 1.4.1 | Pruebas de la API | 36 horas | lun 08/04/24 | mar 23/04/24 | | |
| 91 | 1.4.1.1 | Pruebas unitarias | 24 horas | lun 08/04/24 | mié 17/04/24 | 25 | Desarrollador de Backend |
| 92 | 1.4.1.2 | Pruebas de integración | 8 horas | jue 18/04/24 | lun 22/04/24 | 91 | Desarrollador de Backend |
| 93 | 1.4.1.3 | Pruebas de seguridad | 4 horas | lun 22/04/24 | mar 23/04/24 | 92 | Desarrollador de Backend |
| 94 | 1.4.2 | Pruebas de la aplicación móvil | 26 horas | mié 24/04/24 | lun 06/05/24 | | |
| 95 | 1.4.2.1 | Pruebas unitarias | 16 horas | mié 24/04/24 | mié 01/05/24 | 90 | Desarrollador de Frontend |
| 96 | 1.4.2.2 | Pruebas de integración | 4 horas | mié 01/05/24 | jue 02/05/24 | 95 | Desarrollador de Frontend |
| 97 | 1.4.2.3 | Pruebas de usabilidad | 6 horas | jue 02/05/24 | lun 06/05/24 | 96 | Desarrollador de Frontend |
| 98 | 1.4.3 | Pruebas E2E | 8 horas | lun 06/05/24 | jue 09/05/24 | 97 | Desarrollador de Frontend;Desarrolla |
| 99 | 1.4.4 | Pruebas en entorno real | 8 horas | jue 09/05/24 | lun 13/05/24 | 98 | Desarrollador de Frontend;Desarrolla |
| 100 | 1.5 | Despliegue y distribución | 20 horas | lun 20/05/24 | mar 28/05/24 | | |
| 101 | 1.5.1 | Despliegue de la API y base de datos | 12 horas | lun 20/05/24 | jue 23/05/24 | 89 | Desarrollador de Backend |
| 102 | 1.5.2 | Distribución de la aplicación móvil | 8 horas | vie 24/05/24 | mar 28/05/24 | 101 | Desarrollador de Frontend |
| 103 | 1.6 | Documentación del proyecto | 18 horas | mar 28/05/24 | mié 05/06/24 | | |
| 104 | 1.6.1 | Manual de programador | 2 horas | mar 28/05/24 | mié 29/05/24 | 100 | Jefe de Proyecto |
| 105 | 1.6.2 | Manual de instalación y despliegue | 4 horas | mié 29/05/24 | jue 30/05/24 | 104 | Jefe de Proyecto |
| 106 | 1.6.3 | Manual de usuario | 12 horas | jue 30/05/24 | mié 05/06/24 | 105 | Jefe de Proyecto |

5.3.1.1 Cambios realizados

| Problema | Cambio realizado |
|--|--|
| Falta de conocimiento en Flutter y falta de tiempo para el desarrollo. | Se retrasa el inicio del desarrollo de la aplicación móvil. |
| Falta de experiencia en Flutter | Se aumenta el trabajo necesario para casi todas las tareas de desarrollo de la aplicación móvil. |
| Falta de recursos para hacer las pruebas en entornos reales | Se retrasa el inicio del bloque de pruebas. |
| Documentación errónea | Se ajustan las tareas de documentación que se van a realizar. |



5.3.1.2 Resumen

El proyecto se completó en 17 meses aproximadamente, de principios de enero de 2023 a principios de junio de 2024. El proyecto conllevó un total de 432 horas repartidas a lo largo de ese tiempo, ya que hubo periodos de inactividad donde no había trabajo efectivo en ninguna de las tareas.

RESUMEN DEL WBS

432 horas

| | |
|-----------------------------------|-----------|
| Análisis y Diseño | 66 horas |
| Organización | 24 horas |
| Desarrollo | 226 horas |
| Pruebas | 78 horas |
| Despliegue y distribución | 20 horas |
| Documentación del proyecto | 18 horas |

5.3.2 Presupuesto Final de Costes

A continuación, se detalla el presupuesto final de costes basándose en la planificación final. Primero, se mostrarán las partidas de manera detallada y al final un resumen del presupuesto de costes.

5.3.2.1 Partidas

5.3.2.1.1 Partida 1: Análisis y Diseño

ANÁLISIS Y DISEÑO

| Descripción | Cantidad | Unidades | Precio | Sub (3) | Sub (2) | Total |
|--------------------------------------|----------|----------|---------|----------|----------|-----------------|
| Análisis del Proyecto | | | | | | 992,20 € |
| Obtención y análisis de stakeholders | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Obtención inicial de requisitos | | | | | 316,83 € | |
| <i>Analista de sistemas</i> | 8 | horas | 39,60 € | 316,83 € | | |
| Validación de Requisitos | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Creación del ERS | | | | | 316,83 € | |
| <i>Analista de sistemas</i> | 8 | horas | 39,60 € | 316,83 € | | |
| Validación del ERS | | | | | 200,11 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| Diseño del Proyecto | | | | | | 607,00 € |



| | | | | | | |
|--|----|-------|---------|----------|----------|-----------------|
| Prototipos de pantallas iniciales | | | | | 132,57 € | |
| <i>Diseñador gráfico</i> | 6 | horas | 22,10 € | 132,57 € | | |
| Validación y refinamiento de los prototipos | | | | | 44,19 € | |
| <i>Diseñador gráfico</i> | 2 | horas | 22,10 € | 44,19 € | | |
| Wireframe de la aplicación | | | | | 265,14 € | |
| <i>Diseñador gráfico</i> | 12 | horas | 22,10 € | 265,14 € | | |
| Validación del wireframe | | | | | 165,10 € | |
| <i>Diseñador gráfico</i> | 2 | horas | 22,10 € | 44,19 € | | |
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| Análisis y Diseño de la infraestructura | | | | | | 912,99 € |
| Análisis de sistemas similares | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Análisis de alternativas de infraestructura | | | | | 158,42 € | |
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |
| Esquema inicial de la arquitectura del sistema | | | | | 158,42 € | |
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |
| Análisis de alternativas tecnológicas | | | | | 158,42 € | |



| | | | | | | |
|------------------------------------|---|-------|---------|----------|----------|-------------------|
| <i>Analista de sistemas</i> | 4 | horas | 39,60 € | 158,42 € | | |
| Diseño de la solución final | | | | | 358,53 € | |
| <i>Analista de sistemas</i> | 6 | horas | 39,60 € | 237,63 € | | |
| <i>Jefe de proyecto</i> | 2 | horas | 60,45 € | 120,90 € | | |
| TOTAL DE ANALISIS Y DISEÑO: | | | | | | 2.512,19 € |

5.3.2.1.2 Partida 2: Organización

ORGANIZACIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|---|----------|----------|---------|----------|-----------------|
| Reunión recurrente con el tutor | | | | | 483,62 € |
| <i>Jefe de Proyecto</i> | 8 | horas | 60,45 € | 483,62 € | |
| <i>Reunión recurrente con stakeholders externos</i> | | | | | 483,62 € |
| <i>Jefe de Proyecto</i> | 8 | horas | 60,45 € | 483,62 € | |
| TOTAL DE ORGANIZACIÓN | | | | | 967,24 € |

5.3.2.1.3 Partida 3: Desarrollo

DESARROLLO

| Descripción | Cant. | Unidades | Precio | Sub (3) | Sub (2) | Total |
|---|-------|----------|---------|----------|----------|-------------------|
| Desarrollo de la API | | | | | | 2.584,87 € |
| Diseño inicial del modelo de base de datos | | | | | 292,63 € | |
| <i>Desarrollador de Backend</i> | 6 | horas | 24,39 € | 146,31 € | | |
| <i>Desarrollador de Frontend</i> | 6 | horas | 24,39 € | 146,31 € | | |
| Diseño inicial de la especificación de la API | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Gestión de empleados | | | | | 268,24 € | |
| <i>Desarrollador de Backend</i> | 11 | horas | 24,39 € | 268,24 € | | |
| Gestión de clientes | | | | | 268,24 € | |
| <i>Desarrollador de Backend</i> | 11 | horas | 24,39 € | 268,24 € | | |
| Gestión de usuarios | | | | | 341,40 € | |



| | | | | | | |
|--|----|-------|---------|----------|----------|-------------------|
| <i>Desarrollador de Backend</i> | 14 | horas | 24,39 € | 341,40 € | | |
| Gestión de robots | | | | | 487,71 € | |
| <i>Desarrollador de Backend</i> | 20 | horas | 24,39 € | 487,71 € | | |
| Gestión de incidencias | | | | | 536,48 € | |
| <i>Desarrollador de Backend</i> | 22 | horas | 24,39 € | 536,48 € | | |
| Gestión de roles | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | 195,08 € | | |
| Desarrollo de la Aplicación Móvil | | | | | | 2.682,41 € |
| Gestión de robots | | | | | 487,71 € | |
| <i>Desarrollador de Frontend</i> | 20 | horas | 24,39 € | 487,71 € | | |
| Gestión de clientes | | | | | 390,17 € | |
| <i>Desarrollador de Frontend</i> | 16 | horas | 24,39 € | 390,17 € | | |
| Gestión de empleados | | | | | 292,63 € | |
| <i>Desarrollador de Frontend</i> | 12 | horas | 24,39 € | 292,63 € | | |
| Asignación de robots | | | | | 341,40 € | |
| <i>Desarrollador de Frontend</i> | 14 | horas | 24,39 € | 341,40 € | | |



| | | | | | | |
|---|----|-------|---------|----------|----------|-------------------|
| Gestión de incidencias | | | | | 243,86 € | |
| <i>Desarrollador de Frontend</i> | 10 | horas | 24,39 € | 243,86 € | | |
| Gestión de sesión | | | | | 582,25 € | |
| <i>Desarrollador de Frontend</i> | 24 | horas | 24,39 € | 582,25 € | | |
| Información detallada de robots | | | | | 341,40 € | |
| <i>Desarrollador de Frontend</i> | 14 | horas | 24,39 € | 341,40 € | | |
| Desarrollo del manejo remoto básico | | | | | | 518,15 € |
| Análisis y decisión de controles básicos | | | | | 79,21 € | |
| <i>Analista de sistemas</i> | 2 | horas | 39,60 € | 79,21 € | | |
| Desarrollo de controles básicos en la API | | | | | 341,40 € | |
| <i>Desarrollador de Backend</i> | 14 | horas | 24,39 € | 341,40 € | | |
| Controles en la pantalla de información detallada | | | | | 97,54 € | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| TOTAL DE DESARROLLO: | | | | | | 5.785,43 € |

5.3.2.1.4 Partida 4: Pruebas

PRUEBAS

| Descripción | Cant. | Unidades | Precio | Sub (3) | Sub (2) | Total |
|---------------------------------------|-------|----------|---------|----------|----------|-----------------|
| Pruebas de la API | | | | | | 877,88 € |
| Pruebas unitarias | | | | | 585,25 € | |
| <i>Desarrollador de Backend</i> | 24 | horas | 24,39 € | 585,25 € | | |
| Pruebas de integración | | | | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | 195,08 € | | |
| Pruebas de seguridad | | | | | 97,54 € | |
| <i>Desarrollador de Backend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Pruebas de la Aplicación Móvil | | | | | | 634,02 € |
| Pruebas unitarias | | | | | 390,17 € | |
| <i>Desarrollador de Frontend</i> | 16 | horas | 24,39 € | 390,17 € | | |
| Pruebas de integración | | | | | 97,54 € | |
| <i>Desarrollador de Frontend</i> | 4 | horas | 24,39 € | 97,54 € | | |
| Pruebas de usabilidad | | | | | 146,31 € | |
| <i>Desarrollador de Frontend</i> | 6 | horas | 24,39 € | 146,31 € | | |



| | | | | | | |
|----------------------------------|---|-------|---------|--|----------|-------------------|
| Pruebas E2E | | | | | | 390,17 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| Pruebas en entorno real | | | | | | 631,98 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Desarrollador de Backend</i> | 8 | horas | 24,39 € | | 195,08 € | |
| <i>Jefe de Proyecto</i> | 4 | horas | 60,45 € | | 241,81 € | |
| TOTAL DE PRUEBAS: | | | | | | 2.534,05 € |

5.3.2.1.5 Partida 5: Despliegue y Distribución

DESPLIEGUE Y DISTRIBUCIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|--|----------|----------|---------|----------|-----------------|
| Despliegue de la API y base de datos | | | | | 292,63 € |
| <i>Desarrollador de Backend</i> | 12 | horas | 24,39 € | 292,63 € | |
| Distribución de la aplicación móvil | | | | | 195,08 € |
| <i>Desarrollador de Frontend</i> | 8 | horas | 24,39 € | 195,08 € | |
| TOTAL DE DESPLIEGUE Y DISTRIBUCIÓN: | | | | | 487,71 € |



5.3.2.1.6 Partida 6: Documentación

DOCUMENTACIÓN

| Descripción | Cantidad | Unidades | Precio | Sub (2) | Total |
|------------------------------------|----------|----------|---------|----------|-------------------|
| Manual de programador | | | | | 120,90 € |
| <i>Jefe de Proyecto</i> | 2 | horas | 60,45 € | 120,90 € | |
| Manual de instalación y despliegue | | | | | 241,81 € |
| <i>Jefe de Proyecto</i> | 4 | horas | 60,45 € | 241,81 € | |
| Manual de usuario | | | | | 725,43 € |
| <i>Jefe de Proyecto</i> | 12 | horas | 60,45 € | 725,43 € | |
| TOTAL DE DOCUMENTACIÓN: | | | | | 1.088,14 € |



5.3.2.2 Resumen

El presupuesto final de costes, el cual se puede ver resumido en la siguiente tabla, es un total de 12.286,61€. En el apartado [5.1.5.3 Presupuesto de Cliente](#), basándose en el presupuesto inicial, se pedía al cliente un total de 15.149,00€. Esto deja un beneficio en el proyecto de 2.862.61€.

PRESUPUESTO FINAL DE COSTES RESUMIDO

| Cod. | Partida | Total |
|------|-------------------|--------------------|
| 01 | Análisis y Diseño | 2.512,19 € |
| 02 | Organización | 967,24 € |
| 03 | Desarrollo | 5.785,43 € |
| 04 | Pruebas | 2.534,05 € |
| 05 | Despliegue | 487,71 € |
| 06 | Documentación | 1.088,14 € |
| | TOTAL | 12.286,61 € |

Capítulo 6 ANÁLISIS DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

ASI



6.1 ASI 1: DEFINICIÓN DEL SISTEMA

6.1.1 Determinación del Alcance del Sistema

El sistema se llamará MowerControl y así será reflejado a lo largo del documento.

MowerControl consistirá en una aplicación móvil que permitirá la gestión y el manejo de robots cortacéspedes en el entorno empresarial. La aplicación deberá permitir ver todos los robots asignados a una empresa, el estado en el que se encuentran y ver las incidencias.

El sistema lo usaran los empleados de las empresas para tener una mayor gestión sobre los robots instalados a los clientes desde la ubicación de la empresa o en el lugar donde esté instalado el robot, por lo que el sistema deberá ser alojado en un servidor remoto con acceso a los empleados de las empresas.

MowerControl debe permitir también una serie de funcionalidades complementarias a la visualización de los robots, tales como: Control remoto de los robots y gestión de empleados y clientes.

6.1.2 Funcionalidades de alto nivel

1. El sistema permitirá dar de altas a nuevas empresas, asignado un empleado con rol de administrador para cada una de ellas.
2. El sistema permitirá iniciar sesión a los empleados registrados anteriormente.
3. El sistema permitirá gestionar los empleados de una empresa.
4. El sistema permitirá gestionar los clientes de una empresa.
5. El sistema permitirá asignar los nuevos robots que se den de alta en el sistema oficial.
6. El sistema permitirá ver una lista de los robots asignados a cada empleado o empresa.
7. El sistema permitirá ver las incidencias causadas por los robots.
8. El sistema permitirá ver información detallada de cada robot.
9. El sistema permitirá a los empleados controlar los robots de manera remota.

6.2 ASI 2: ESTABLECIMIENTO DE REQUISITOS

6.2.1 Obtención de los Requisitos del Sistema

Las especificaciones técnicas necesarias para que un software ejecute correctamente una aplicación o programa se conocen como requisitos del sistema. Estas especificaciones garantizan que el software opere de manera eficiente y sin inconvenientes en un dispositivo o ambiente específico. Para facilitar su gestión, estos requisitos se clasifican en funcionales y no funcionales.

6.2.1.1 Requisitos funcionales

Los requisitos funcionales definen las tareas y operaciones concretas que un sistema o aplicación debe ejecutar. Estos requisitos explican las funcionalidades y comportamientos que se esperan del sistema, detallando las características y actividades que debe realizar.

1. El sistema debe permitir registrar a nuevas empresas

1.1. Solicitará los siguientes datos relacionados con la empresa

1.1.1. Nombre de la empresa.

1.1.1.1. Este campo será obligatorio.

1.1.2. CIF.

1.1.2.1. Este campo será obligatorio.

1.1.2.2. Deberá ser un CIF válido.

1.2. Solicitará los siguientes datos relacionados con el empleado administrador.

1.2.1. Nombre.

1.2.1.1. Este campo será obligatorio.

1.2.2. Primer Apellido.

1.2.2.1. Este campo será obligatorio.

1.2.3. Segundo Apellido.

1.2.3.1. Este campo será opcional.

1.2.4. Nombre de usuario.

1.2.4.1. Este campo será obligatorio.

1.2.5. Contraseña.

1.2.5.1. Este campo será obligatorio.

1.2.5.2. El sistema deberá comprobar que la contraseña es válida:

1.2.5.2.1. Debe tener al menos 8 caracteres.

1.2.5.2.2. Debe incluir al menos una letra mayúscula

1.2.5.2.3. Debe incluir al menos una letra minúscula.

1.2.5.2.4. Debe incluir al menos un número.



- 1.3. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.
- 1.4. En el caso de que los datos sean válidos, el sistema deberá permitir al usuario introducir sus credenciales de la cuenta de Husqvarna.
 - 1.4.1. Solicitará los siguientes datos
 - 1.4.1.1. Correo electrónico
 - 1.4.1.1.1. Este campo será obligatorio
 - 1.4.1.2. Contraseña
 - 1.4.1.2.1. Este campo será obligatorio
 - 1.4.2. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.
 - 1.4.3. En el caso de que los datos sean válidos:
 - 1.4.3.1. El sistema deberá guardar todos los datos descritos en 1.1 y 1.2.
 - 1.4.3.2. El sistema mostrará un mensaje al usuario de que se ha registrado correctamente.

2. El sistema deberá permitir iniciar sesión a los usuarios registrados previamente.

- 2.1. Solicitará los siguientes datos.
 - 2.1.1. Usuario.
 - 2.1.1.1. Este campo será obligatorio.
 - 2.1.2. Contraseña.
 - 2.1.2.1. Este campo será obligatorio.
- 2.2. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.
- 2.3. En el caso de que los datos sean válidos:
 - 2.3.1. El sistema deberá comprobar que el usuario existe.
 - 2.3.2. El sistema deberá comprobar que la contraseña coincida con la guardada en el sistema.
 - 2.3.3. En el caso de que lo establecido en 2.3.1 y 2.3.2 no se cumpla, el sistema mostrará un mensaje de error.
 - 2.3.4. En el caso de que lo establecido en 2.3.1 y 2.3.2 se cumpla, el sistema mostrará un mensaje al usuario de que se ha autenticado correctamente.



3. El sistema debe permitir dar de alta a nuevos empleados asignados a una empresa.

3.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

3.1.1. En el caso de que no lo sea, el sistema no deberá permitir dar de alta a un nuevo empleado.

3.2. Solicitará los siguientes datos.

3.2.1. Nombre.

3.2.1.1. Este campo será obligatorio.

3.2.2. Primer Apellido.

3.2.2.1. Este campo será obligatorio.

3.2.3. Segundo Apellido.

3.2.3.1. Este campo será opcional.

3.3. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.

3.4. En el caso de que los datos sean válidos:

3.4.1. El sistema generará un usuario utilizando los datos especificados en 3.2.

3.4.1.1. El usuario estará compuesto por el nombre completo y la primera letra del primer apellido.

3.4.1.2. En el caso de que ya exista un usuario igual, el sistema añadirá la primera letra del segundo apellido en el caso de que tenga.

3.4.1.3. En el caso de que ya exista un usuario igual, el sistema añadirá un número que será incremental.

3.4.2. El sistema generará una contraseña aleatoria

3.4.2.1. Deberá cumplir unos requerimientos únicos:

3.4.2.1.1. Deberá tener al menos 8 caracteres.

3.4.2.1.2. Deberá tener al menos una letra mayúscula.

3.4.2.1.3. Deberá tener al menos una letra minúscula.

3.4.2.1.4. Deberá tener al menos un número.

3.4.3. El sistema deberá guardar los datos especificados en 3.2, 3.4.1 y 3.4.2.

3.4.4. El sistema mostrará un mensaje al usuario que se ha registrado un nuevo empleado correctamente.

3.4.4.1. El sistema mostrará el usuario generado.

3.4.4.2. El sistema mostrará la contraseña generada.



4. El sistema permitirá ver una lista de los empleados de una empresa.

4.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

4.1.1. En el caso de que no lo sea, el sistema no deberá permitir ver la lista de empleados.

4.2. El sistema mostrará una lista con la siguiente información.

4.2.1. Nombre y apellidos del empleado.

4.2.2. Nombre de usuario del empleado.

5. El sistema permitirá eliminar empleados de una empresa

5.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

5.1.1. En el caso de que no lo sea, el sistema no deberá permitir eliminar un empleado.

5.2. El sistema solicitará el id del empleado

5.3. En el caso de que el id del empleado no exista en la base de datos, se mostrará un mensaje de error al usuario.

5.4. En el caso de que el id del empleado existe en la base de datos, se hará lo siguiente:

5.4.1. Se mostrará un mensaje de confirmación al usuario.

5.4.2. En caso de que confirme el borrado:

5.4.2.1. Todos los robots asignados a ese empleado quedarán sin empleado.

5.4.2.2. Se borrará el empleado de la base de datos.

5.4.3. En caso de que no confirme el borrado, el sistema no realizará ninguna acción.



6. El sistema debe permitir dar de alta a nuevos clientes asignados a una empresa.

6.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

6.1.1. En el caso de que no lo sea, el sistema no deberá permitir dar de alta a un nuevo cliente.

6.2. Solicitará los siguientes datos.

6.2.1. Nombre y apellidos.

6.2.1.1. Este campo será obligatorio.

6.2.2. Teléfono.

6.2.2.1. Este campo será obligatorio.

6.2.3. Dirección.

6.2.3.1. Este campo será opcional.

6.3. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.

6.4. En el caso de que los datos sean válidos:

6.4.1. El sistema guardará en la base de datos, los datos solicitados en el requisito 6.2.

6.4.2. El sistema mostrará un mensaje al usuario indicando que el cliente se ha creado correctamente.

7. El sistema permitirá ver una lista de los clientes de una empresa.

7.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

7.1.1. En el caso de que no lo sea, el sistema no deberá permitir ver la lista de clientes.

7.2. El sistema mostrará una lista con la siguiente información

7.2.1. Nombre y apellidos del cliente.

7.2.2. Número de teléfono del cliente.

7.2.3. Dirección del cliente.



8. El sistema permitirá eliminar clientes de una empresa

8.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

8.1.1. En el caso de que no lo sea, el sistema no deberá permitir eliminar un cliente.

8.2. El sistema solicitará el id del cliente.

8.3. En el caso de que el id del cliente no exista en la base de datos, se mostrará un mensaje de error al usuario.

8.4. En el caso de que el id del cliente exista en la base de datos, se hará lo siguiente:

8.4.1. Se mostrará un mensaje de confirmación al usuario.

8.4.2. En caso de que confirme el borrado:

8.4.2.1. Todos los robots asignados a ese cliente quedarán sin asignar.

8.4.2.2. Se borrará el cliente de la base de datos.

8.4.3. En caso de que no confirme el borrado, el sistema no realizará ninguna acción.

9. El sistema deberá permitir asignar los nuevos robots que se den de alta en el sistema oficial.

9.1. El sistema deberá comprobar que el empleado que realiza esta acción tiene rol de administrador en la empresa.

9.1.1. En el caso de que no lo sea, el sistema no deberá permitir asignar un robot a un cliente y empleado.

9.2. El sistema realizará una comprobación cada hora para comprobar si se han añadidos nuevos robots al sistema.

9.3. En el caso de que se hayan añadido nuevos robots, el sistema mostrará una lista de los nuevos robots.

9.3.1. El sistema deberá mostrar el nombre del robot.

9.3.2. El sistema deberá mostrar el número de serie del robot.

9.3.3. El sistema deberá mostrar el modelo del robot.

9.4. El sistema deberá permitir al usuario administrador registrar un nuevo robot.

9.4.1. Solicitará los siguientes datos.

9.4.1.1. Cliente.

9.4.1.1.1. Este campo será obligatorio.

9.4.1.2. Empleado.

9.4.1.2.1. Este campo será obligatorio.

9.4.2. En el caso de que los datos no sean válidos, el sistema mostrará un mensaje de error.

9.4.3. En el caso de que los datos sean válidos:

9.4.3.1. El sistema asignará ese robot al cliente y empleado especificados en el apartado 9.4.1.



- 9.4.3.2. El sistema mostrará un mensaje al usuario de que el robot se ha registrado con éxito en el sistema.
- 9.4.3.3. El sistema deberá mostrar este robot en la lista especificada en el apartado 10.

10. El sistema debe permitir ver al usuario una lista de los robots asignados.

- 10.1. El sistema mostrará todos los robots asignados a la empresa al empleado administrador.
- 10.2. El sistema mostrará solo los robots asignados a un empleado en caso de ser un empleado normal.
- 10.3. El sistema debe permitir a todos los usuarios buscar por diferentes valores:
 - 10.3.1. Nombre del robot.
 - 10.3.2. Número de serie.
 - 10.3.3. Nombre del cliente.
- 10.4. El sistema debe mostrar la siguiente información por cada robot:
 - 10.4.1. Nombre del robot.
 - 10.4.2. Nombre del cliente.
 - 10.4.3. Estado del robot.
 - 10.4.4. Porcentaje de batería del robot.
 - 10.4.5. Modelo del robot.
 - 10.4.6. Número de serie.



11. El sistema debe permitir ver al usuario las incidencias causadas por los robots.

- 11.1. El sistema mostrará todas las incidencias de la empresa al empleado administrador.
- 11.2. El sistema mostrará solo las incidencias de los robots asignados a un empleado en caso de ser un empleado normal.
- 11.3. El sistema realizará una comprobación cada hora del estado de todos los robots.
- 11.4. En el caso de que el estado de los robots sea error, el sistema generará una nueva incidencia.
 - 11.4.1. Guardará la siguiente información
 - 11.4.1.1. ID del robot.
 - 11.4.1.2. Código de error de la incidencia.
 - 11.4.1.3. Mensaje de error.
 - 11.4.1.4. Fecha y hora de la incidencia.
- 11.5. El sistema mostrará una lista con la siguiente información:
 - 11.5.1. Nombre del robot.
 - 11.5.2. Causa de la incidencia.
 - 11.5.3. Hora de la incidencia.
- 11.6. El sistema debe permitir al usuario marcar como leída dicha incidencia.
 - 11.6.1. Una vez marcada como leída dejara de aparecer en la lista de incidencias.

12. El sistema debe permitir al usuario ver información detallada de un robot.

- 12.1. El sistema debe permitir al usuario elegir un robot de la lista especificada en 6.
- 12.2. El sistema mostrará la siguiente información:
 - 12.2.1. Nombre del robot.
 - 12.2.2. Número de serie del robot.
 - 12.2.3. Modelo del robot.
 - 12.2.4. Estado del robot.
 - 12.2.5. Porcentaje de batería del robot.
 - 12.2.6. Cliente asignado al robot.
 - 12.2.7. Empleado asignado al robot.



13. El sistema debe permitir a los usuarios controlar los robots de manera remota.

- 13.1. El sistema debe permitir al usuario poner en marcha el robot.
 - 13.1.1. En el caso de que la operación no sea válida, el sistema deberá mostrar un mensaje de error.
 - 13.1.2. En el caso de que la operación se válida:
 - 13.1.2.1. El estado del robot deberá cambiar a CORTANDO.
 - 13.1.2.2. El sistema mostrará un mensaje indicando que la orden ha sido procesada correctamente.
- 13.2. El sistema debe permitir al usuario parar el robot.
 - 13.2.1. En el caso de que la operación no sea válida, el sistema deberá mostrar un mensaje de error.
 - 13.2.2. En el caso de que la operación sea válida.
 - 13.2.2.1. El estado del robot deberá cambiar a PARADO.
 - 13.2.2.2. El sistema mostrará un mensaje indicando que la orden ha sido procesada correctamente.
- 13.3. El sistema debe permitir al usuario cargar el robot.
 - 13.3.1. En el caso de que la operación no sea válida, el sistema deberá mostrar un mensaje de error.
 - 13.3.2. En el caso de que la operación sea válida.
 - 13.3.2.1. El estado del robot deberá cambiar a CARGANDO.
 - 13.3.2.2. El sistema mostrará un mensaje indicando que la orden ha sido procesada correctamente.

6.2.1.2 Requisitos no funcionales

1. El sistema deberá utilizar el protocolo HTTPS para asegurar la confidencialidad de los datos en las conexiones.
2. La aplicación móvil deberá ser compatible con los sistemas operativos Android y iOS.
3. La aplicación móvil deberá ser usable, proporcionando una interfaz intuitiva y fácil de usar.



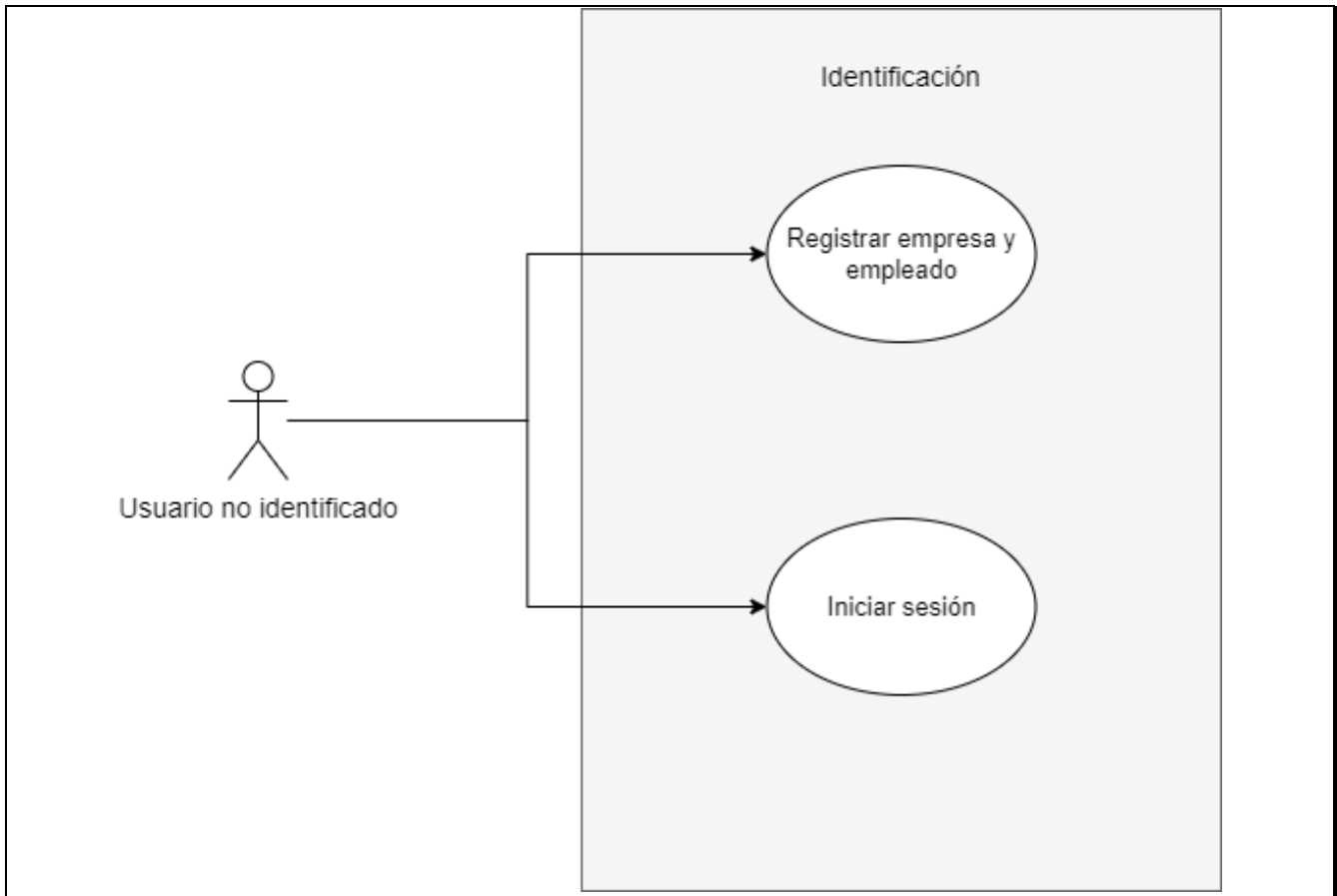
6.2.2 Identificación de Actores del Sistema

Los actores que se podrán encontrar en el sistema son los siguientes:

- **Usuario no identificado:** Usuario que no se ha identificado en el sistema.
- **Usuario identificado:** Usuario no administrador que se ha identificado en el sistema.
- **Usuario administrador:** Usuario de tipo administrador que se ha identificado en el sistema.
- **Husqvarna Automower API:** Proveedor de la información de los robots y manejo de estos.
- **Firebase:** Proveedor del servicio de notificaciones.

6.2.3 Especificación de Casos de Uso

6.2.3.1 Usuario no identificado



| |
|-------------------------------|
| Nombre del Caso de Uso |
|-------------------------------|

| |
|-------------------------------|
| Registrar empresa y empleado. |
|-------------------------------|

| |
|--------------------|
| Descripción |
|--------------------|

| |
|--|
| El usuario no identificado introducirá los datos necesarios para registrar una nueva empresa y el usuario administrador de la misma. |
|--|

| |
|---|
| El sistema validará los datos y los guardará en el sistema de persistencia. |
|---|

| |
|-------------------------------|
| Nombre del Caso de Uso |
|-------------------------------|

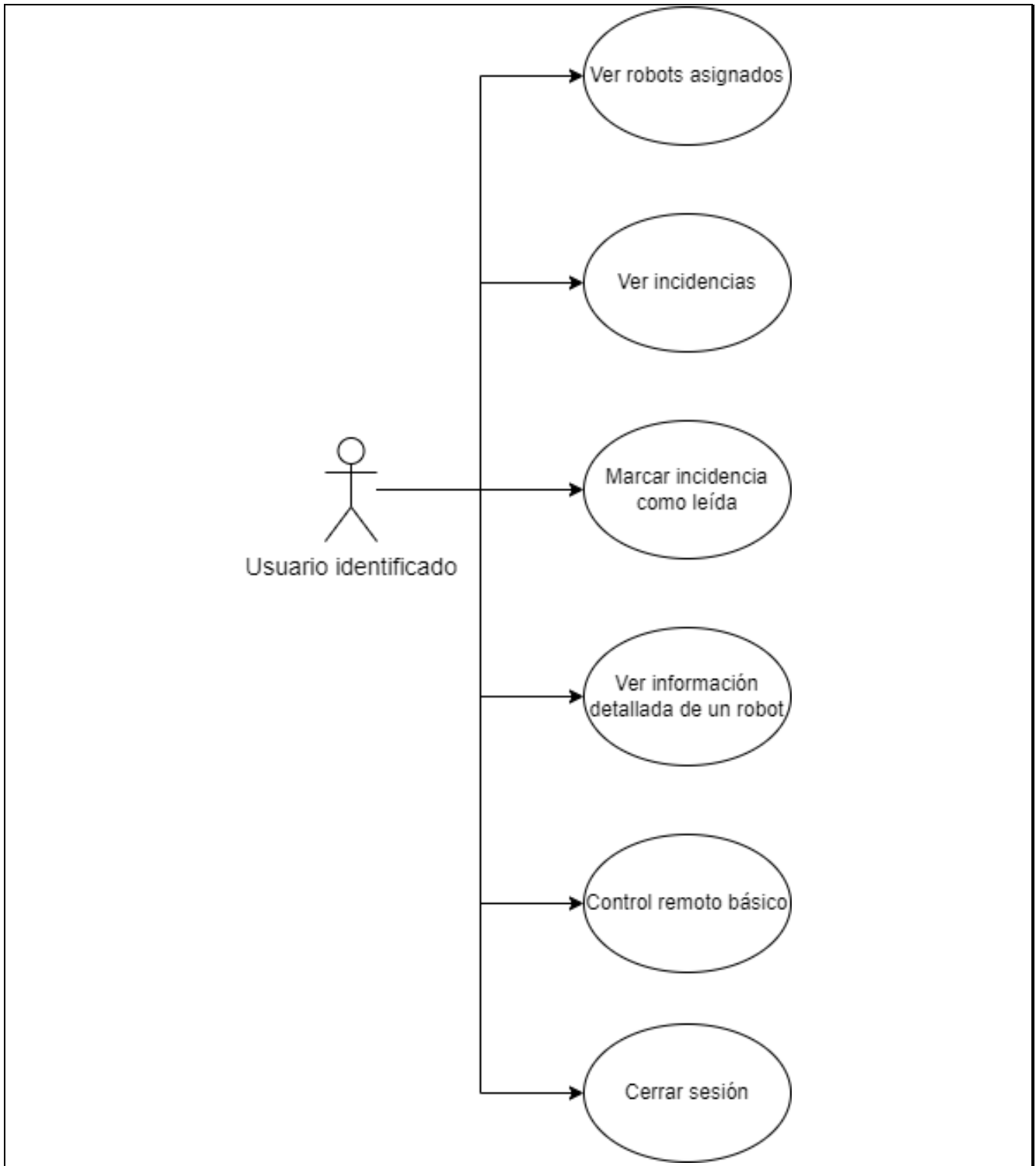
| |
|-----------------|
| Iniciar sesión. |
|-----------------|

| |
|--------------------|
| Descripción |
|--------------------|

| |
|--|
| El usuario no identificado introducirá sus credenciales para iniciar sesión. |
|--|

| |
|---------------------------------------|
| El sistema validará las credenciales. |
|---------------------------------------|

6.2.3.2 Usuario identificado





| |
|--|
| Nombre del Caso de Uso |
| Ver robots asignados. |
| Descripción |
| El usuario identificado verá todos los robots que tiene asignados, con información básica de cada uno. |

| |
|---|
| Nombre del Caso de Uso |
| Ver incidencias. |
| Descripción |
| El usuario identificado verá una lista con las incidencias para los robots asignados. |

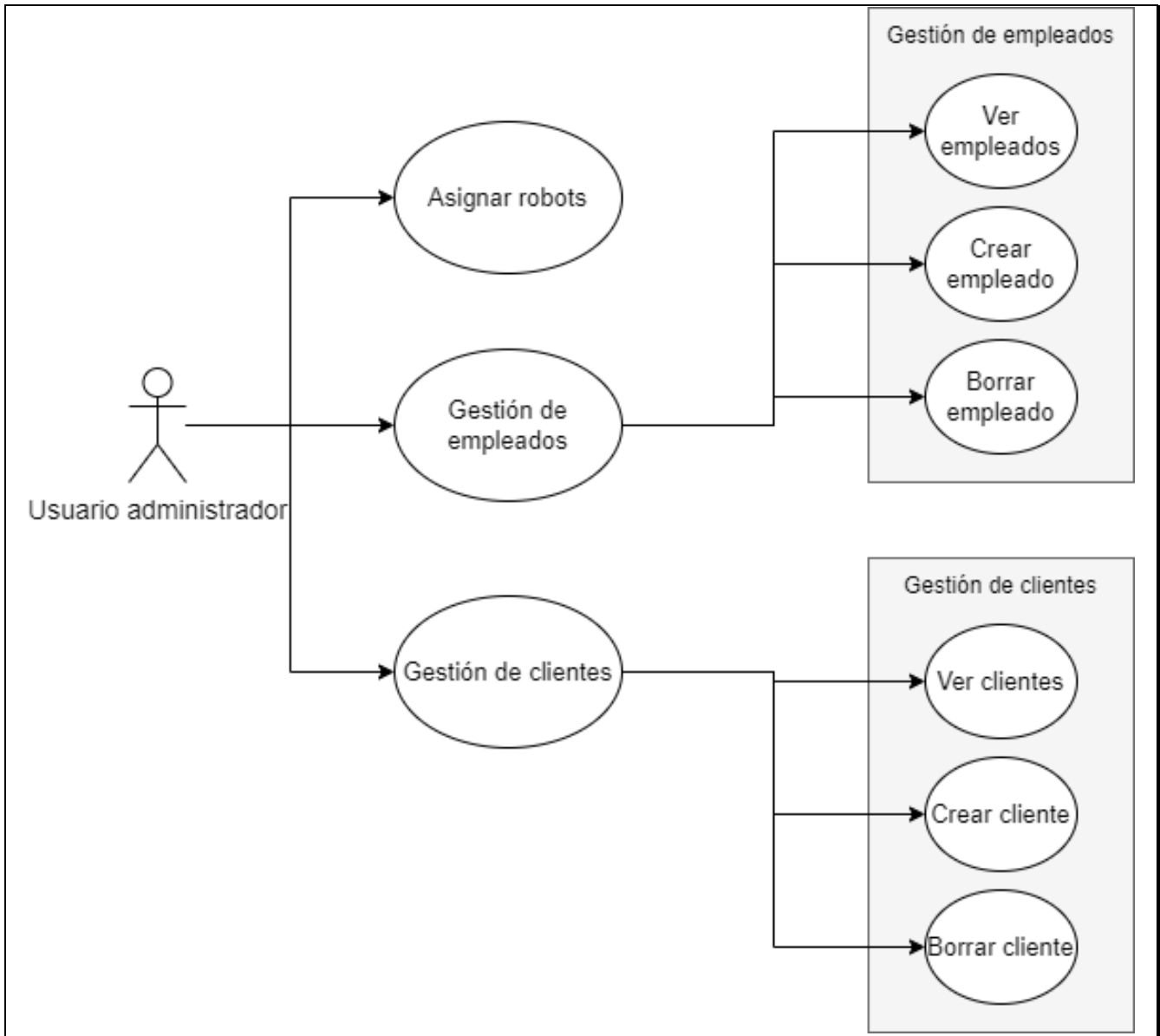
| |
|---|
| Nombre del Caso de Uso |
| Marcar incidencia como leída. |
| Descripción |
| El usuario identificado marcará una incidencia como leída. Esta incidencia desaparecerá de la lista de incidencias. |

| |
|--|
| Nombre del Caso de Uso |
| Ver información detallada de un robot. |
| Descripción |
| El usuario identificado verá información detallada de un robot. Todos los datos, estado actual del robot, así como el cliente y empleados asignados a dicho robot. |

| |
|---|
| Nombre del Caso de Uso |
| Control remoto básico. |
| Descripción |
| El usuario identificado controlará los robots de manera remota. Podrá arrancar, parar y mandar a cargar a un robot. |

| |
|---|
| Nombre del Caso de Uso |
| Cerrar sesión. |
| Descripción |
| El usuario identificado cerrará sesión del sistema. |

6.2.3.3 Usuario administrador





| |
|--|
| Nombre del Caso de Uso |
| Asignar robots. |
| Descripción |
| El usuario administrador asignará un robot a un cliente y un empleado. El robot aparecerá en la lista principal de robots. |

| |
|---|
| Nombre del Caso de Uso |
| Gestión de empleados. |
| Descripción |
| El usuario administrador gestionará los empleados de su empresa. El usuario administrador verá una lista con todos los empleados. El usuario administrador creará un empleado introduciendo sus datos. El sistema validará los datos y los almacenará en el sistema de persistencia. El usuario administrador borrará un empleado. El sistema eliminará los datos del sistema de persistencia. |

| |
|---|
| Nombre del Caso de Uso |
| Gestión de clientes. |
| Descripción |
| El usuario administrador gestionará los clientes de su empresa. El usuario administrador verá una lista con todos los clientes. El usuario administrador creará un cliente introduciendo sus datos. El sistema validará los datos y los almacenará en el sistema de persistencia. El usuario administrador borrará un cliente. El sistema eliminará los datos del sistema de persistencia. |



6.3 ASI 3: IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS

El sistema que se describe en este documento se puede dividir en los subsistemas que se describen a continuación, para así facilitar su posterior análisis.

6.3.1 Descripción de los Subsistemas

6.3.1.1 Sistema de interfaz de usuario (UI)

El Sistema de Interfaz de Usuario (UI) se enfocará en el desarrollo y manejo de los elementos visuales e interactivos que los usuarios emplean para interactuar con la aplicación. Su principal objetivo es ofrecer una experiencia de usuario que sea clara, intuitiva y agradable, facilitando la navegación, la comprensión de las funciones disponibles y la realización de tareas específicas.

Esto abarca el diseño de pantallas, botones, menús, formularios y otros componentes visuales, así como la implementación de respuestas visuales y sonoras a las acciones de los usuarios.

6.3.1.2 Sistema de entrada

El sistema de entrada es el que se encargará de recibir todas las peticiones que los usuarios hagan a través de la interfaz. Realizará validaciones de seguridad, para evitar el acceso no deseado de otros usuarios ajenos al sistema o que no tengan el rol adecuado para acceder a determinados recursos del sistema. Además, también hará pequeñas validaciones de datos que vengan en la petición del usuario, para poder devolver lo más rápido posible información al usuario en caso de error en su petición.

6.3.1.3 Sistema de negocio

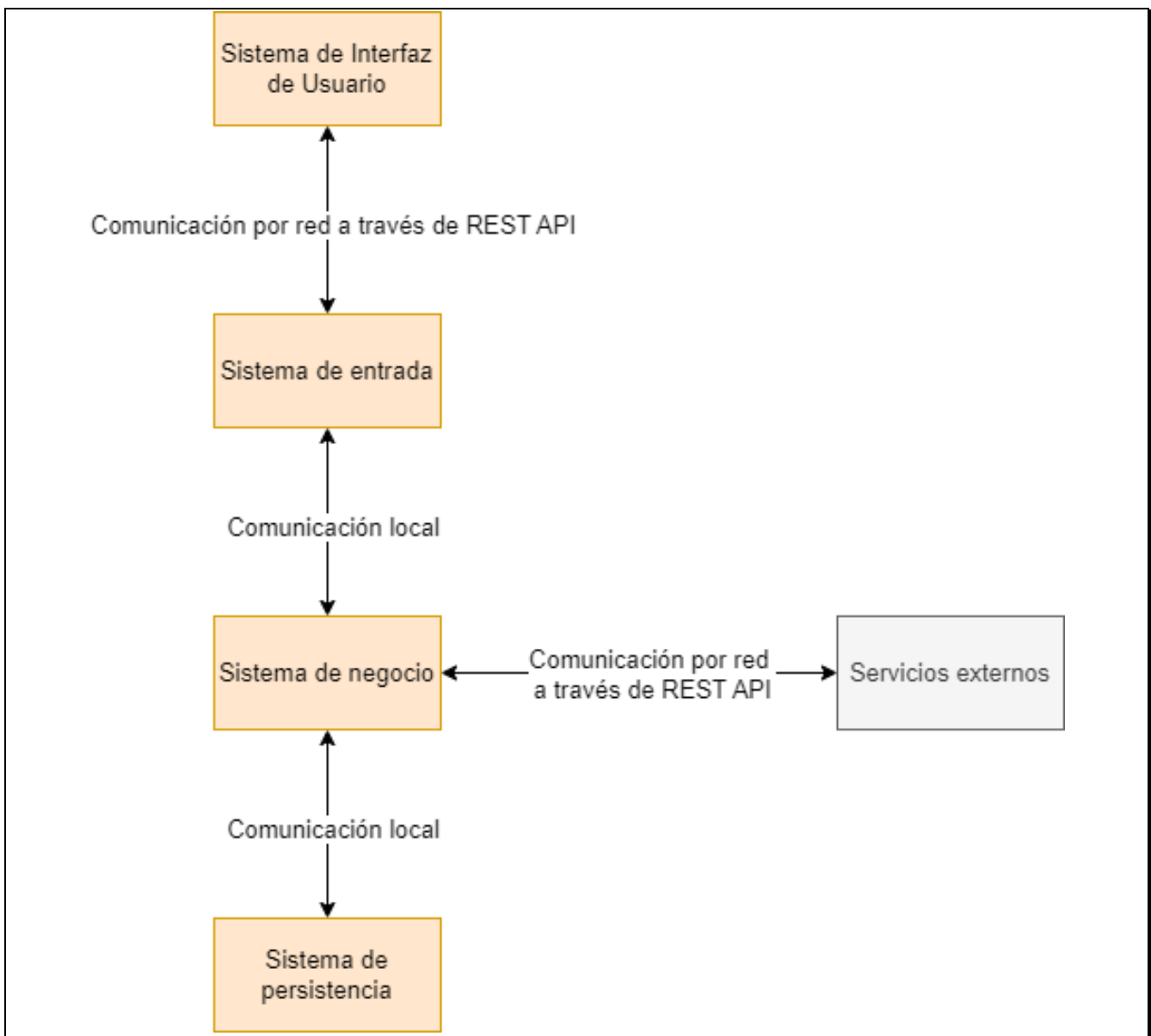
El sistema de negocio se encargará de procesar las peticiones que lleguen del sistema de entrada y llevar a cabo toda la lógica y acciones del sistema. Organizará datos de todos los recursos del sistema, para devolverle la máxima información posible al usuario. Además comprobará extensivamente que todos los datos del sistema son correctos y válidos. Por último, también será el encargado de conectarse con servicios externos para obtener y mandar información.

6.3.1.4 Sistema de persistencia

El sistema de persistencia se responsabilizará de manejar y guardar los datos de forma eficaz, garantizando su disponibilidad y consistencia a lo largo del tiempo. Esto incluye procedimientos para crear, modificar, eliminar y obtener registros en la base de datos. Por último, maneja todos los errores de conexión y de datos que pueda haber con el sistema de almacenamiento.

6.3.2 Descripción de los Interfaces entre Subsistemas

A continuación, se muestra el diagrama con las relaciones entre los subsistemas como así el tipo de comunicación que existirá entre ellas.



6.4 ASI 4: ANÁLISIS DE LOS CASOS DE USO

En este apartado, se realizará un análisis de los casos de uso importantes del sistema, quitando aquellos que solo sean para la visualización de datos o con iteraciones muy sencillas.

6.4.1 Caso de Uso 1: Registro

| REGISTRO | |
|---|--|
| PRECONDICIONES | El usuario deberá ser no identificado. |
| POSCONDICIONES | El usuario y la empresa estarán registradas. |
| ACTORES | Iniciada por un usuario no identificado y terminado por un usuario administrador registrado. |
| DESCRIPCIÓN | <p>El sistema muestra la pantalla de registro.</p> <p>El usuario introduce todos los datos necesarios (nombre de la empresa, CIF, su nombre, usuario, etc.).</p> <p>El sistema valida la información introducida.</p> <p>El sistema redirige al inicio de sesión del proveedor externo.</p> <p>El usuario introduce sus credenciales.</p> <p>El proveedor externo valida las credenciales.</p> <p>El sistema redirige a la pantalla de registro.</p> <p>El usuario entra correctamente en el sistema.</p> |
| VARIACIONES (ESCENARIOS SECUNDARIOS) | <p>Escenario Alternativo 1: Faltan campos obligatorios o datos incorrectos en el formulario de registro.</p> <p>Notificar el error al usuario.</p> <p>Volver al paso 1 del escenario principal.</p> <p>Escenario Alternativo 2: Credenciales del proveedor externo incorrectas.</p> <p>Notificar el error al usuario.</p> <p>Volver al paso 4 del escenario principal.</p> <p>Escenario Alternativo 3: Empresa con el mismo CIF o usuario con el mismo nombre de usuarios ya existentes.</p> <p>Notificar el error al usuario.</p> <p>Volver al paso 1 del escenario principal.</p> |
| EXCEPCIONES | <p>La REST API no responde: En caso de este error, se mostrará al usuario un mensaje de error.</p> <p>El proveedor externo no responde: En caso de este error, se mostrará al usuario un mensaje de error.</p> |



6.4.2 Caso de Uso 2: Inicio de sesión

| INICIO DE SESION | |
|---|--|
| PRECONDICIONES | El usuario deberá ser no identificado. |
| POSCONDICIONES | El usuario está identificado. |
| ACTORES | Iniciada por un usuario no identificado y terminado por un usuario identificado dentro del sistema. |
| DESCRIPCIÓN | El sistema muestra la pantalla de inicio de sesión. El usuario introduce su nombre de usuario y contraseña. El sistema valida la información introducida. El usuario accede al sistema correctamente. |
| VARIACIONES (ESCENARIOS SECUNDARIOS) | Escenario Alternativo 1: Faltan campos obligatorios o datos incorrectos en el formulario de registro. Notificar el error al usuario. Volver al paso 1 del escenario principal. Escenario Alternativo 2: Usuario y/o contraseña incorrectos Notificar el error al usuario. Volver al paso 1 del escenario principal. |
| EXCEPCIONES | La REST API no responde: En caso de este error, se mostrará al usuario un mensaje de error. |
| NOTAS | - |

6.4.3 Caso de uso 3: Asignar robot

| ASINGAR ROBOT | |
|---|--|
| PRECONDICIONES | El usuario debe estar identificado y tener rol de administrador. |
| POSCONDICIONES | El robot está asignado a un cliente y empleado. |
| ACTORES | Iniciado y terminado por un usuario administrador. |
| DESCRIPCIÓN | <p>El sistema muestra la lista de robots.</p> <p>El usuario accede a la lista de robots sin asignar.</p> <p>El usuario elige un robot para asignar.</p> <p>El sistema le muestra la pantalla para asignar un robot.</p> <p>El usuario elige un cliente de la lista.</p> <p>El usuario elige un empleado de la lista.</p> <p>El usuario confirma la selección.</p> <p>El sistema asigna el robot al cliente y empleado.</p> <p>El sistema redirige a la lista de robots y muestra el nuevo.</p> |
| VARIACIONES (ESCENARIOS SECUNDARIOS) | <p>Escenario Alternativo 1: El robot ya tiene asignado un cliente. El usuario podrá saltarse el paso 5 del escenario principal.</p> <p>Escenario Alternativo 2: El robot ya tiene asignado un empleado. El usuario podrá saltarse el paso 6 del escenario principal.</p> <p>Escenario Alternativo 3: Cliente y/o empleado no seleccionado. Notificar el error al usuario. Volver al paso 4 del escenario principal.</p> |
| EXCEPCIONES | La REST API no responde: En caso de este error, se mostrará al usuario un mensaje de error. |
| NOTAS | - |

6.4.4 Caso de uso 4: Creación de un empleado

| CREACIÓN DE UN EMPLEADO | |
|---|---|
| PRECONDICIONES | El usuario debe estar identificado y tener rol de administrador. |
| POSCONDICIONES | Un nuevo empleado y usuario es creado para la empresa. |
| ACTORES | Iniciado y terminado por un usuario administrador. |
| DESCRIPCIÓN | <p>El sistema muestra la lista de empleados.</p> <p>El usuario accede al formulario para crear uno nuevo.</p> <p>El usuario introduce los datos necesarios. Nombre, primer apellido y segundo apellido, este último es opcional.</p> <p>El usuario confirma la creación.</p> <p>El sistema registra al nuevo empleado.</p> <p>El sistema muestra al usuario las credenciales para el nuevo empleado.</p> <p>El usuario confirma haber guardado dichas credenciales.</p> <p>El sistema redirige al usuario a la lista de empleados y muestra el nuevo.</p> |
| VARIACIONES (ESCENARIOS SECUNDARIOS) | <p>Escenario Alternativo 1: Faltan datos obligatorios o datos incorrectos en el formulario.</p> <p>Notificar el error al usuario.</p> <p>Volver al paso 2 del escenario principal.</p> |
| EXCEPCIONES | <p>La REST API no responde: En caso de este error, se mostrará al usuario un mensaje de error.</p> |
| NOTAS | - |

6.4.5 Caso de uso 5: Borrado de un cliente

| BORRADO DE UN CLIENTE | |
|---|--|
| PRECONDICIONES | El usuario debe estar identificado y tener rol de administrador. |
| POSCONDICIONES | Un cliente es borrado de la empresa. |
| ACTORES | Iniciado y terminado por un usuario administrador. |
| DESCRIPCIÓN | <p>El sistema muestra la lista de clientes.</p> <p>El usuario desliza hacia la izquierda el cliente que quiere borrar.</p> <p>El sistema muestra un diálogo de confirmación.</p> <p>El usuario confirma la acción.</p> <p>El sistema elimina al cliente de la base de datos.</p> <p>El sistema muestra un botón durante unos segundos para deshacer la acción.</p> <p>El usuario no deshace la acción.</p> <p>El cliente es borrado definitivamente.</p> |
| VARIACIONES (ESCENARIOS SECUNDARIOS) | <p>Escenario Alternativo 1: El usuario deniega la acción.</p> <p>El cliente no se elimina.</p> <p>Volver al paso 1 del escenario principal.</p> <p>Escenario Alternativo 2: El usuario deshace la acción.</p> <p>El cliente se vuelve a introducir en el sistema.</p> <p>Volver al paso 1 del escenario principal.</p> |
| EXCEPCIONES | <p>La REST API no responde: En caso de este error, se mostrará al usuario un mensaje de error.</p> |
| NOTAS | - |

6.5 ASI 8: DEFINICIÓN DE INTERFACES DE USUARIO

En esta sección se presentan los diseños preliminares de la aplicación móvil multiplataforma que es parte del sistema desarrollado. Dado que estos son diseños iniciales, es posible que se modifiquen y sufran pequeños cambios en etapas posteriores del desarrollo.

Todos los diseños de la aplicación fueron creados utilizando el programa Balsamiq Wireframes.

6.5.1 Descripción de la Interfaz

6.5.1.1 Pantalla de inicio de sesión



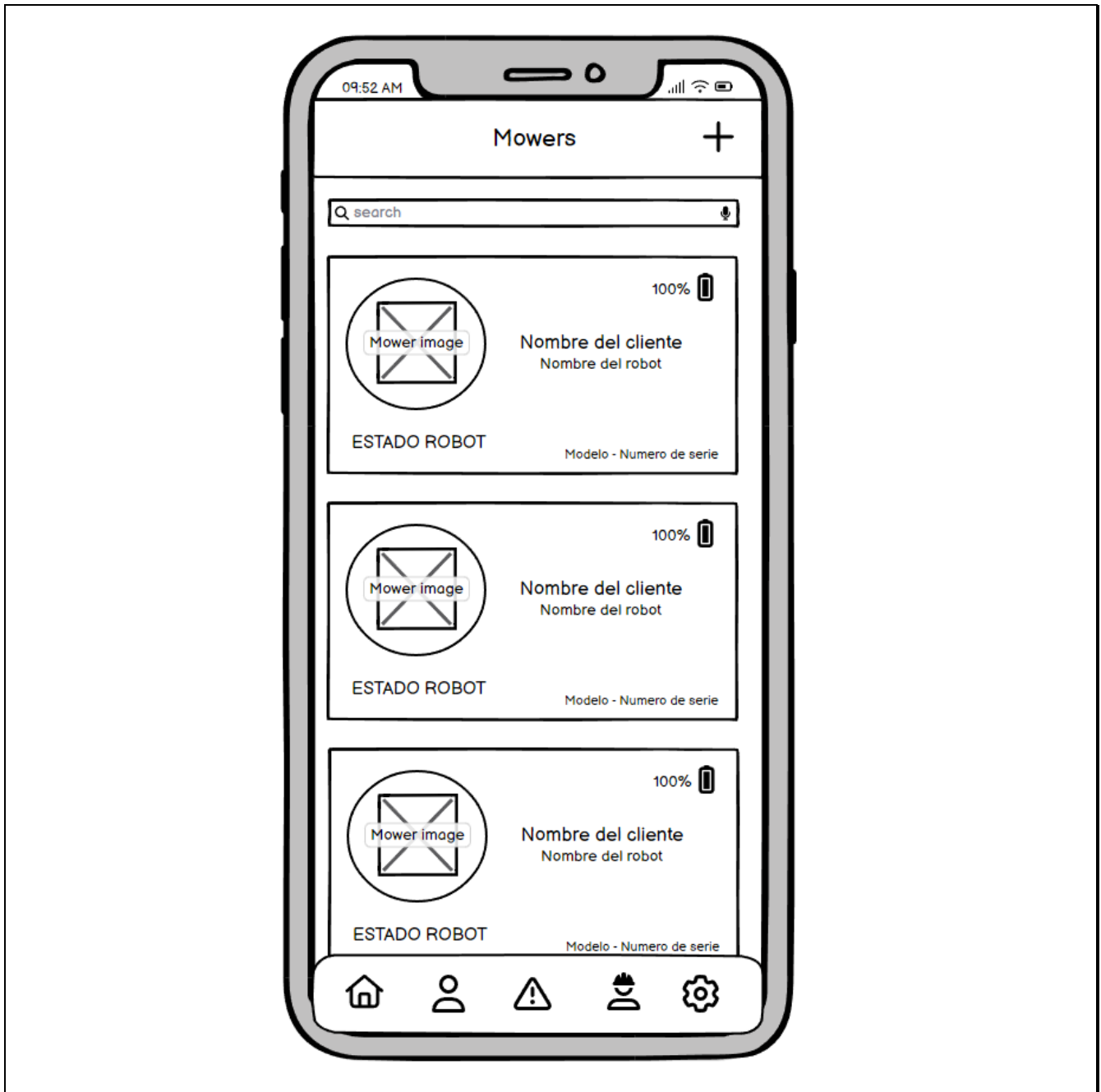
La pantalla de inicio de la aplicación es la pantalla donde el usuario puede introducir sus credenciales para iniciar sesión. En la parte de arriba se encuentra una imagen con el logo de la aplicación. También hay un botón “Crear una cuenta” que nos lleva a la pantalla de registro explicada en el siguiente apartado.

6.5.1.2 Pantalla de registro



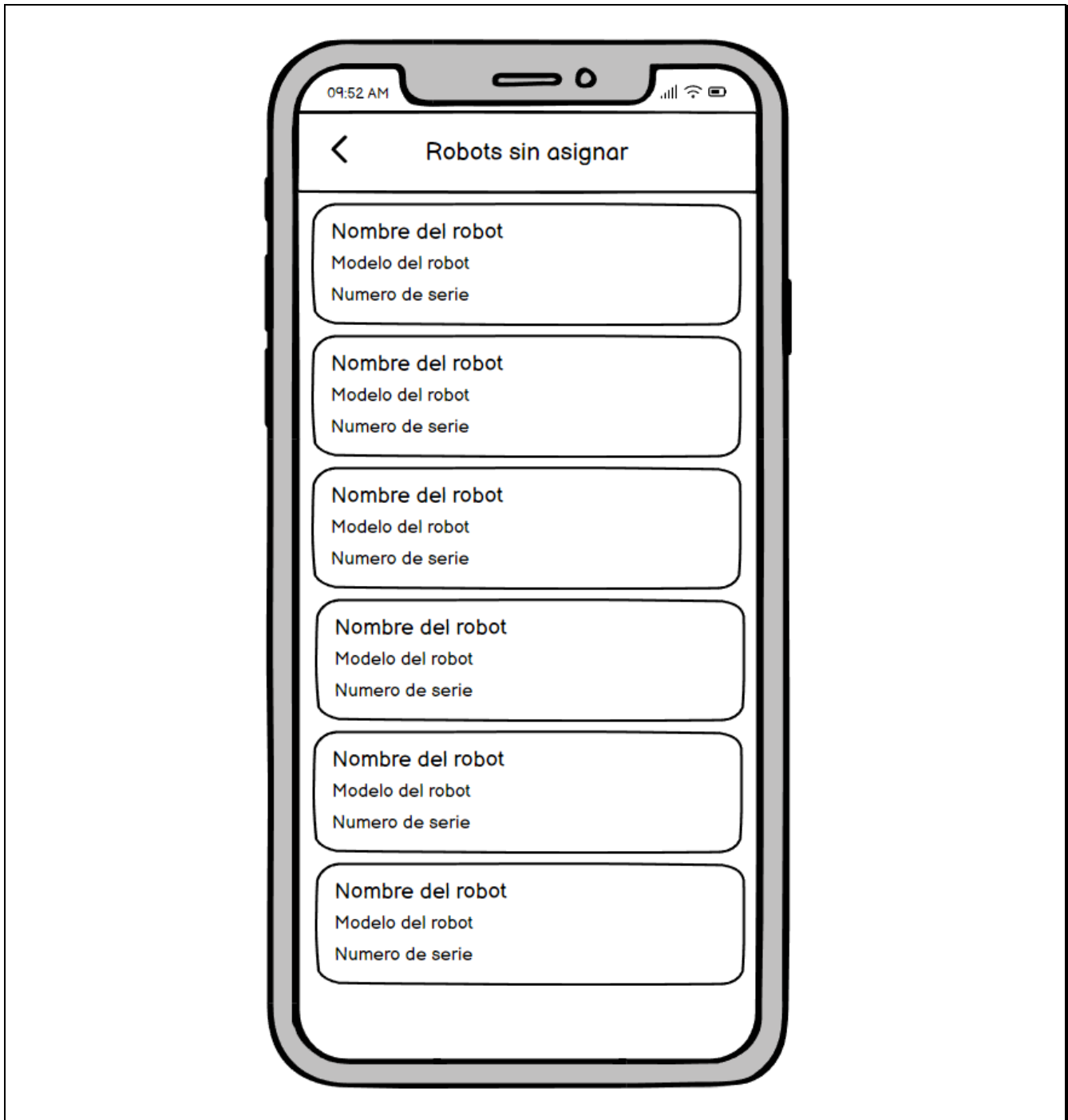
La pantalla de registro se compone también del logo de la aplicación y todos los campos necesarios para que el usuario registre una nueva empresa y un usuario administrador para esta. Además cuenta con un botón “Ya tengo una cuenta” que nos lleva a la pantalla de inicio de sesión especificada en el apartado anterior.

6.5.1.3 Pantalla de lista de robots



Esta es la pantalla principal de la aplicación, donde un usuario accede cuando inicia sesión o se registra. En esta pantalla se puede ver una lista de todos los robots que ya están asignados, es decir tienen empleado y cliente. Podemos ver una imagen del modelo del robot, su estado, la batería e información básica del mismo. En la parte de arriba encontramos una barra de búsqueda, para que el usuario pueda encontrar el robot de forma más sencilla. En el botón "+" de arriba a la derecha, nos lleva a la lista de robots sin asignar que se describe a continuación.

6.5.1.4 Pantalla de robots sin asignar



En esta pantalla se muestra una lista de los robots que están sin asignar, es decir que no tienen cliente y/o empleado asignado. Por lo tanto no aparecen en la pantalla principal. Si el usuario hace clic en cualquiera de los robots, le lleva a la pantalla para asignar dicho robot. Esta pantalla se describe en el apartado siguiente.

6.5.1.5 Pantalla de asignar un robot

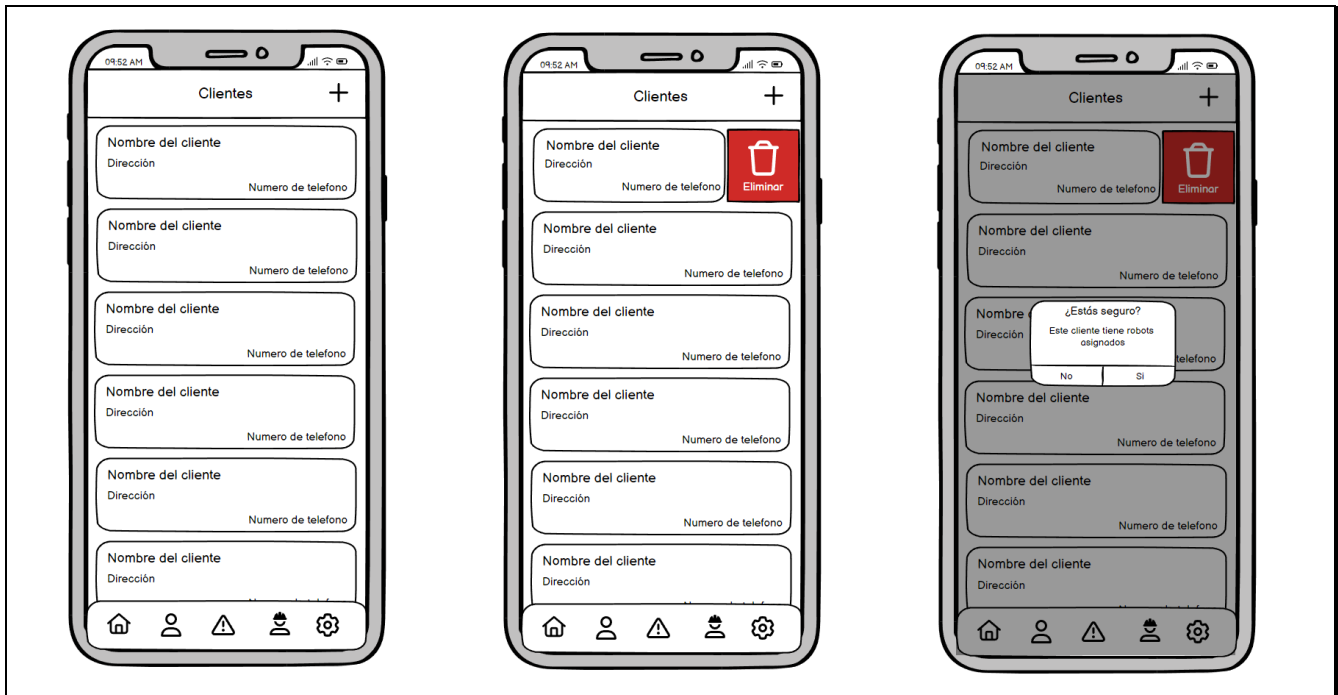


The screenshot displays a mobile application interface for assigning a robot. The screen is titled "Asignar robot" and features a back arrow in the top left corner. The status bar at the top shows the time as 09:52 AM and signal strength indicators. The main content area contains a form with the following elements:

- A rounded rectangular box containing three text input fields: "Nombre del robot", "Modelo del robot", and "Numero de serie".
- A dropdown menu labeled "Cliente".
- A dropdown menu labeled "Empleado".
- A button labeled "Asignar robot" centered at the bottom of the form.

En esta pantalla, el usuario puede asignar un robot a un empleado y cliente. En la parte superior aparece la información básica del robot a asignar. Justo debajo podemos encontrar dos selectores, para que el usuario seleccione el cliente y empleado que desee. Estos selectores contarán también con un buscador, para encontrar más fácil al cliente y empleado. Una vez que el usuario asigna el robot, vuelve a la pantalla principal y ya puede ver el robot en dicha lista.

6.5.1.6 Pantalla de lista de clientes

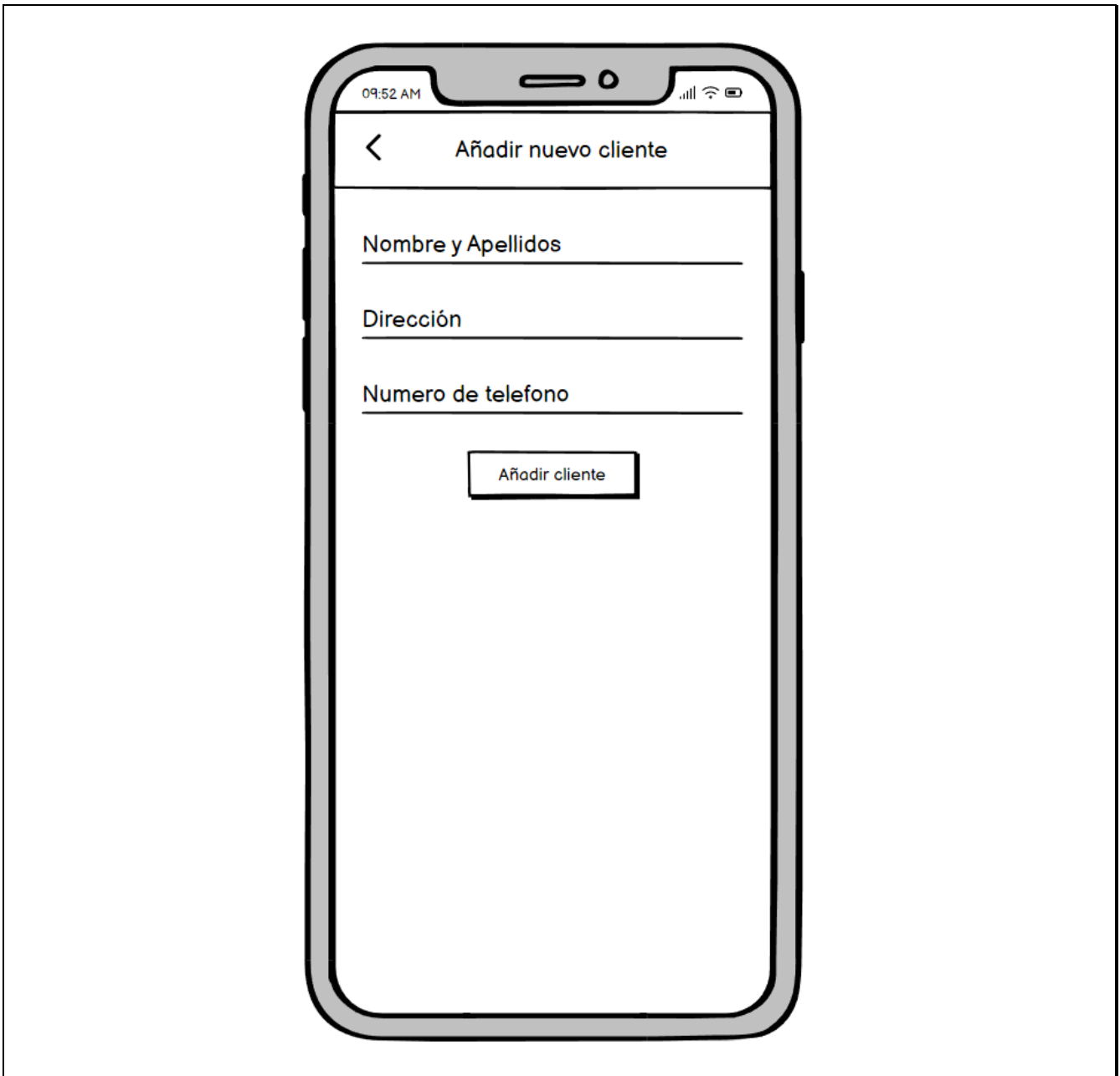


En esta pantalla solo disponible para administradores, se podrá ver una lista de todos los clientes de la empresa. Para cada cliente se muestra, su nombre, la dirección y el número de teléfono.

Deslizando cada cliente hacia la izquierda, da la posibilidad al usuario de eliminar un cliente. En caso de que realice esta opción, saldrá un dialogo de confirmación, donde se informa al usuario de si el cliente tiene robots asignados. Y le permite cancelar el borrado o confirmarlo.

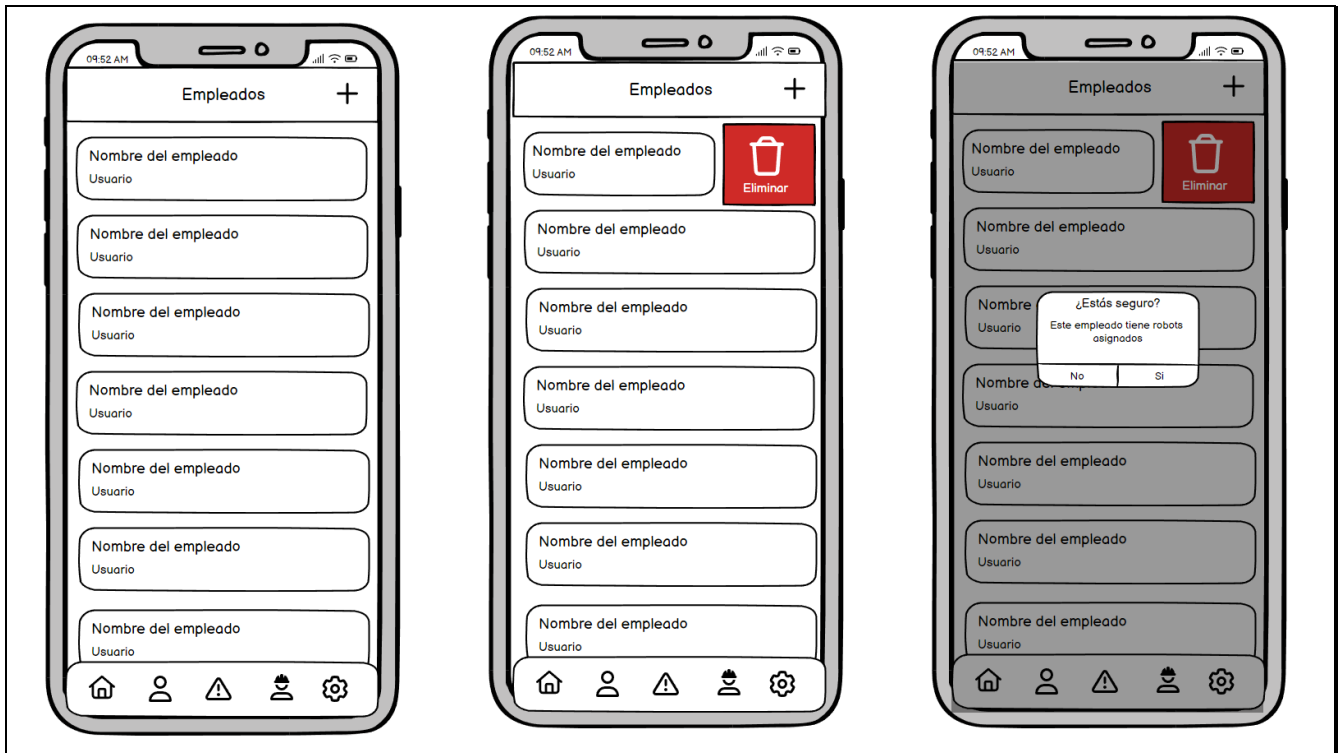
Haciendo uso del botón “+” de arriba a la derecha, lleva a la pantalla que se describe a continuación, donde el usuario podrá crear un cliente.

6.5.1.7 Pantalla de añadir nuevo cliente



En esta pantalla, el usuario podrá añadir un nuevo cliente. Tendrá que rellenar los campos de nombre y apellidos, dirección y número de teléfono. Una vez añadido y confirmado, se redirigirá de nuevo a la pantalla de lista de clientes, donde ya aparecerá el nuevo.

6.5.1.8 Pantalla de lista de empleados



En esta pantalla solo disponible para administradores, se podrá ver una lista de todos los empleados de la empresa. Para cada empleado se muestra, su nombre y apellidos, justo debajo se muestra también su nombre de usuario

Deslizando cada empleado hacia la izquierda, da la posibilidad al usuario de eliminar un empleado. En caso de que realice esta opción, saldrá un dialogo de confirmación, donde se informa al usuario de si el empleado tiene robots asignados. Y le permite cancelar el borrado o confirmarlo.

Haciendo uso del botón “+” de arriba a la derecha, lleva a la pantalla que se describe a continuación, donde el usuario podrá crear un empleado nuevo.

6.5.1.9 Pantalla de añadir nuevo empleado



The screenshot shows a mobile application interface for adding a new employee. The screen is titled "Añadir nuevo empleado" and features three input fields for "Nombre", "Primer apellido", and "Segundo apellido (opcional)". A button labeled "Añadir empleado" is positioned below the fields. The status bar at the top indicates the time as 09:52 AM and shows signal, Wi-Fi, and battery icons.

En esta pantalla, el usuario podrá añadir un nuevo empleado. Tendrá que rellenar los campos de nombre, primer apellido y segundo apellido. Este último es opcional, mientras que los dos primeros son obligatorios. Una vez añadido y confirmado, se redirigirá de nuevo a la pantalla de lista de empleados, donde ya aparecerá el nuevo.

6.5.1.10 Pantalla de lista de incidencias



En esta pantalla, el usuario podrá ver una lista de las incidencias causadas por los robots que tiene asignados. Para cada incidencia, aparecerá el mensaje de error, la fecha y la hora de cuando se produjo el error y el nombre del robot que causa la incidencia. Las incidencias aparecerán ordenadas por fecha y hora, de la más nueva a la más antigua.

Si el usuario desliza para la derecha, una incidencia, podrá marcarla como leída. Una vez hecho esto la incidencia no aparecerá más en esta lista. Excepto si el usuario deshace la operación, ya que justo al marcar como leída, aparece en la parte inferior, una snackbar donde hay un botón para deshacer la operación.

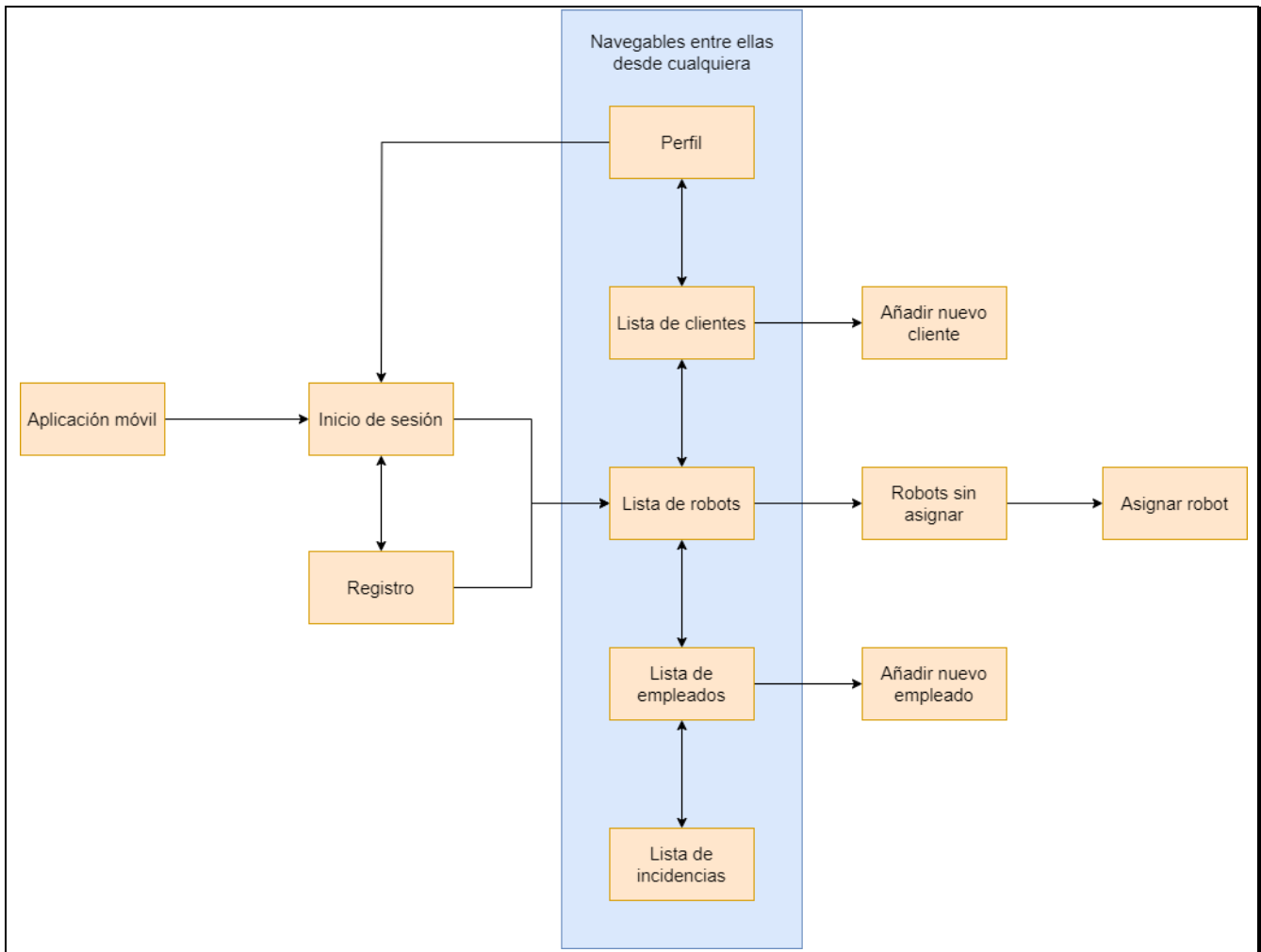
6.5.1.11 Pantalla de perfil



Esta es una pantalla muy sencilla, donde puede ver su nombre de usuario y la empresa a la que está asignado. Además, un botón para poder cerrar sesión en la aplicación. Haciendo clic ahí le mandará a la pantalla de inicio de sesión descrita en el primer apartado.

6.5.2 Diagrama de Navegabilidad

En este apartado, se muestra el diagrama de navegabilidad de la aplicación móvil y sus vistas.





6.6 ASI 10: ESPECIFICACIÓN DEL PLAN DE PRUEBAS

A continuación, se especifican todos los tipos de pruebas que se van a realizar en este proyecto para comprobar el funcionamiento del sistema. Se dividen en dos partes debido a su naturaleza, ya que la mayoría se ejecutarán de manera automática y algunas de manera manual.

6.6.1 Pruebas automáticas

Todas las pruebas automáticas del sistema, tanto de la REST API como la aplicación móvil, se ejecutarán de manera automática cada vez que se realice un commit a la rama principal en cada repositorio de GitHub. Esto se consigue gracias a las GitHub Actions que nos proporciona la plataforma de control de versiones.

6.6.1.1 Pruebas unitarias

Se realizarán pruebas unitarias para cada componente de la aplicación. La REST API maneja casi toda la lógica de la aplicación por lo que será probada más a fondo que la aplicación móvil.

En la REST API se harán pruebas a los controladores, servicios y repositorios por separado, evaluando las posibles entradas y salidas de cada caso de uso, para así comprobar todos los escenarios y que la cobertura sea lo más alta posible.

Para hacer estas pruebas se usará el framework Jest, que es un marco de pruebas en JavaScript muy popular y eficaz. Nos permite mockear los diferentes partes del sistema, para así poder aislar por completo el componente que estamos probando.

Respecto a la aplicación móvil, se harán también estas pruebas de cada componente, comprobando que el funcionamiento de pequeñas partes de lógica de la APP, así como que todo se visualice correctamente.



6.6.1.2 Pruebas de integración o de sistema

Para comprobar que el componente principal (REST API) que maneja toda la lógica del sistema funciona de manera completa, se realizarán pruebas de integración, o también las podemos llamar de sistema, ya que comprobarán el flujo completo del sistema.

Las pruebas de integración de la REST API, se realizarán usando el framework Jest, al igual que las pruebas unitarias y también la tecnología TestContainers, la cual nos servirá para simular el funcionamiento de una base de datos real y así comprobar que nuestro sistema funciona de manera completo y se conecta de manera satisfactoria a una base de datos.

6.6.1.3 Pruebas de seguridad o control de acceso

Este tipo de pruebas al igual que las anteriores se realizarán en los dos componentes principales del sistema. Se comprobará que cada tipo de usuario puede acceder a los recursos a los que se le permite con su rol.

En las REST API, se comprobará la mayoría de estas pruebas de seguridad. Se comprobará que los usuarios de tipo administrador pueden modificar recursos como los clientes o empleados, que los usuarios de tipo estándar no pueden. Y una parte muy importante del sistema, es comprobar que un usuario no puede acceder a cualquier información de otra empresa.

Respecto a la aplicación móvil, solo se comprobará que el usuario puede ver las vistas a las que tiene acceso, para que no pueda hacer uso de determinadas funcionalidades del sistema.



6.6.2 Pruebas manuales

6.6.2.1 Pruebas E2E y en entornos reales

Estas pruebas se realizarán al final del proyecto, para comprobar que todo el sistema funciona correctamente de manera completa. Se comprobará que la aplicación móvil se integra bien a un servidor remoto donde se alojará la REST API.

Además, se realizarán pruebas en entornos reales, usando robots cortacésped de prueba, para ver que todas las funcionalidades de gestión de incidencias y control remoto básico funcionan como se espera.

Para estas pruebas se realizarán en una pista de pruebas, que simula la instalación de un robot cortacésped, que un cliente podría tener en su jardín.

6.6.2.2 Pruebas de usabilidad

Este tipo de pruebas asegurarán que la aplicación móvil tiene una interfaz sencilla y usable para el tipo de usuarios a los que está orientada la aplicación.

Ya que estas pruebas requieren mucho tiempo y disponibilidad de muchas personas, se realizarán sobre los casos de uso más importantes del sistema. Los cuáles son los siguientes:

- Registro de una empresa y usuario administrador.
- Asignación de un robot a un cliente y empleado.
- Creación de un empleado.

Capítulo 7 DISEÑO DEL SISTEMA DE INFORMACIÓN

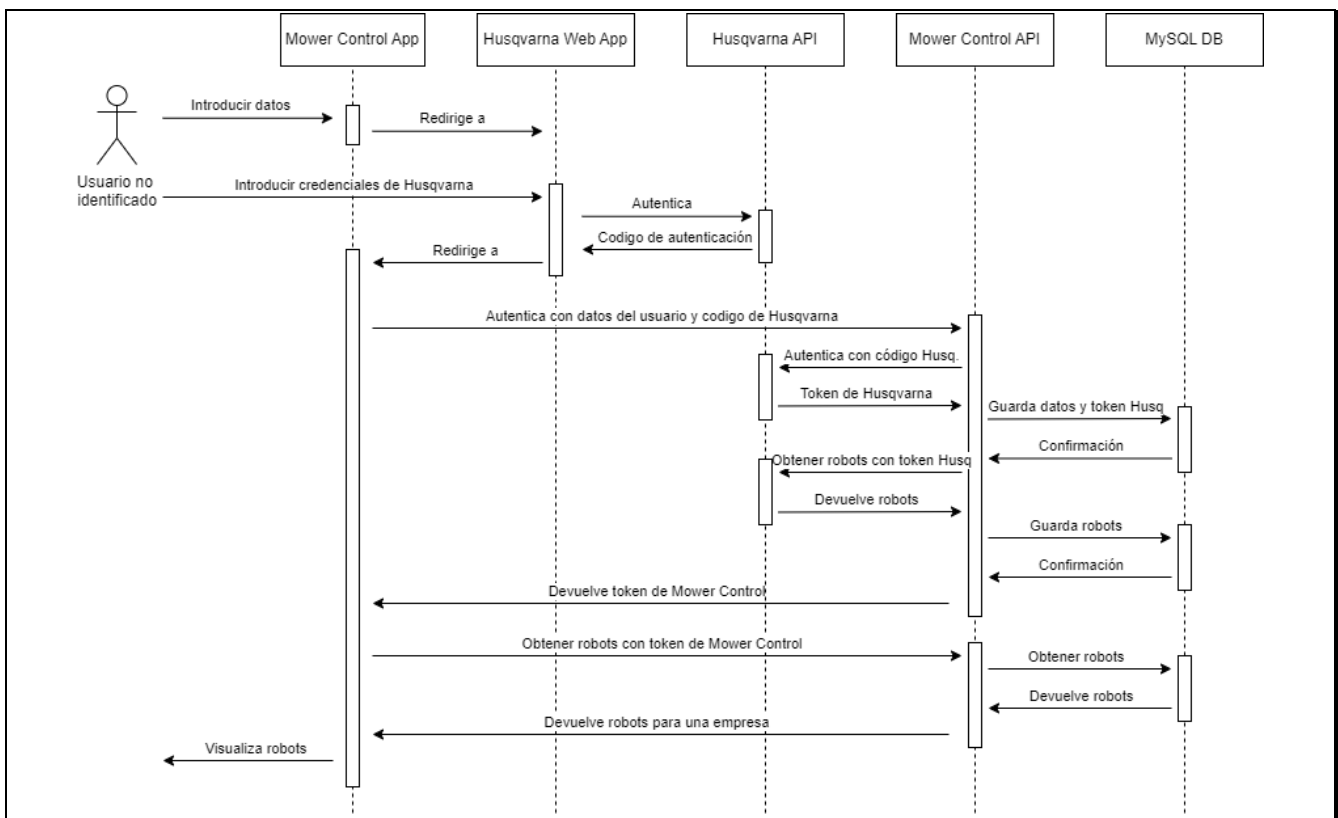
FASE DE DESARROLLO

DSI

7.1 DSI 3: DISEÑO DE CASOS DE USO REALES

En esta sección se presentarán los casos de uso más complejos y relevantes del sistema mediante diagramas de secuencia. El objetivo es describir en detalle el comportamiento que el sistema debe adoptar en cada uno de estos casos y las interacciones entre las diferentes partes del sistema, así como con servicios externos.

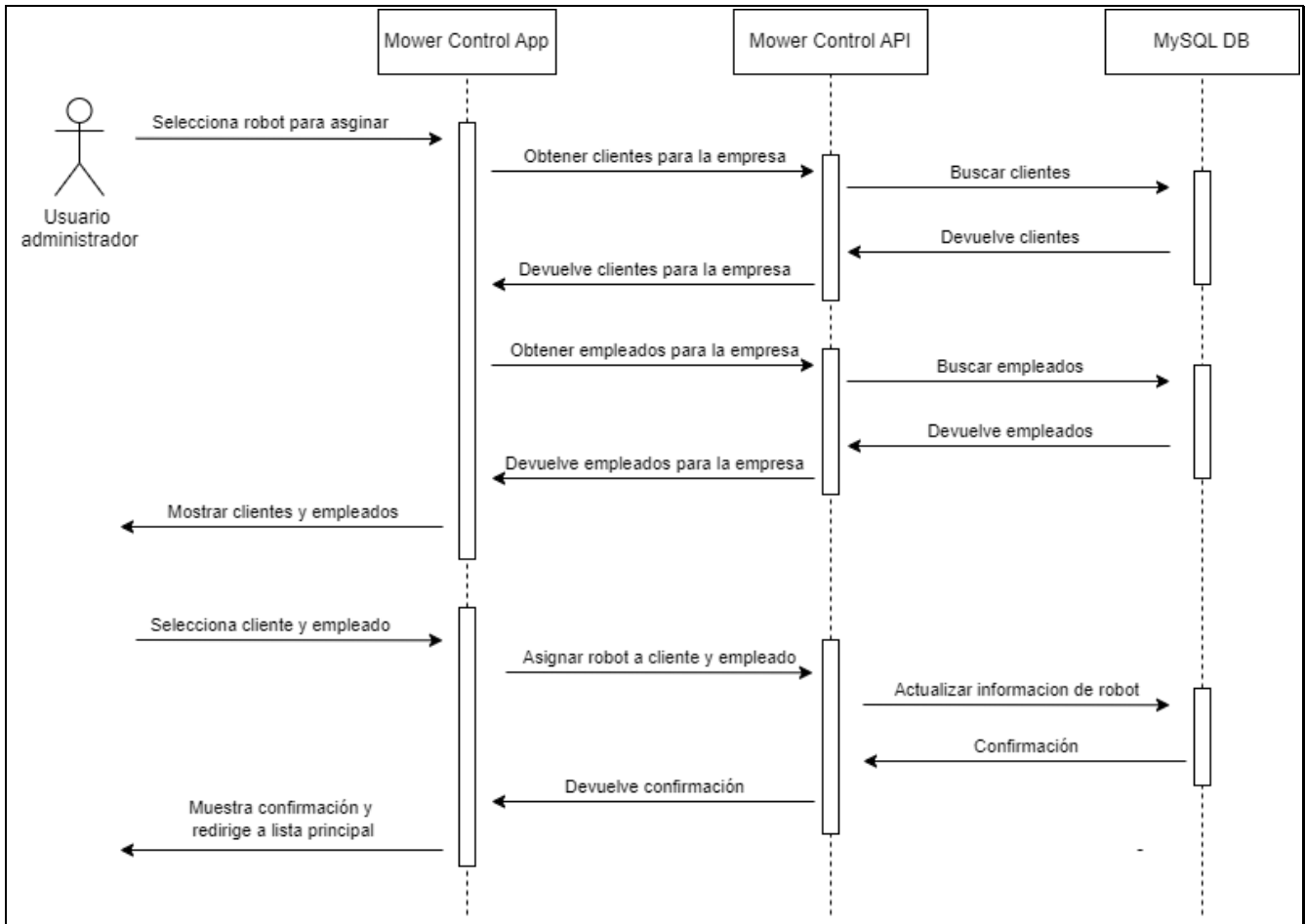
7.1.1 Caso de uso 1: Registro



En este diagrama, se muestran todas las interacciones y llamadas que se realizan en un proceso para registrar una nueva empresa y un usuario administrador.

El usuario no identificado introduce los datos en la aplicación móvil, después redirige a la aplicación web de Husqvarna para introducir sus credenciales. Este paso devuelve un código de autenticación que se manda junto a los demás datos a la REST API. La REST API se autentica contra la API de Husqvarna para obtener el token que después usa para obtener toda la información de los robots. Guarda toda esta información y devuelve a la aplicación móvil el nuevo token generado. La aplicación móvil muestra la vista de robots al usuario después de hacer una nueva llamada a la API para obtener todos los robots de la empresa.

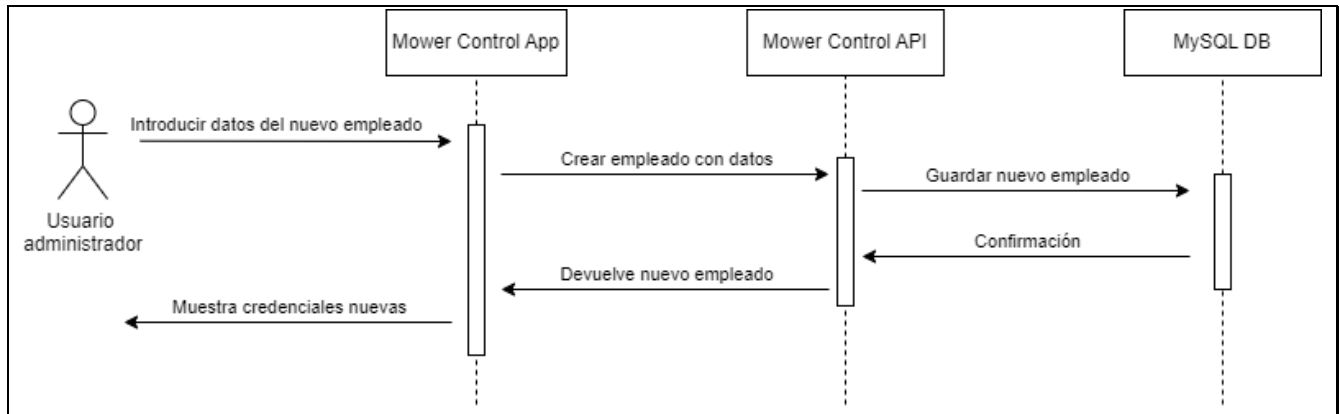
7.1.2 Caso de uso 2: Asignar robot



En el diagrama de arriba, se muestran las interacciones entre las diferentes partes principales del sistema, aplicación móvil, REST API y base de datos, para el caso de uso de asignar un robot.

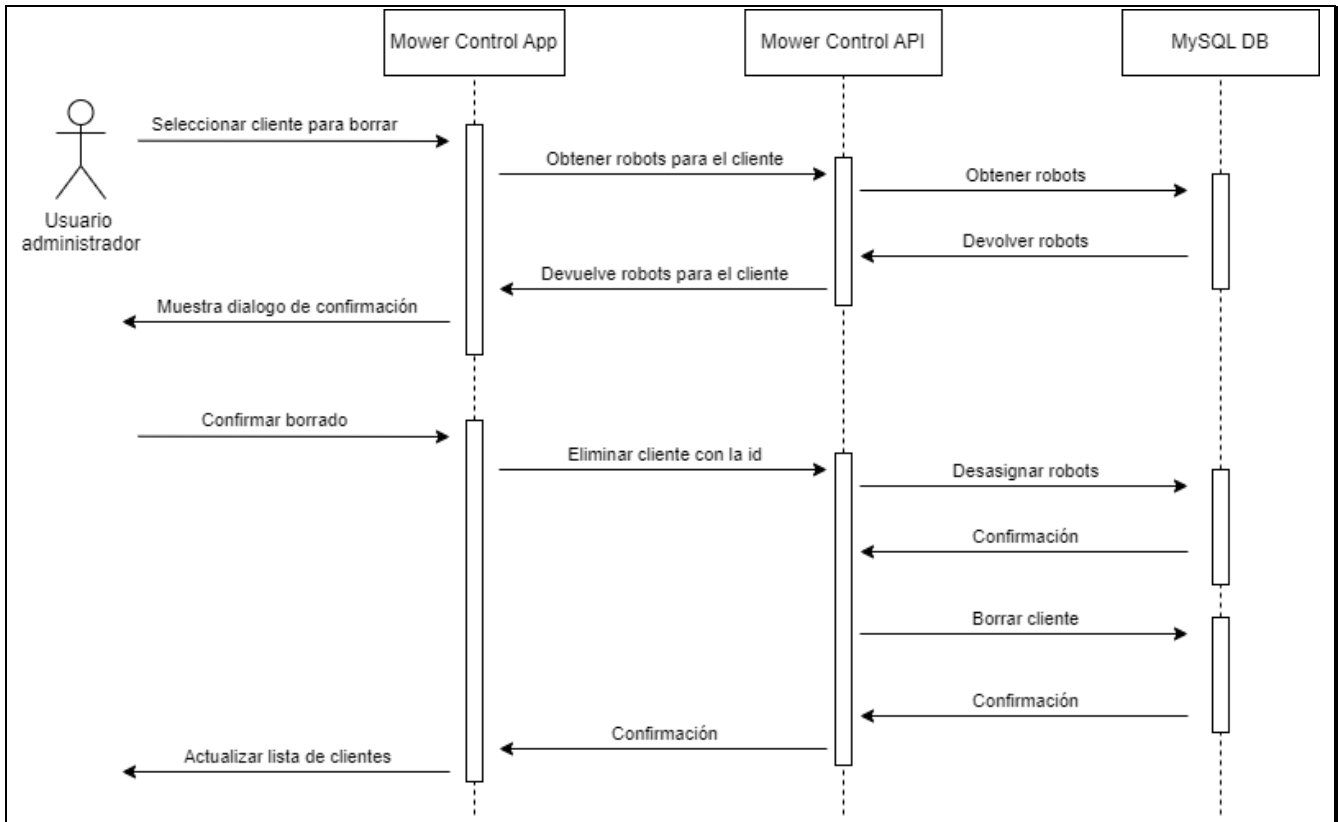
El usuario, en este caso tiene que ser administrador, selecciona un robot para asignar de la lista que le aparece. La aplicación móvil hace dos peticiones para obtener los clientes y empleados de la empresa, para poder mostrárselos al usuario y que pueda elegir el que quiera. Una vez elegidos y el usuario confirma la acción, la aplicación móvil manda todos los datos necesarios a la API que se encarga de actualizar la información para el robot asignado en la base de datos y devolver una confirmación a la aplicación móvil. Por último, cuando la aplicación móvil recibe la confirmación de que todo ha ido bien, se le muestra al usuario y se le redirige a la lista principal de robots, donde ya puede ver el nuevo robot asignado.

7.1.3 Caso de uso 3: Creación de un empleado



Este diagrama, muestra el caso de uso de la creación de un nuevo empleado y las llamadas que se hacen entre las diferentes partes del sistema. El usuario administrador introduce los datos necesarios para crear un nuevo empleado. La aplicación móvil hace una llamada a la API con los datos para crear el empleado, genera las credenciales para el nuevo empleado, guarda toda la información en la base de datos y devuelve a la aplicación móvil el empleado creado. Por último, la aplicación móvil muestra estas credenciales al usuario.

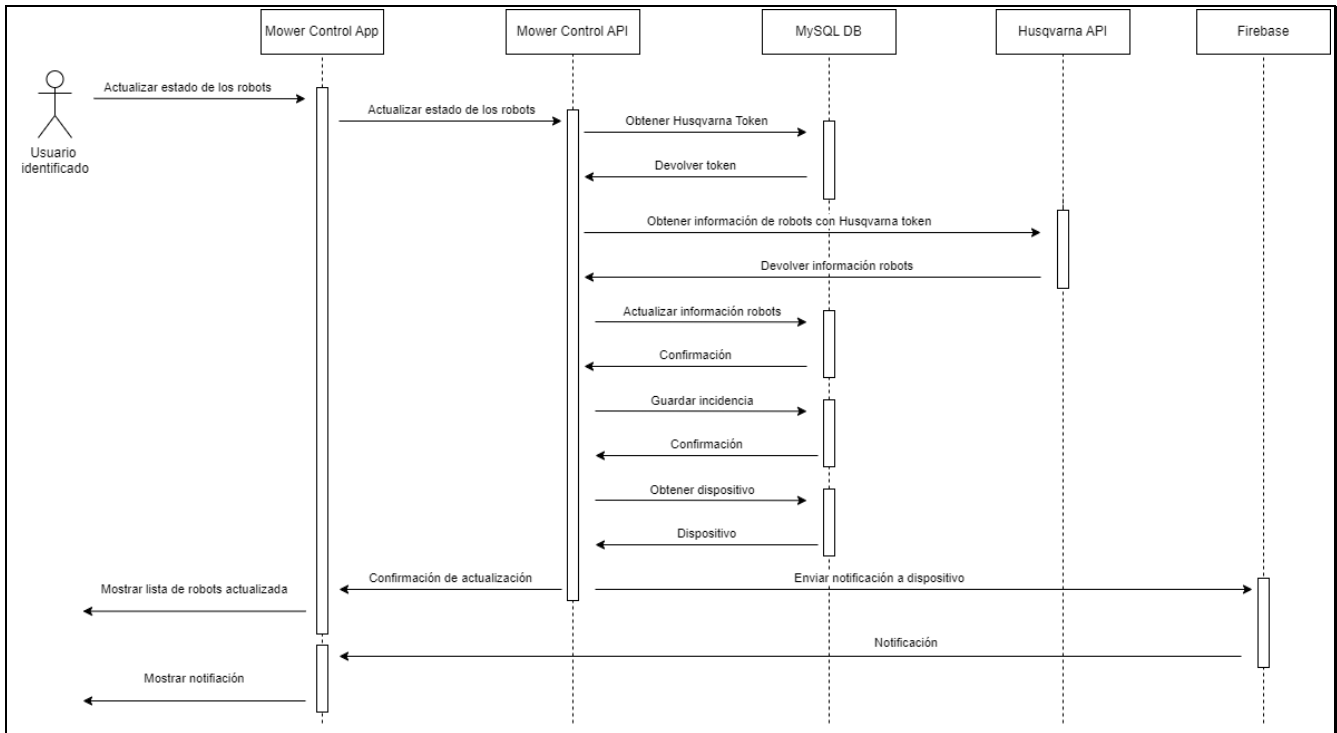
7.1.4 Caso de uso 4: Borrado de un cliente



El diagrama superior muestra las interacciones entre los diferentes componentes del sistema, para el caso de uso de borrado de un cliente.

El usuario administrador selecciona el cliente que quiere borrar, la aplicación móvil hace una petición al servidor para obtener los robots que tiene el cliente asignado. Una vez recibe la respuesta del servidor, muestra un dialogo de confirmación indicando si el cliente tiene robots asignados o no. En caso de que el usuario confirme la acción, la aplicación móvil envía una petición a la API, para borrar el cliente con una determinada ID. En caso de que el cliente tenga robots asignados, se harán dos peticiones a la base de datos. Una para desasignar los robots y otra para borrar el cliente. En caso de que no tenga robots asignados, solo se hará la segunda petición. Una vez hecho esto, el servidor devuelve una confirmación a la aplicación móvil y esta actualiza la lista de clientes.

7.1.5 Caso de uso 5: Generación de incidencia

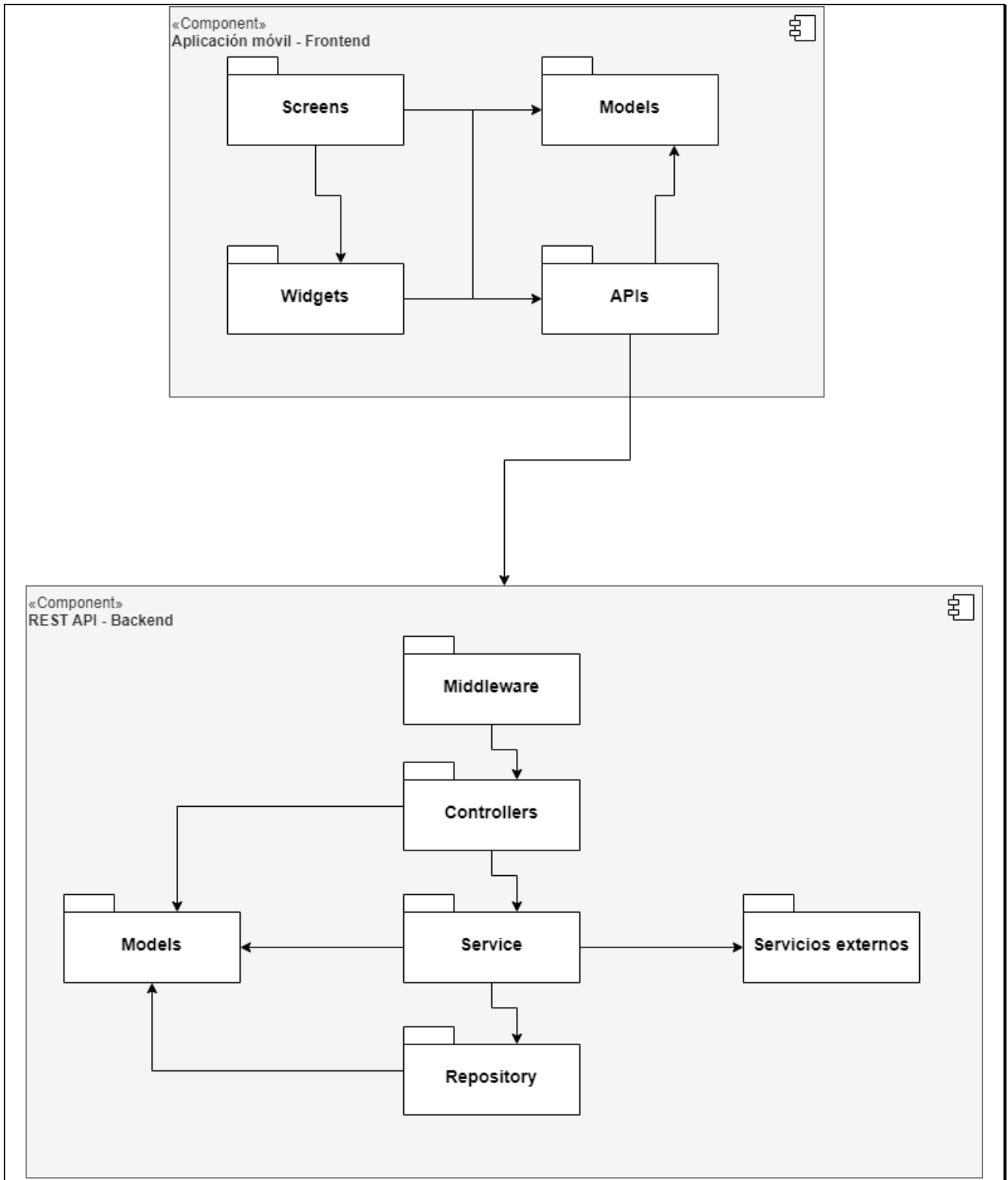


El diagrama superior muestra las interacciones y llamadas que se hacen cuando el usuario solicita una actualización del estado de los robots y se genera una incidencia. Tener en cuenta que esta actualización del estado de los robots también se hace de manera automática cada hora.

El usuario identificado haciendo clic en el botón de actualizar hace que la aplicación móvil, haga una petición a la API para actualizar los robots. La API busca en la base de datos el token para la empresa, el cual después usa para llamar a la API de Husqvarna y obtener información actualizada de todos los robots. Guarda esta información en la base de datos y en caso de que algún robot tenga un estado de error, la API se encargará de generar una incidencia. Guardará la incidencia en la base de datos para que quede registrada y obtendrá la ID del dispositivo del usuario asignado al robot que causa la incidencia. Con esta ID del dispositivo, la API manda una petición al servicio externo Firebase, el cual se encarga después de mandar la notificación al dispositivo indicado. Una vez enviada la petición a Firebase, la API devuelve a la aplicación móvil una confirmación de que la actualización ha ido correctamente, para que pueda actualizar la lista de robots.

7.2 DSI 4: DISEÑO DE CLASES

A continuación, se expone el diagrama de paquetes del sistema. En los siguientes apartados se definirá cada parte por separado para un mayor detalle.





7.2.1 REST API – Backend

7.2.1.1 Controladores

Los controladores tienen la función de gestionar las peticiones que recibe el sistema. Su labor principal es recibir dichas peticiones y proceder a una validación para asegurar que se cumplen con los requisitos necesarios. Una vez validadas, los controladores interactúan con los servicios correspondientes para ejecutar las operaciones requeridas. Finalmente, se encargan de estructurar y enviar la respuesta adecuada al cliente que hizo la petición, asegurando que esta contenga la información solicitada o el resultado de la operación efectuada.

7.2.1.2 Servicios

Los servicios son componentes fundamentales que se encapsulan la lógica de negocio y las operaciones que el sistema debe realizar. Su función es principal es manejar las tareas específicas relacionadas con el procesamiento de datos. Reciben peticiones de los controladores y se comunican con los repositorios para ejecutar acciones sobre el sistema de persistencia.

A parte de los repositorios, los servicios también se comunican con los servicios externos para realizar determinadas operaciones, que requiere el sistema.

7.2.1.3 Repositorios

Los repositorios desempeñan el papel crucial en la interacción con la base de datos. Su función es realizar las operaciones de lectura y escritura necesarias para la gestión de datos. Proporcionan una capa de abstracción y facilitan el mantenimiento y escalabilidad del sistema. Cada repositorio se enfoca en una entidad específica del modelo, por ejemplo el “ClientsRepository” maneja operaciones relacionadas con los clientes, como obtener, crear y eliminar dichos recursos.



7.2.1.4 Middleware

Los middlewares son componentes de software que se sitúan entre la petición del cliente y el controlador de una API, ejecutando diversas operaciones en este trayecto. Son esenciales para añadir capas de funcionalidad y seguridad sin necesidad de modificar el código de controlador, permitiendo así una arquitectura más modular y mantenible.

En este sistema, se implemente un middleware, responsable de verificar la identidad del usuario que realizar la petición y determinar si posee los permisos necesarios para ejecutarla.

7.2.1.5 Servicios externos

Los servicios externos son funcionalidades proporcionadas por un tercero a los que se accede mediante peticiones a su REST API.

Este sistema usa dos servicios externos, el primero y pieza angular del sistema, es la API del proveedor de robots Husqvarna. De este servicio podemos obtener toda la información de los robots de una cuenta. El sistema desarrollado extrae de esta API la información en tiempo real de los robots y la almacena en la base de datos.

El segundo servicio, proporcionado por Firebase, se utiliza para el envío de notificaciones cuando se genera una incidencia. Este sistema externo nos permite enviar notificaciones a cualquier dispositivo de manera sencilla y sin necesidad de implementar lógica complicada.



7.2.2 Aplicación móvil – Frontend

7.2.3 Pantallas

Las pantallas (screens) son las vistas de la aplicación móvil. Estas pantallas también son widgets, pero que actúan como la vista completa de la aplicación. Están formadas por componentes, también llamados widgets, para obtener un diseño similar a los que se presentaron en el apartado [6.5.1 Descripción de la Interfaz](#).

Las pantallas utilizan el paquete de APIs, el cual se encarga de realizar las llamadas al backend para obtener toda la información del sistema.

7.2.4 Widgets

El paquete de widgets está formado por pequeños componentes, que se utilizan en las pantallas. Se encuentran en este paquete por diversas razones, la principal es poder reutilizar el mismo componente en diversas pantallas y también extraer el código de los widgets de pantallas, para que sea mucho más pequeño y legible. Esto último facilita el mantenimiento y extensión del sistema.

Al igual que las pantallas, algunos de estos widgets hacen uso del paquete de APIs para obtener información de la REST API.

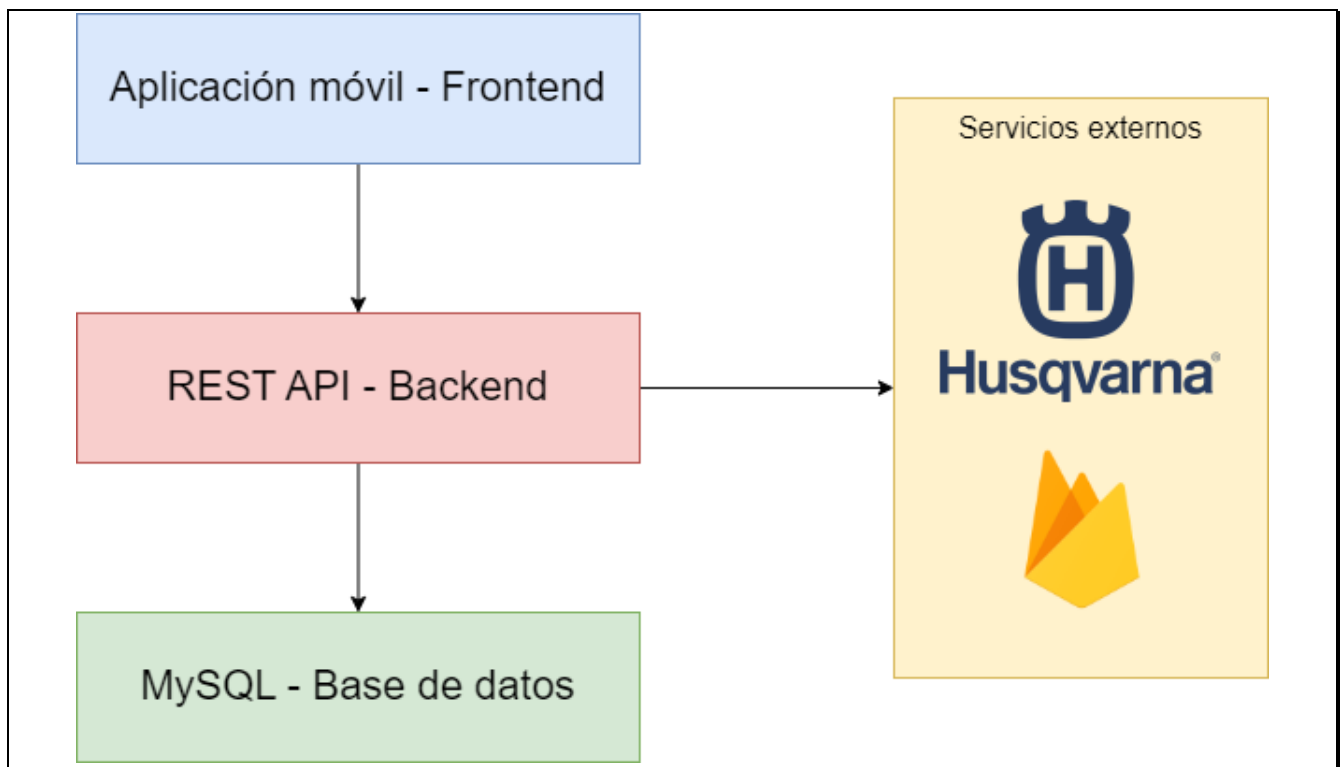
7.3 DSI 5: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

7.3.1 DSI 5.1 Diseño de Módulos del Sistema

7.3.1.1 Vista general

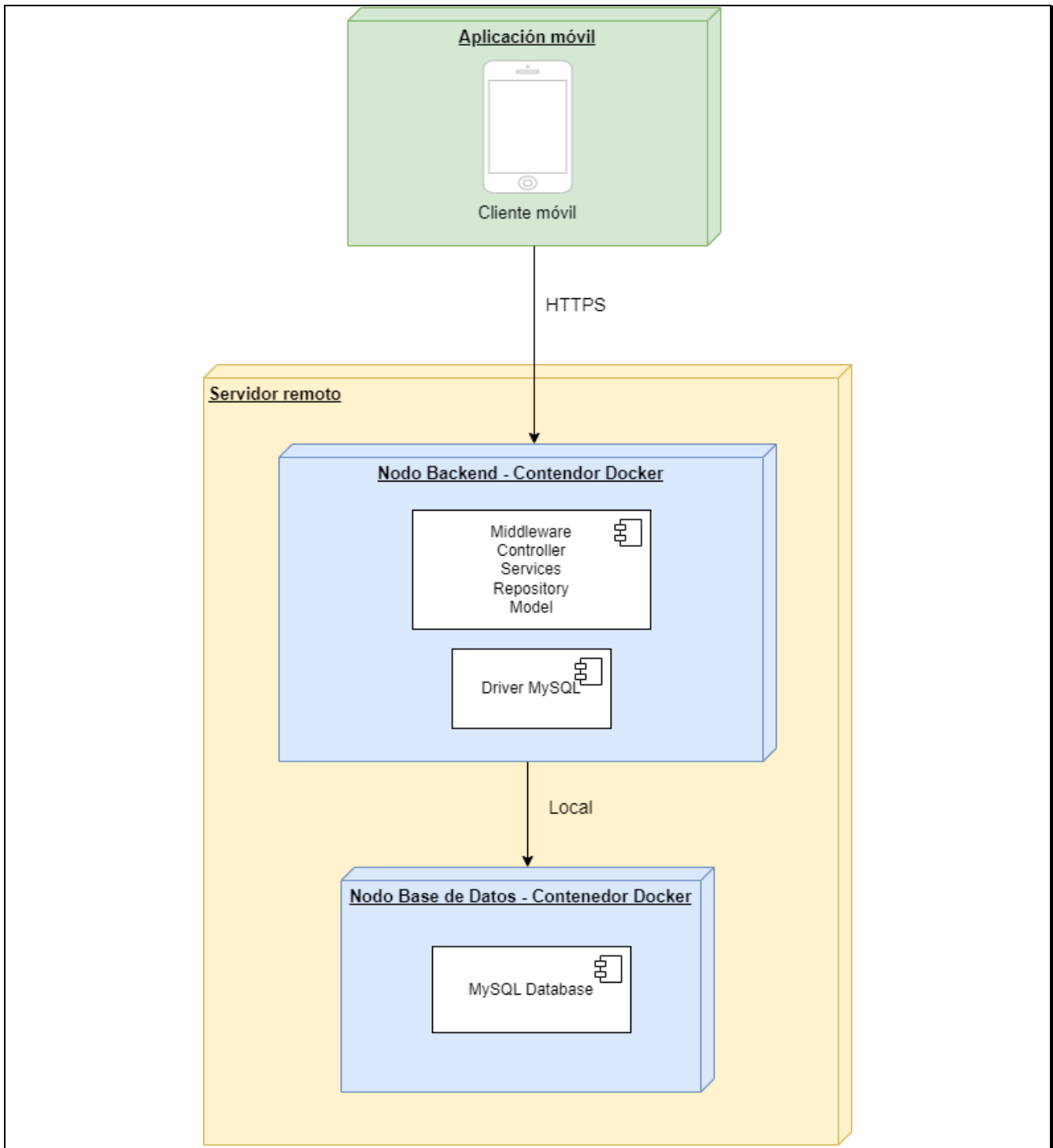
El componente central del sistema es el backend, que se encarga de interactuar tanto con la base de datos para efectuar consultas y modificaciones, como con los servicios externos para funciones adicionales. La aplicación móvil accede a los datos y lleva a cabo operaciones mediante la comunicación con el backend.

A continuación, se expone un diagrama con la vista general del sistema:



7.3.1.2 Diagrama de despliegue

A continuación, se muestra el diagrama de despliegue del sistema completo:



El sistema está formado por dos nodos, que serán contenedores Docker, los cuáles se desplegarán sobre un servidor remoto para tener acceso desde cualquier dispositivo. La aplicación móvil se instalará en los dispositivos donde se vaya a utilizar, por ejemplo con un archivo .apk.

7.4 DSI 6: DISEÑO FÍSICO DE DATOS

7.4.1 Descripción del SGBD Usado

En este sistema, como ya se ha indicado en el apartado [3.2.1 Base de datos](#), se ha optado por utilizar una base de datos MySQL como Sistema de Gestión de Bases de Datos (SGBD). MySQL es una de las bases de datos más populares debido a su robustez, flexibilidad y facilidad de uso. Además, su amplia comunidad de usuarios y desarrolladores garantiza un soporte constante y actualizado.

7.4.1.1 Plan de Copias de Seguridad

El plan de copias de seguridad diseñado para este sistema es sencillo pero efectivo, adecuado para la naturaleza y el tamaño de los datos que se manejarán. Dado que no se espera que los datos alcancen un volumen considerable, la estrategia de copias de seguridad se basa en la simplicidad y la eficiencia.

7.4.1.1.1 Política de Copias de Seguridad

Frecuencia: Se realizará una copia de seguridad completa de la base de datos una vez por semana. Esta periodicidad es suficiente para asegurar la integridad de los datos y minimizar la pérdida de información en caso de algún fallo.

Medios de Almacenamiento: Las copias de seguridad se almacenarán en la nube, cualquier proveedor como GCP (Bucket), Azure (Blob Storage) o AWS (S3). Este medio de almacenamiento garantiza una rápida recuperación de los datos en caso de necesidad y una disponibilidad muy alta.

Formato: Las copias de seguridad se realizarán en formato SQL, lo que facilita su restauración en cualquier sistema MySQL y permite una manipulación sencilla si fuese necesario.

7.4.1.1.2 Procedimiento de Realización de Copias:

Se utilizará la herramienta `mysqldump` para generar las copias de seguridad. Este comando permite exportar la base de datos completa a un archivo `.sql`, que puede ser fácilmente importado en el futuro.

El proceso será automatizado mediante un script que se ejecutará semanalmente a través de un cron job en el servidor, eliminando la necesidad de intervención manual y asegurando que las copias de seguridad se realicen puntualmente.



7.4.1.2 Restablecimiento de los Datos en Caso de Pérdida

El restablecimiento de los datos se alinea con el plan de copias de seguridad diseñado. En caso de pérdida de datos, el proceso de restauración será manual, siguiendo los siguientes pasos:

- **Identificación del Backup Más Reciente:** Se seleccionará la copia de seguridad más reciente almacenada en la nube.
- **Importación de la Copia de Seguridad:** Se utilizará el comando `mysql` para importar el archivo `.sql` generado por `mysqldump`. Este comando permite restaurar rápidamente toda la estructura y los datos de la base de datos. El proceso de importación se realizará mediante un script de restauración, que garantizará que todos los pasos se sigan de manera correcta y ordenada.
- **Verificación de la Integridad de los Datos:** Una vez completada la restauración, se llevarán a cabo pruebas para verificar que todos los datos se han restaurado correctamente y que la base de datos está operativa.

7.4.1.3 Política de Purgado de Datos

No se implementará una política de purgado de datos. Dado el bajo volumen de datos esperado, no se prevé que la base de datos alcance un tamaño que pueda comprometer el rendimiento del sistema. Por lo tanto, se mantiene todo el historial de datos sin necesidad de eliminación periódica o archivado.

7.4.1.4 Mantenimiento de los Datos

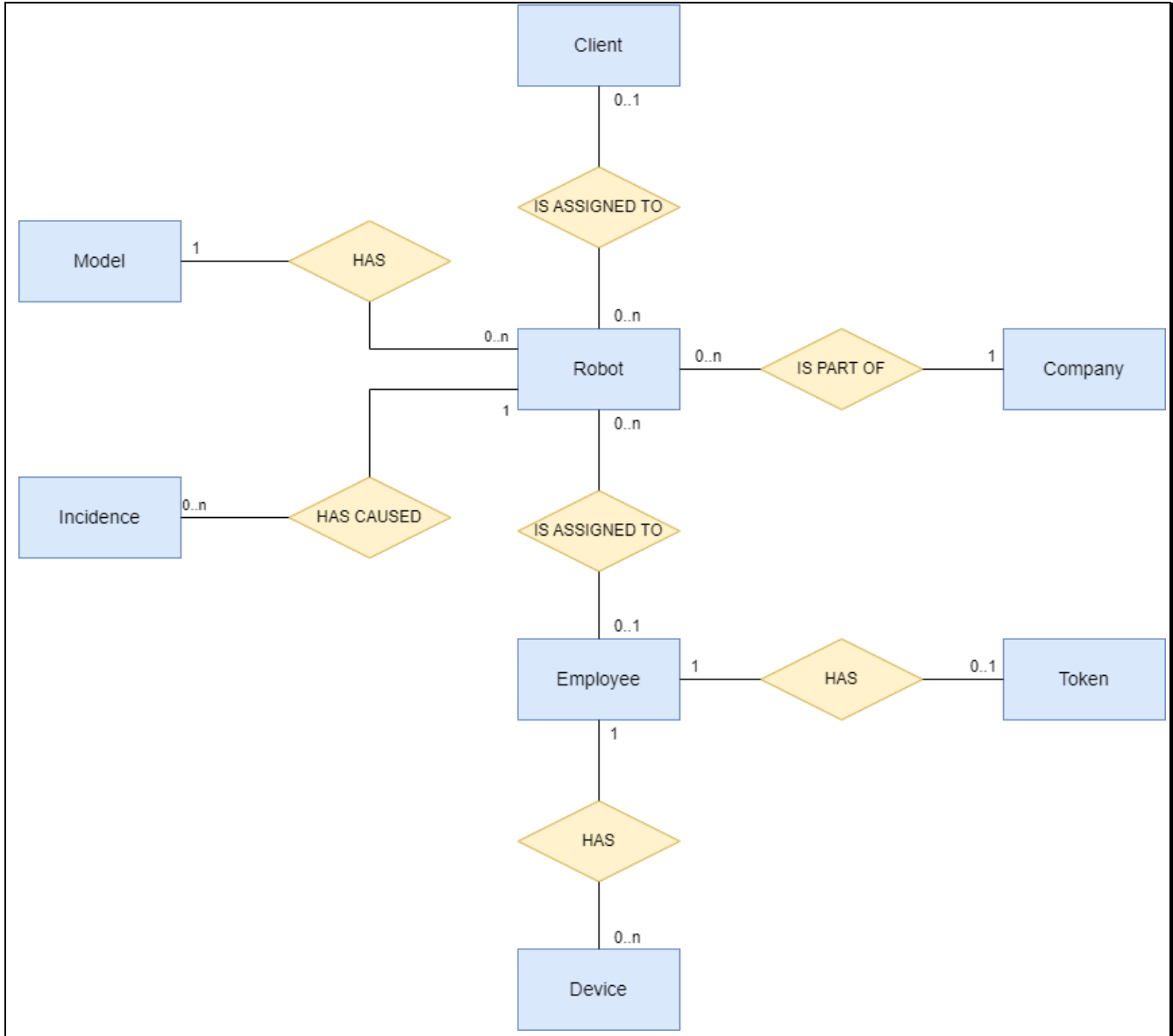
En este sistema no se considera necesario realizar operaciones periódicas de mantenimiento de datos, tales como la limpieza de datos redundantes o la eliminación de duplicados. La estructura y el manejo de los datos están diseñados para evitar la acumulación de información innecesaria, garantizando así que la base de datos se mantenga eficiente y operativa sin necesidad de intervenciones adicionales.

7.4.2 Integración del SGBD en Nuestro Sistema

Para enlazar el backend con la base de datos MySQL, se ha empleado la biblioteca `mysql2` de Node.js. Esta biblioteca proporciona capacidades para establecer conexiones, realizar consultas y gestionar resultados de manera asíncrona.

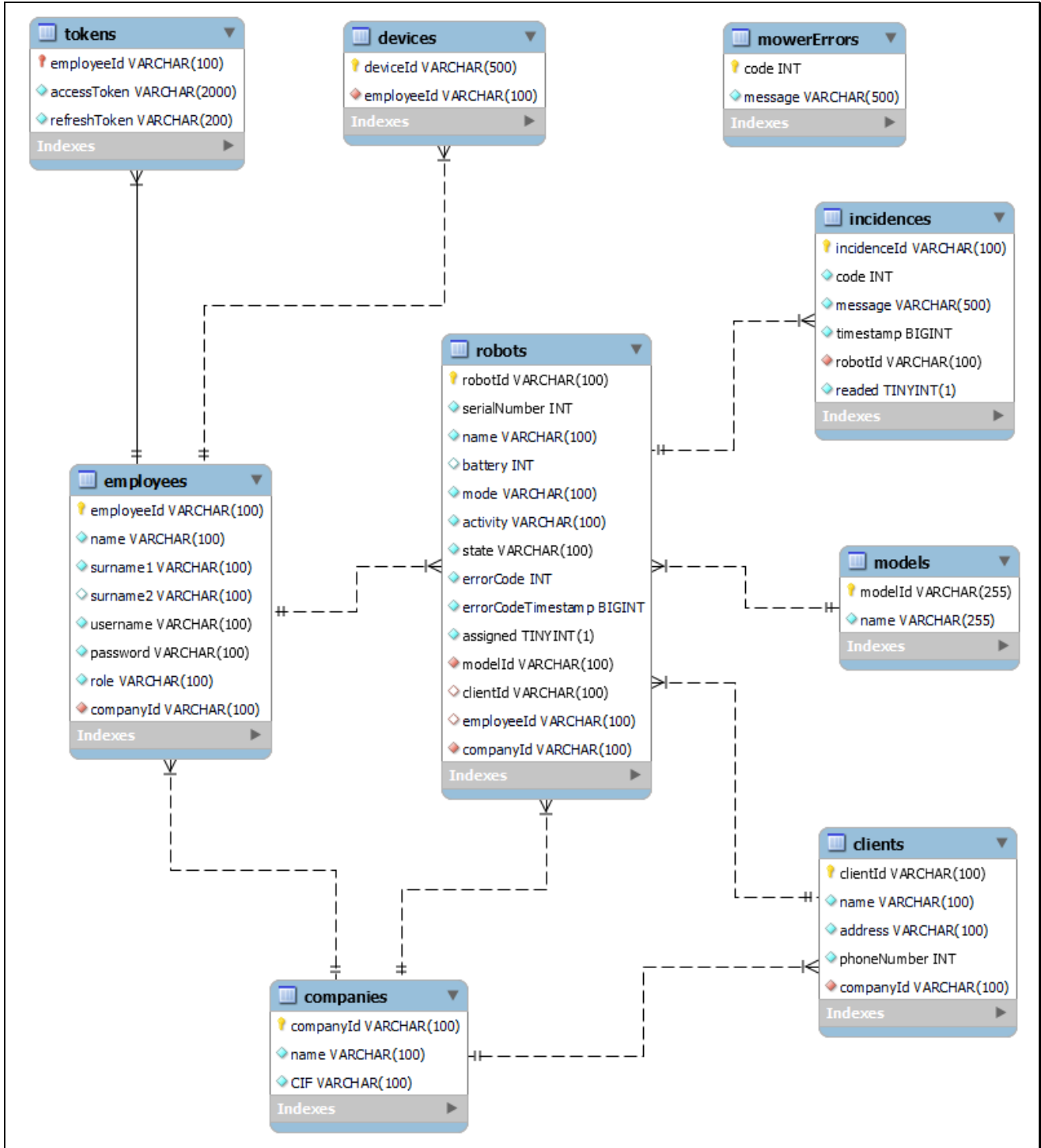
7.4.3 Diagrama E-R

En este apartado, se expone el diagrama Entidad – Relación que representa el modelo de datos utilizado en el SGBD del sistema:



7.4.4 Diagrama relacional

A continuación, se presenta el diagrama relacional que corresponde al diagrama Entidad – Relación previamente mencionado, incluyendo las tablas respectivas:





7.5 DSI 10: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS

En este apartado, se detallan las pruebas que se implementarán para asegurar el correcto funcionamiento del sistema.

7.5.1 Pruebas Unitarias

Anteriormente explicado, en el apartado [6.6.1.1 Pruebas Unitarias](#), la REST API será el sistema que más al completo se va a probar ya que contiene toda la lógica del sistema así como el manejo de todos los datos.

A continuación, se detallarán las pruebas unitarias que se realizarán sobre todos los componentes del subsistema backend.

7.5.1.1 Entorno

Las pruebas unitarias se harán usando el framework Jest, ya que nos permite una fácil implementación de estas. Además, nos proporciona una funcionalidad interesante, poder simular el comportamiento de otras partes del sistema, también llamados *mocks*. Esto nos permite aislar cada componente y probarlo sin depender de otras partes del sistema.

Todas las pruebas realizadas se ejecutarán usando las GitHub Actions. Las pruebas serán lanzadas cada vez que se realice un commit en el repositorio.

7.5.1.2 Diseño técnico de las pruebas

A continuación, se definirán todas las pruebas del backend, divididas por cada clase que forma el sistema.

7.5.1.2.1 Clients Controller

7.5.1.2.1.1 Get Clients By Company

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|-------------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de clientes y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |



7.5.1.2.1.2 Create Client

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------|
| 1 | Válida | El servicio funciona correctamente | Cliente creado y 201 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.1.3 Delete Client

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|--------------------|
| 1 | Válida | El servicio funciona correctamente | Cuerpo vacío y 204 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.2 Companies Controller

7.5.1.2.2.1 Get Companies

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|-------------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de empresas y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.2.2 Get Company By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|------------------|
| 1 | Válida | El servicio funciona correctamente | Empresa y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.3 Employees Controller

7.5.1.2.3.1 Get Employees By Company

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|--------------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de empleados y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.3.2 Create Employee

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|------------------|
| 1 | Válida | El servicio funciona correctamente | Empleado y 201 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.3.3 Delete Employee

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|--------------------|
| 1 | Válida | El servicio funciona correctamente | Cuerpo vacío y 204 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.4 Incidences Controller

7.5.1.2.4.1 Update Readed Status

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|------------------|
| 1 | Válida | El servicio funciona correctamente | Mensaje y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |



7.5.1.2.4.2 Get Incidences By Company

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de incidencias y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.4.3 Get Incidences By Robot

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de incidencias y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.5 Robots Controller

7.5.1.2.5.1 Get Robots By Company

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|-----------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de robots y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.5.2 Get Robots By Employee

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|-----------------------|
| 1 | Válida | El servicio funciona correctamente | Lista de robots y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.5.3 Assign Robot to Client and Employee

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------|
| 1 | Válida | El servicio funciona correctamente | Robot asignado y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |



7.5.1.2.6 Login Controller

7.5.1.2.6.1 Sign Up

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------|
| 1 | Válida | El servicio funciona correctamente | Token generado y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.6.2 Log In

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|----------------------|
| 1 | Válida | El servicio funciona correctamente | Token generado y 200 |
| 2 | Válida | El servicio devuelve un error | Propaga el error |
| 3 | No válida | El servicio funciona correctamente | Propaga el error |

7.5.1.2.7 Clients Repository

7.5.1.2.7.1 Get By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|--------------------|
| 1 | Válida | El cliente existe en la base de datos | Cliente encontrado |
| 2 | Válida | El cliente no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.7.2 Get By Company Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|-------------------|
| 1 | Válida | Hay clientes para la empresa | Lista de clientes |
| 2 | Válida | No hay clientes para la empresa | Lista vacía |
| 3 | Válida | No existe la empresa | Lista vacía |
| 4 | No válida | - | Lanza un error |
| 5 | Válida | La base de datos lanza un error | Propaga el error |



7.5.1.2.7.3 Create

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|------------------|
| 1 | Válida | - | Cliente creado |
| 2 | No válida | - | Lanza un error |
| 3 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.7.4 Remove

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|------------------|
| 1 | Válida | El cliente existe en la base de datos | - |
| 2 | Válida | El cliente no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.8 Companies Repository

7.5.1.2.8.1 Get All

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|-------------------|
| 1 | Válida | - | Lista de empresas |
| 2 | No válida | - | Lanza un error |
| 3 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.8.2 Get By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|--------------------|
| 1 | Válida | La empresa existe en la base de datos | Empresa encontrada |
| 2 | Válida | La empresa no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |



7.5.1.2.8.3 Create

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|------------------|
| 1 | Válida | No existe una con el mismo CIF | Cliente creado |
| 2 | Válida | Existe una con el mismo CIF | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.8.4 Remove

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|------------------|
| 1 | Válida | La empresa existe en la base de datos | - |
| 2 | Válida | La empresa no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.9 Employees Repository

7.5.1.2.9.1 Get By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---|---------------------|
| 1 | Válida | El empleado existe en la base de datos | Empleado encontrado |
| 2 | Válida | El empleado no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.9.2 Get By Username

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---|---------------------|
| 1 | Válida | El empleado existe en la base de datos | Empleado encontrado |
| 2 | Válida | El empleado no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |



7.5.1.2.9.3 Get By Company Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|----------------------------------|--------------------|
| 1 | Válida | Hay empleados para la empresa | Lista de empleados |
| 2 | Válida | No hay empleados para la empresa | Lista vacía |
| 3 | Válida | No existe la empresa | Lista vacía |
| 4 | No válida | - | Lanza un error |
| 5 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.9.4 Create

| ID | Entrada | Notas | Salida esperada |
|----|-----------|------------------------------------|------------------|
| 1 | Válida | No existe uno con el mismo usuario | Empleado creado |
| 2 | Válida | Existe uno con el mismo usuario | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.9.5 Remove

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---|------------------|
| 1 | Válida | El empleado existe en la base de datos | - |
| 2 | Válida | El empleado no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.10 Incidences Repository

7.5.1.2.10.1 Get By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---|-----------------------|
| 1 | Válida | La incidencia existe en la base de datos | Incidencia encontrada |
| 2 | Válida | La incidencia no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.10.2 Get By Robot Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|----------------------|
| 1 | Válida | - | Lista de incidencias |
| 2 | Válida | El robot no existe en la base de datos | Lista vacía |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.10.3 Create

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|------------------|
| 1 | Válida | - | Cliente creado |
| 2 | No válida | - | Lanza un error |
| 3 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.10.4 Update

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|----------------------------------|
| 1 | Válida | La incidencia no está leída | Incidencia marcada como leída |
| 2 | Válida | La incidencia está leída | Incidencia marcada como no leída |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.10.5 Get By Company Id

| ID | Entrada | Notas | Salida esperada |
|----|---------------------------|---------------------------------|--------------------------------|
| 1 | Válida | - | Lista de incidencias |
| 2 | Válida con readed = true | - | Lista de incidencias leídas |
| 3 | Válida con readed = false | - | Lista de incidencias no leídas |
| 4 | Válida | No existe la empresa | Lista vacía |
| 5 | No válida | - | Lanza un error |
| 6 | Válida | La base de datos lanza un error | Propaga el error |



7.5.1.2.11 Model Repository

7.5.1.2.11.1 Get By Name

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---|-------------------|
| 1 | Válida | El modelo existe en la base de datos | Modelo encontrado |
| 2 | Válida | El modelo no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12 Robots Repository

7.5.1.2.12.1 Get By Id

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|------------------|
| 1 | Válida | El robot existe en la base de datos | Robot encontrado |
| 2 | Válida | El robot no existe en la base de datos | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.2 Get By Id or Serial Number

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|------------------|
| 1 | Válida | El robot existe en la base de datos | Robot encontrado |
| 2 | Válida | El robot no existe en la base de datos | Lanza un error |
| 3 | Válida | Existe más de un robot | Lanza un error |
| 4 | No válida | - | Lanza un error |
| 5 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.3 Get By Company Id

| ID | Entrada | Notas | Salida esperada |
|----|--------------------------------|---------------------------------|---------------------------------|
| 1 | Válida | - | Lista de robots |
| 2 | Válida con assigned = true | - | Lista de robots asignados |
| 3 | Válida con assigned = false | - | Lista de robots no asignados |
| 4 | No válida | - | Lanza un error |
| 5 | Válida | La base de datos lanza un error | Propaga el error |



7.5.1.2.12.4 Get By Employee

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|------------------|
| 1 | Válida | - | Lista de robots |
| 2 | Válida | No existe el empleado | Lista vacía |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.5 Create

| ID | Entrada | Notas | Salida esperada |
|----|-----------|--|------------------|
| 1 | Válida | - | Cliente creado |
| 2 | Válida | Existe un robot con el mismo número de serie | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.6 Assing Robot

| ID | Entrada | Notas | Salida esperada |
|----|-----------|---------------------------------|------------------|
| 1 | Válida | - | Robot asignado |
| 2 | Válida | No existe el robot | Lanza un error |
| 3 | No válida | - | Lanza un error |
| 4 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.7 Unassing Robot For Client

| ID | Entrada | Notas | Salida esperada |
|----|---------|---------------------------------|-------------------|
| 1 | Válida | - | Robot desasignado |
| 2 | Válida | No existe el robot | Lanza un error |
| 3 | Válida | La base de datos lanza un error | Propaga el error |

7.5.1.2.12.8 Unassing Robot For Employee

| ID | Entrada | Notas | Salida esperada |
|----|---------|---------------------------------|-------------------|
| 1 | Válida | - | Robot desasignado |
| 2 | Válida | No existe el robot | Lanza un error |
| 3 | Válida | La base de datos lanza un error | Propaga el error |



7.5.2 Pruebas de Seguridad

Las pruebas de seguridad comprobarán que los usuarios con determinado rol pueden acceder a los recursos que se espera. Los usuarios administradores pueden realizar más funciones que un usuario estándar, así como usuarios de una empresa no pueden acceder a información de otra.

7.5.2.1 Entorno

Las pruebas de seguridad se harán usando el framework Jest, ya que nos permite una fácil implementación de estas. La aplicación se levantará de manera completa, para así tener en cuenta todas las diferentes capas por donde viaja la petición.

Todas las pruebas realizadas se ejecutarán usando las GitHub Actions. Las pruebas serán lanzadas cada vez que se realice un commit en el repositorio.

7.5.2.2 Diseño técnico de las pruebas

A continuación, se definirán todas las pruebas de seguridad. La mayoría de las pruebas se realizarán sobre el subsistema backend, ya que maneja toda la lógica de usuarios y permisos, pero también habrá alguna prueba que se realizará sobre el frontend, para comprobar que un usuario ve determinadas vistas.

7.5.2.2.1 Backend

| ID | Descripción | Salida esperada |
|----|---|------------------------|
| 1 | Un usuario no identificado intenta realizar una búsqueda | Se deniega la petición |
| 2 | Un usuario no administrador intenta crear un cliente | Se deniega la petición |
| 3 | Un usuario administrador intenta crear un cliente | Se admite la petición |
| 4 | Un usuario no administrador intenta buscar clientes | Se deniega la petición |
| 5 | Un usuario administrador intenta buscar clientes | Se admite la petición |
| 6 | Un usuario administrador intenta buscar clientes para otra empresa | Se deniega la petición |
| 7 | Un usuario no administrador intenta asignar un robot | Se deniega la petición |
| 8 | Un usuario administrador intenta asignar un robot | Se admite la petición |
| 9 | Un usuario no administrador intenta buscar todos los robots para la empresa | Se deniega la petición |
| 10 | Un usuario administrador intenta buscar todos los robots para la empresa | Se admite la petición |

7.5.2.2.2 Frontend

| ID | Descripción | Salida esperada |
|----|---|--|
| 1 | Un usuario administrador accede a la aplicación | Se visualizan 5 pestañas en la barra de navegación |
| 2 | Un usuario no administrador accede a la aplicación | Se visualizan 3 pestañas en la barra de navegación |
| 3 | Un usuario administrador accede a la vista de robots | Se visualiza el botón de asignar un nuevo robot |
| 4 | Un usuario no administrador accede a la vista de robots | No se visualiza el botón de asignar un nuevo robot |

7.5.3 Pruebas de Integración

Todas las pruebas de integración se harán en el subsistema backend, ya que es el que maneja toda la lógica de la aplicación. Las pruebas de integración entre la aplicación móvil y la REST API, tanto como pruebas del sistema completo se definen en el apartado de Pruebas E2E y en Entornos Reales

7.5.3.1 Entorno

Las pruebas de integración se harán usando el framework Jest, ya que nos permite una fácil implementación de estas. Además, se usará la tecnología Test Containers, la cual nos permite simular una base de datos MySQL en la propia ejecución de los tests. Dicha base de datos se encontrará en el mismo estado al inicio de cada prueba, para así no tener interferencias con una prueba anterior o posterior.

Todas las pruebas realizadas se ejecutarán usando las GitHub Actions. Las pruebas serán lanzadas cada vez que se realice un commit en el repositorio.

7.5.3.2 Diseño técnico de las pruebas

A continuación, se definirán todas las pruebas del sistema backend, divididas por recurso que modifican o funcionalidad que realizan. Se omiten algunas pruebas que son solo de consulta, en este diseño, ya que se prueban también en otros niveles.

7.5.3.2.1 Clientes y empleados

Los siguientes casos de pruebas sirven para tanto clientes como empleados, ya que su manejo es el mismo en ambos recursos.

| ID | Descripción | Entrada | Salida esperada |
|----|--|-----------|-------------------------|
| 1 | Creación de un cliente | Válida | Cliente y 201 |
| 2 | Creación de un cliente | No válida | Mensaje de error y 400 |
| 3 | Borrado de un cliente | Válida | Cuerpo vacío y 204 |
| 4 | Borrado de un cliente no existente | Válida | Mensaje de error y 403 |
| 6 | Obtener clientes para una empresa | Válida | Lista de clientes y 200 |
| 7 | Obtener clientes para una empresa no existente | Válida | Mensaje de error y 403 |

7.5.3.2.2 Asignación de un robot a un cliente y empleado

| ID | Descripción | Entrada | Salida esperada |
|----|---|-----------|------------------------|
| 1 | Asignación de un robot | Válida | Robot asignado y 200 |
| 2 | Asignación de un robot con cliente no válido | No válida | Mensaje de error y 400 |
| 3 | Asignación de un robot con empleado no válido | No válida | Mensaje de error y 400 |

7.5.3.2.3 Incidencias

| ID | Descripción | Entrada | Salida esperada |
|----|--|---------|-----------------------------|
| 1 | Marcar como leída una incidencia | Válida | Marcada como leída y 200 |
| 2 | Marcar como no leída una incidencia | Válida | Marcada como no leída y 200 |
| 3 | Marcar como leída una incidencia que no existe | Válida | Mensaje de error y 404 |

7.5.4 Pruebas E2E y en entornos reales

Las pruebas E2E y en entornos reales son pruebas que comprueban el correcto funcionamiento del sistema completo, así como la integración entre la aplicación móvil y las REST API.

7.5.4.1 Entorno

Estas pruebas serán manuales y se realizarán sobre la aplicación móvil directamente instalada en un dispositivo móvil y la REST API desplegada en un servidor remoto.

Además, se proporcionará un entorno real, con un robot instalado en una pista de pruebas, para poder así realizar todas las pruebas necesarias con el robot real.

7.5.4.2 Diseño

Al ser pruebas manuales, estas pruebas se centrarán exclusivamente en las pruebas que requieren una conexión con el robot o casos de uso que no han podido ser probados en niveles más bajos de prueba.

A continuación, se describen dichas pruebas:

7.5.4.2.1 Control remoto

| ID | Descripción | Estado inicial | Estado final |
|----|------------------------|---|---|
| 1 | Poner a segar un robot | El robot se encuentra en la base de carga y el usuario hace clic en el botón de segar | El robot empieza a segar y el estado cambia a SEGANDO |
| 2 | Poner a segar un robot | El robot se encuentra parado y el usuario hace clic en el botón de segar | El robot empieza a segar y el estado cambia a SEGANDO |
| 3 | Poner a segar un robot | El robot se encuentra segando y el usuario hace clic en el botón de segar | El robot sigue segando |
| 4 | Parar un robot | El robot se encuentra en la base de carga y el usuario hace clic en el botón de parar | El robot sigue en la base de carga |
| 5 | Parar un robot | El robot se encuentra segando y el usuario hace clic en el botón de parar | El robot se para y el estado cambia a PAUSADO |



| | | | |
|---|-----------------|---|--|
| 6 | Parar un robot | El robot se encuentra parado y el usuario hace clic en el botón de parar | El robot sigue parado |
| 7 | Cargar un robot | El robot se encuentra parada y el usuario hace clic en el botón de cargar | El robot se dirige a la base de carga y el estado cambia a VOLVIENDO |
| 8 | Cargar un robot | El robot se encuentra segando y el usuario hace clic en el botón de cargar | El robot se dirige a la base de carga y el estado cambia a VOLVIENDO |
| 9 | Cargar un robot | El robot se encuentra cargando y el usuario hace clic en el botón de cargar | El robot sigue cargando |

7.5.4.2.2 Estado e incidencias

| ID | Descripción | Estado inicial | Estado final |
|----|------------------------------|---|---|
| 1 | Cambio de estado | El robot se encuentra en segando y el usuario hace clic en el botón de cargar | El estado del robot cambia a CARGANDO o APARCADO |
| 2 | Generación de una incidencia | El robot se encuentra segando y se levanta del suelo | Se genera una incidencia y llega una notificación |

7.5.5 Pruebas de Usabilidad

Para mejorar la usabilidad de la aplicación móvil, se realizarán pruebas con 3 personas las cuáles vayan a ser el perfil de usuario final que vaya a utilizar la aplicación, para conseguir unos resultados acordes a lo que se quiere mejor.

Las tareas que se realizarán serán las siguientes:

1. Creación de un cliente
2. Asignación de un robot a un cliente y empleado
3. Acción de control remoto sobre un robot.

A continuación, se presenta el formulario que se rellenará para cada persona y cada tarea que haga la prueba con el fin de recoger los resultados.

| Datos personales | |
|-------------------------|--|
| Edad | |
| Puesto laboral | |
| Experiencia informática | |
| Resultado | |
| Número de prueba | |
| Tiempo | |
| Experiencia | |
| Pasos | |
| Notas adicionales | |

Para más claridad, a continuación se explica cada fila de la tabla anterior:

- **Edad:** Edad de la persona que realiza la prueba.
- **Puesto laboral:** Puesto dentro de la empresa de la persona que realiza la prueba.
- **Experiencia informática:** Experiencia o habilidad de la persona con la informática y los dispositivos móviles.
- **Número de prueba:** Número de prueba realizada, de las indicadas arriba.
- **Tiempo:** Tiempo que le lleva a la persona completar la tarea indicada.
- **Experiencia:** Experiencia o calidad de la ejecución de la tarea por la persona que la realiza.
- **Pasos:** Pasos realizados por la persona que realiza la prueba hasta llegar a ser completada o abandonada.
- **Notas adicionales:** Cualquier comentario que quiere añadir la persona que supervisa las pruebas.

Capítulo 8 CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE DESARROLLO

CSI



8.1 CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

8.1.1 Estándares y normas seguidos

A continuación, se tratan los estándares y normas seguidos que el sistema debe cumplir para seguir con las regulaciones que existen hoy en día.

8.1.1.1 Ley de Servicios de la Sociedad de la Información

La Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSI-CE) es una normativa española que regula aspectos importantes del comercio electrónico y la prestación de servicios en línea, estableciendo un marco legal para las actividades realizadas a través de Internet y otros medios electrónicos. Aunque la aplicación no implique ventas ni comunicaciones comerciales, es crucial cumplir con ciertos aspectos de esta ley para garantizar la legalidad y seguridad en la gestión de datos de empresas, empleados y clientes.

En primer lugar, es necesario proporcionar información clara y accesible sobre la identidad del desarrollador o proveedor de la aplicación, incluyendo el nombre o denominación social, domicilio, dirección de correo electrónico y otros datos de contacto que permitan una comunicación directa y efectiva. Además, deben indicarse claramente los datos de inscripción en el Registro Mercantil u otro registro público donde la empresa esté inscrita.

La protección de datos personales es un aspecto fundamental, por lo que es obligatorio cumplir con el Reglamento General de Protección de Datos (RGPD) y la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD). Es esencial informar a los usuarios sobre la recogida y tratamiento de sus datos personales, obteniendo su consentimiento explícito cuando sea necesario, e implementar medidas de seguridad adecuadas para proteger estos datos contra accesos no autorizados, alteraciones, pérdidas o destrucción.

Si la aplicación utiliza cookies u otras tecnologías de seguimiento, es obligatorio informar a los usuarios y obtener su consentimiento para su instalación, proporcionando información clara y completa sobre el uso de estas tecnologías y su finalidad. La aplicación móvil no utiliza cookies, por lo que no aplica en este caso y no es necesario tomar ninguna medida.

8.1.1.2 Reglamento General de Protección de Datos

El sistema almacena diferentes datos personales de distintas entidades y personas. En concreto, se almacenan datos de empresas, como el CIF y el nombre, datos de empleados, como el nombre, y datos de clientes, incluyendo el nombre, número de teléfono y dirección. Dado que estos son datos personales, es imperativo cumplir con los requisitos establecidos por el Reglamento General de Protección de Datos (RGPD).

Para cumplir con la RGPD, es esencial incluir un aviso informativo (derecho de información) en el momento del registro del usuario. Este aviso debe detallar los datos del responsable del servicio y los derechos del usuario, tal como indica el artículo 11 de la RGPD. Además, es crucial que la pantalla de registro incluya un acceso claro a este aviso, indicando que al registrarse, el usuario acepta las condiciones establecidas.

Más allá del derecho de información, la RGPD también requiere la implementación de medidas de seguridad adecuadas y proporcionadas al riesgo asociado al tipo y cantidad de datos personales recogidos. Aunque en este caso los datos recolectados son de bajo riesgo y en cantidades limitadas, se deben adoptar ciertas medidas básicas para asegurar su protección. Estas medidas incluyen:

- **Accesos individuales protegidos por contraseña:** Cada usuario que accede a los datos personales debe hacerlo mediante credenciales únicas, asegurando que solo personas autorizadas puedan acceder a la información.
- **Uso de protocolo seguro HTTPS:** Implementar HTTPS en el despliegue de la aplicación garantiza que la transmisión de datos entre el servidor y el usuario esté cifrada, protegiendo la información contra intercepciones durante el tránsito.
- **Copias de seguridad periódicas:** Realizar copias de seguridad de los datos de manera regular, conforme a un plan de backup predefinido, asegura que en caso de pérdida de datos o fallo del sistema, la información pueda ser recuperada sin mayores inconvenientes.

Además de estas medidas, se recomienda mantener un registro detallado de las actividades de tratamiento de datos y asegurarse de que todo el personal involucrado en el manejo de datos personales esté adecuadamente formado en las políticas de protección de datos y en la importancia de mantener la confidencialidad y seguridad de la información. Implementar auditorías periódicas para revisar y actualizar las políticas de protección de datos también puede ayudar a asegurar el cumplimiento continuo con la RGPD.



8.1.1.3 Licencias de bibliotecas y servicios de terceros.

En el sistema, se emplean diversas bibliotecas de terceros que se integran en el código para simplificar el desarrollo de ciertas funcionalidades. Estas funcionalidades incluyen, entre otras, cajas de búsqueda avanzadas, selectores de elementos, y validación de datos. Todas las bibliotecas utilizadas son de código abierto (Open Source) y se adecuan a los objetivos y el uso previsto de este sistema gratuito. Es obligatorio incluir el texto de cada una de estas licencias en la aplicación, lo cual puede realizarse, por ejemplo, a través de una sección específica en la interfaz del usuario.

Además, se utilizan servicios externos proporcionados por terceros para extender las capacidades del sistema. Ejemplos de estos servicios incluyen la API de Husqvarna, que se emplea para obtener información sobre los robots, y Firebase, que se utiliza para enviar notificaciones a dispositivos móviles. Cada uno de estos servicios externos tiene su propia política de uso, y se ha verificado que dichas políticas son compatibles con los objetivos y el uso previsto de este sistema. Por lo tanto, se asegura que el uso de estos servicios no infringe ninguna normativa y se ajusta perfectamente a las necesidades del sistema.

8.1.2 Lenguajes de programación y herramientas

En este apartado, se especifican los lenguajes y herramientas utilizados en el desarrollo del sistema:

8.1.2.1 Backend – REST API y Base de Datos



Para el desarrollo de la REST API, que forma el backend del sistema, se utilizó el entorno de ejecución Node.js junto a Express.js, utilizando como lenguaje de programación Typescript. La persistencia de los datos como ya se explicó anteriormente se hace utilizando MySQL.

Las herramientas utilizadas, se usó Visual Studio Code, ya que es gratuito y ofrece una amplia gama de funcionalidades, DBEaver como visualizador para la base de datos y Postman para mandar peticiones a la API.

La API de Husqvarna es el servicio externo principal, de donde se obtiene toda la información de los robots. Además se usa Firebase para el envío de notificaciones.

8.1.2.2 Frontend – Aplicación móvil

Lenguajes



Dart

Tecnologías



Flutter

Herramientas



Visual Studio Code



Apple Simulator



Android Studio

Servicios externos



Firebase

Para el desarrollo de la aplicación móvil, se utilizó la tecnología Flutter, para un desarrollo multiplataforma, junto al lenguaje de programación Dart.

Al igual que para el backend, se utilizó Visual Studio Code como IDE para manipular el código. Además, se utilizó Apple Simulator y Android Studio para poder ejecutar la aplicación móvil de manera local en el entorno de desarrollo.

Como en la API, se hace uso de Firebase como servicio externo, para el envío y recepción de notificaciones.

8.1.2.3 Pruebas del sistema

Lenguajes



Dart



Typescript

Tecnologías

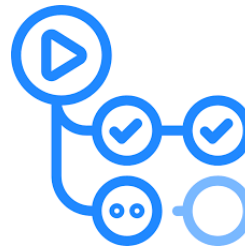


Jest



Test Containers

Herramientas



GitHub Actions

Las pruebas del sistema se realizaron en el backend usando Jest y Test Containers como tecnologías, para poder simular un entorno real y así hacer el mayor número de tipos de prueba. Como lenguaje de programación para estas pruebas se utilizó Typescript.

En cuanto al frontend, se hicieron utilizando el lenguaje de programación Dart.

Ambas pruebas se ejecutan cuando se suben al repositorio de GitHub, haciendo uso de la herramienta GitHub Actions. La cual se utilizó para automatizar flujos de trabajo de pruebas y despliegue, permitiendo ejecutar pruebas automáticamente con cada commit, lo que facilita la integración y el despliegue continuos.

8.1.2.4 Documentación y control de versiones

Documentación



Microsoft Word



Microsoft Project



Microsoft Excel

Presentación



Microsoft PowerPoint

Diagramas y diseños



Draw.io



Balsamiq

Control de versiones



OneDrive



Git



GitHub



GitKraken

Para la redacción de la memoria y documentación, se utilizó Microsoft Word. Una parte importante de la memoria, la planificación, se creó usando Microsoft Project. Además, los presupuestos y riesgos se realizaron en Microsoft Excel.

La presentación de la defensa se realizó usando la herramienta Microsoft PowerPoint.

Diagramas y diseños que forman parte de esta memoria, se realizaron usando draw.io, así como Balsamiq para los prototipos de pantalla.

Respecto al control de versiones, para la memoria se utilizó OneDrive. Mientras que para el código se utilizó GitHub, basado en la tecnología Git, a través de la herramienta GitKraken.



8.2 CSI 3: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

Todas las pruebas especificadas en los apartados [7.5.1 Pruebas Unitarias](#) y [7.5.2 Pruebas de Seguridad](#) han sido superadas satisfactoriamente. Estas pruebas han obtenido una cobertura de un 86% del total del código del backend, ejecutándose 176 tests en total.

```
Test Suites: 29 passed, 29 total
Tests:      176 passed, 176 total
Snapshots:  0 total
Time:       14.916 s, estimated 17 s
```

Respecto al frontend, como ya se comentó en apartados anteriores, las pruebas que se hicieron fueron menores que las pruebas relacionadas con el backend. El 52% del código del frontend fue cubierto gracias a las pruebas unitarias que se realizaron.

8.3 CSI 4: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

Las pruebas especificadas en el apartado [7.5.3 Pruebas de Integración](#) se han superado correctamente. A continuación, se presenta una tabla con la salida obtenida en cada una de las pruebas.

| ID | Descripción | Entrada | Salida esperada | Salida obtenida |
|----|--|-----------|-----------------------------|-----------------------------|
| 1 | Creación de un cliente | Válida | Cliente y 201 | Cliente y 201 |
| 2 | Creación de un cliente repetido | Válida | Mensaje de error y 409 | Mensaje de error y 409 |
| 3 | Creación de un cliente | No válida | Mensaje de error y 400 | Mensaje de error y 400 |
| 4 | Borrado de un cliente | Válida | Cuerpo vacío y 204 | Cuerpo vacío y 204 |
| 5 | Borrado de un cliente no existente | Válida | Mensaje de error y 404 | Mensaje de error y 404 |
| 6 | Borrado de un cliente | No válida | Mensaje de error y 400 | Mensaje de error y 400 |
| 7 | Obtener clientes para una empresa | Válida | Lista de clientes y 200 | Lista de clientes y 200 |
| 8 | Obtener clientes para una empresa | No válida | Mensaje de error y 400 | Mensaje de error y 400 |
| 9 | Asignación de un robot | Válida | Robot asignado y 200 | Robot asignado y 200 |
| 10 | Asignación de un robot con cliente no válido | No válida | Mensaje de error y 400 | Mensaje de error y 400 |
| 11 | Asignación de un robot con empleado no válido | No válida | Mensaje de error y 400 | Mensaje de error y 400 |
| 12 | Marcar como leída una incidencia | Válida | Marcada como leída y 200 | Marcada como leída y 200 |
| 13 | Marcar como no leída una incidencia | Válida | Marcada como no leída y 200 | Marcada como no leída y 200 |
| 14 | Marcar como leída una incidencia que no existe | Válida | Mensaje de error y 404 | Mensaje de error y 404 |

8.4 CSI 5: EJECUCIÓN DE LAS PRUEBAS DEL SISTEMA

8.4.1 Pruebas E2E y en entornos reales

Las pruebas especificadas en el apartado [7.5.4 Pruebas E2E y en entornos reales](#), se han ejecutado correctamente excepto dos que han dado un resultado incorrecto. A continuación, se presenta una tabla con el estado final obtenido.

| ID | Descripción | Estado inicial | Estado final esperado | Estado final obtenido |
|----|------------------------|---|---------------------------------------|---------------------------------------|
| 1 | Poner a segar un robot | El robot se encuentra en la base de carga y el usuario hace clic en el botón de segar | El robot empieza a segar | El robot se queda en la base de carga |
| 2 | Poner a segar un robot | El robot se encuentra parado y el usuario hace clic en el botón de segar | El robot empieza a segar | El robot continúa parado |
| 3 | Poner a segar un robot | El robot se encuentra segando y el usuario hace clic en el botón de segar | El robot sigue segando | El robot sigue segando |
| 4 | Parar un robot | El robot se encuentra en la base de carga y el usuario hace clic en el botón de parar | El robot sigue en la base de carga | El robot sigue en la base de carga |
| 5 | Parar un robot | El robot se encuentra segando y el usuario hace clic en el botón de parar | El robot se para | El robot se para |
| 6 | Parar un robot | El robot se encuentra parado y el usuario hace clic en el botón de parar | El robot sigue parado | El robot sigue parado |
| 7 | Cargar un robot | El robot se encuentra parada y el usuario hace clic en el botón de cargar | El robot se dirige a la base de carga | El robot se dirige a la base de carga |



| | | | | |
|---|-----------------|---|---------------------------------------|---------------------------------------|
| 8 | Cargar un robot | El robot se encuentra segando y el usuario hace clic en el botón de cargar | El robot se dirige a la base de carga | El robot se dirige a la base de carga |
| 9 | Cargar un robot | El robot se encuentra cargando y el usuario hace clic en el botón de cargar | El robot sigue cargando | El robot sigue cargando |

8.4.1.1 Cambios realizados

Debido a los dos fallos encontrados en estas pruebas, se realiza un cambio en el código de la REST API, para que utilice otra función de la API externa de robots. Con este cambio dichas dos pruebas así como las demás, funcionan correctamente.

8.4.2 Prueba de Usabilidad

Las pruebas de usabilidad que se van a realizar así como el formulario que se utilizará para registrar los resultados se definen en el apartado [7.5.5 Pruebas de Usabilidad](#).

Se realizaron las tres pruebas definidas con dos personas diferentes. Una vez terminadas, las pruebas se recogen los resultados y se definen las mejoras a realizar sobre la aplicación.

8.4.2.1 Pruebas realizadas

| Datos personales | |
|-------------------------|---|
| Edad | 52 |
| Puesto laboral | Jefe |
| Experiencia informática | Baja |
| Resultado | |
| Número de prueba | 1 |
| Tiempo | 30 segundos |
| Experiencia | Media |
| Pasos | Le cuesta llegar a la vista de clientes, una vez ahí tarda poco en darle al botón de creación de un cliente, introducir los datos y presionar el botón de creación de un cliente. |

| Datos personales | |
|-------------------------|--|
| Edad | 53 |
| Puesto laboral | Contable |
| Experiencia informática | Media |
| Resultado | |
| Número de prueba | 1 |
| Tiempo | 45 segundos |
| Experiencia | Normal |
| Pasos | Llega bien a la vista de clientes, encuentra el botón de añadir un nuevo cliente sin mucha dificultad, introduce de manera errónea los datos, aparece el error, cambia los datos erróneos y crea el cliente de manera satisfactoria. |



| Datos personales | |
|--------------------------------|---|
| Edad | 52 |
| Puesto laboral | Jefe |
| Experiencia informática | Baja |
| Resultado | |
| Número de prueba | 2 |
| Tiempo | 1 minuto |
| Experiencia | Mala / Normal |
| Pasos | Al igual que la primera prueba, no se da cuenta del botón de añadir un nuevo robot, se le tiene que explicar. Una vez ahí, selecciona fácilmente el robot, el cliente y empleado. Asigna el robot sin mayor problema. |

| Datos personales | |
|--------------------------------|--|
| Edad | 53 |
| Puesto laboral | Contable |
| Experiencia informática | Media |
| Resultado | |
| Número de prueba | 2 |
| Tiempo | 30 segundos |
| Experiencia | Buena |
| Pasos | Llega bien a la vista para asignar un robot. Selecciona de manera correcta el cliente y empleado utilizando el buscador, y asigna el robot de manera correcta. |



| Datos personales | |
|--------------------------------|---|
| Edad | 52 |
| Puesto laboral | Jefe |
| Experiencia informática | Baja |
| Resultado | |
| Número de prueba | 3 |
| Tiempo | 2 minutos |
| Experiencia | Mala |
| Pasos | Llega bien a la vista de información detallada y realiza una acción de control remoto, pero al volver a la lista de robots e intentar visualizar el nuevo estado, no es capaz de actualizar la información de los robots deslizando para abajo. Se le tiene que explicar. |

| Datos personales | |
|--------------------------------|---|
| Edad | 53 |
| Puesto laboral | Contable |
| Experiencia informática | Media |
| Resultado | |
| Número de prueba | 3 |
| Tiempo | 1 minuto |
| Experiencia | Normal |
| Pasos | Al igual que la persona anterior, realiza la acción sin mayor problema, pero tiene problemas para actualizar la información aunque se da cuenta por sí misma. |

8.4.2.2 Cambios realizados

El principal problema es la actualización de la información de los robots. El deslizar desde la lista no parece muy intuitivo, por lo que se decide prescindir de este sistema e implementar un botón de actualizar en la barra superior junto al botón de asignar un nuevo robot.

Otra mejora que se estudió, pero no se llegó a implementar, es mostrar en la barra inferior un texto que acompañe al icono, para así facilitar la navegación. Debido a la falta de espacio se decide no hacerlo ya que es inviable incluso reduciendo el tamaño.



8.5 CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO

8.5.1 Manual de Instalación o Despliegue

El manual a continuación detalla los pasos necesarios para el despliegue de la REST API y aplicación móvil desarrolladas para este sistema.

8.5.1.1 REST API

En el repositorio de GitHub donde se encuentra almacenado todo el código para el backend del sistema, existe una GitHub Action encargada de realizar todos los pasos del despliegue. Esta acción se ejecuta cada vez que se crea una release. Crea una imagen de Docker basándose en el tag de dicha release y la sube al repositorio de imágenes de GitHub (gchr.io). Por último, se conecta al servidor donde se va a desplegar y lanza una serie de comandos de Docker, para parar el contenedor donde se está ejecutando la API y desplegar un nuevo con la imagen actualizada. Todo este proceso facilita el despliegue, ya que ahorra mucho tiempo y sigue la tendencia de la industria de realizar entornos CI/CD.

A continuación, se explican los pasos manuales que se deben hacer para desplegar la API en caso de así requerirlo.

Primero, debemos contar con un servidor remoto corriendo en la nube, que nos proporciona mucha disponibilidad y un coste reducido. Antes de desplegar la REST API, debemos crear la base de datos MySQL. Para ello necesitamos instalar una instancia de dicho software, crear una base de datos llamada “mowercontrol”, crear las tablas necesarias e introducir los datos iniciales. En la carpeta `src/db` se proporciona un script que facilita la inicialización de las tablas y datos.

Cuando la base de datos este corriendo correctamente, ya podemos desplegar la REST API. Para ello primero necesitamos una serie de credenciales o secretos para el correcto funcionamiento del sistema y conectarse a servicios externos:

- URL y credenciales de la base de datos MySQL
- Credenciales de la cuenta de desarrollador de Husqvarna
- Cuenta del servicio de Firebase

El primer paso es crear la imagen de Docker, para ello en el repositorio raíz del proyecto se provee un archivo `Dockerfile` que se encarga de este proceso. Se sustituyen las variables de entorno por los secretos mencionados anteriormente y con el comando `docker build -t mower-control-api` se crea la imagen de Docker.



Como último paso hay que desplegar esta imagen en el servidor remoto. Primero, hay que tener la imagen en el servidor remoto, para esto hay varias alternativas. Subir la imagen a un repositorio y descargarla en el servidor o copiarla directamente a través de un protocolo como scp.

Cuando ya tengamos la imagen en el servidor, con el siguiente comando creamos un contenedor donde se ejecuta la REST API.

```
docker run -d --name mower_control_api -p 8080:8080  
ghcr.io/alvarorg14/mower_contro_api:latest
```

Se debe sustituir la última parte del comando por el directorio donde se encuentre la imagen creada.

Una vez hechos todos estos pasos, la REST API ya se encontraría funcionando en el servidor remoto, accesible para cualquier usuario.

8.5.1.2 Aplicación móvil

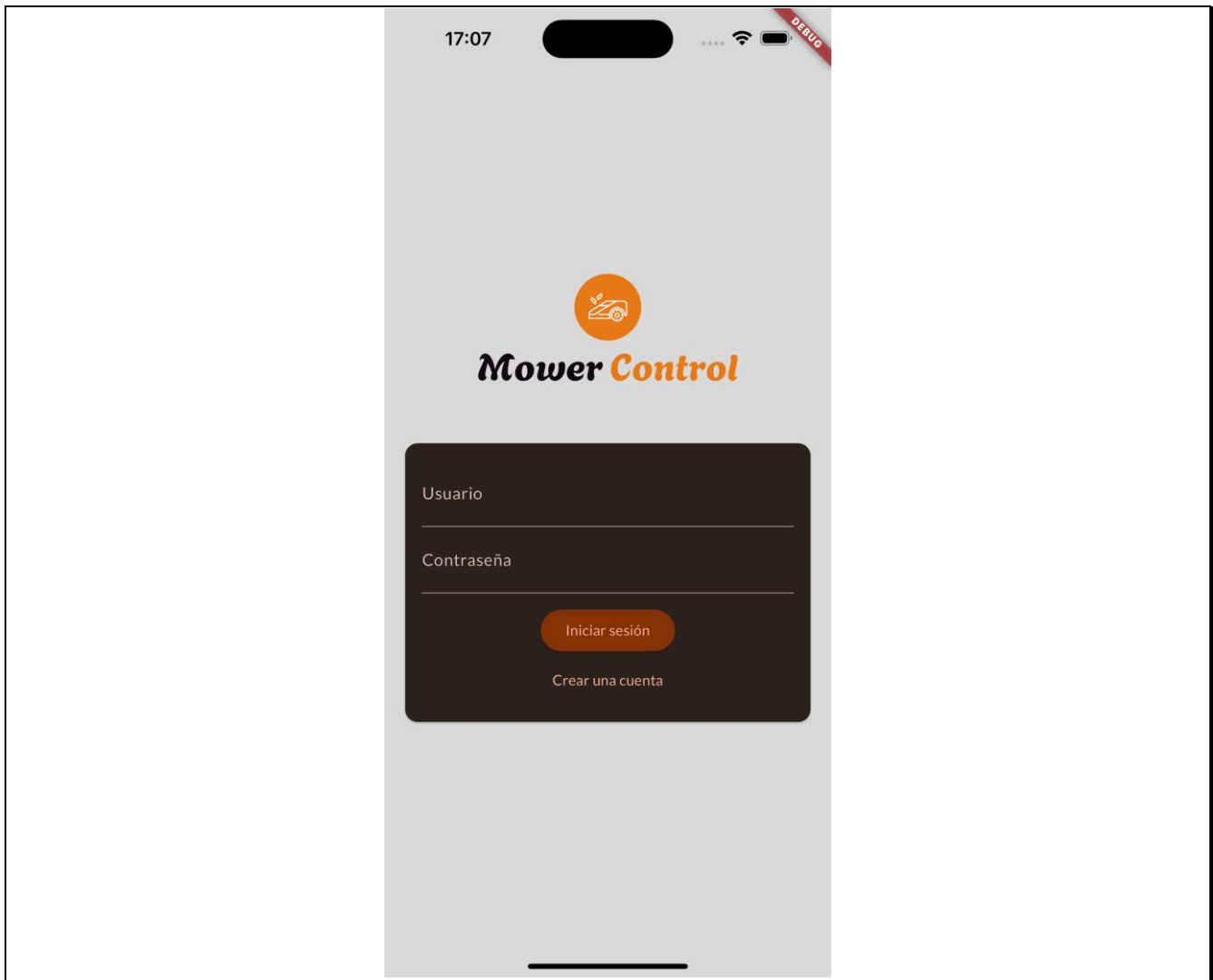
La aplicación móvil no requiere un despliegue, ya que se ejecuta de manera local en cada dispositivo. Para ello se debe generar un archivo APK para la aplicación, el cual podrá ser instalado en un dispositivo móvil. Este archivo se genera utilizando los comandos de Flutter para generar una APK dependiendo de en qué plataforma se vaya a ejecutar la aplicación.

8.5.2 Manual de Usuario

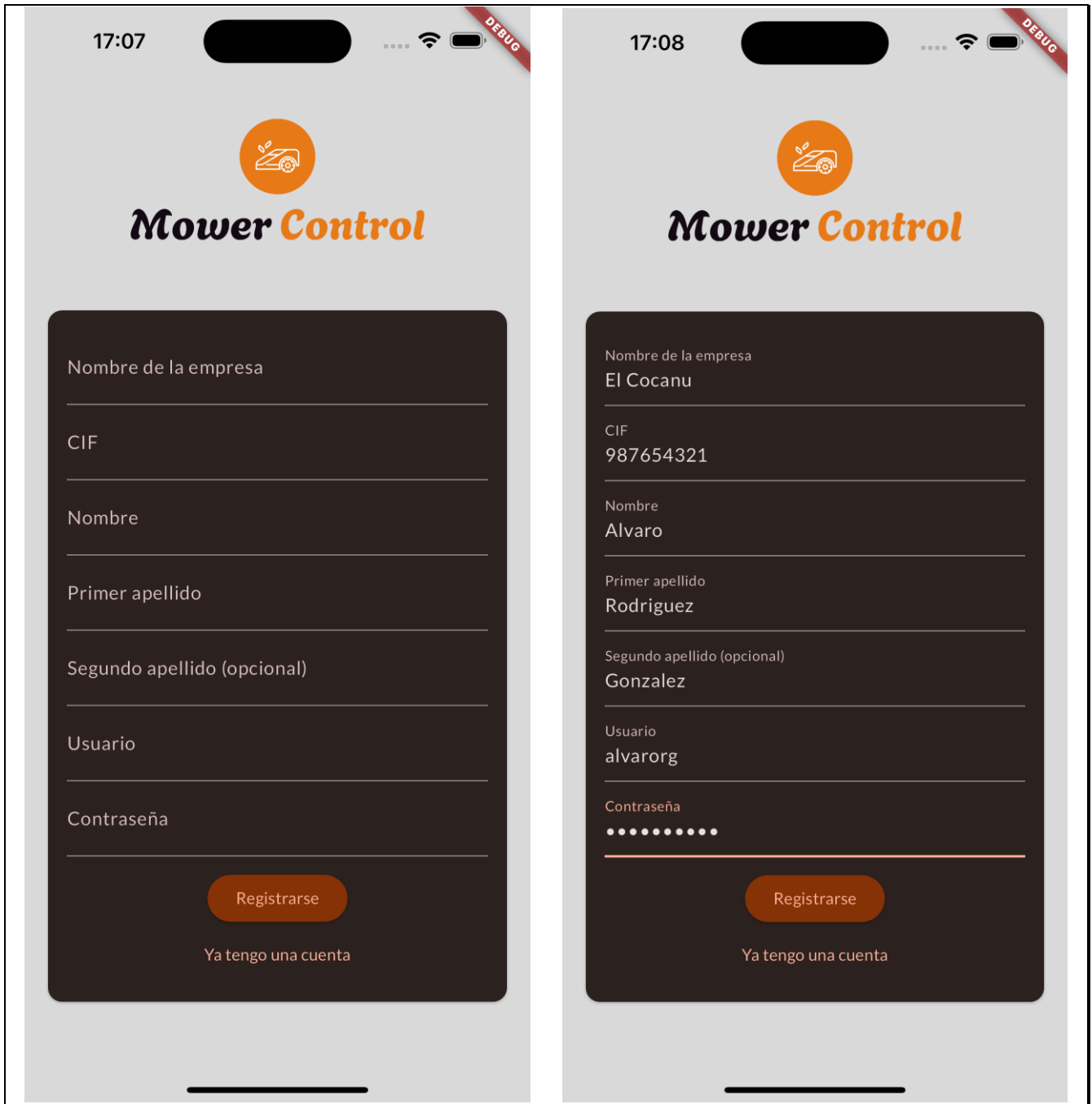
En esta sección, se detalla el manual de usuario, para que los usuarios finales que vaya a utilizar la aplicación sepan el correcto funcionamiento del sistema y todas las funcionalidades que proporciona.

8.5.2.1 Inicio de sesión y Registro

Al iniciar la aplicación móvil por primera vez, se presenta la pantalla de inicio de sesión. En caso de que el usuario ya se haya registrado previamente, debe introducir sus credenciales, usuario y contraseña, en este caso y darle al botón de iniciar sesión. Con esto el usuario se autenticará y la aplicación le mostrará la lista de robots. En caso de no estar registrado previamente debe hacer clic en el botón de “Crear una cuenta”, esto abrirá el formulario para registrar una nueva empresa y empleado administrador.

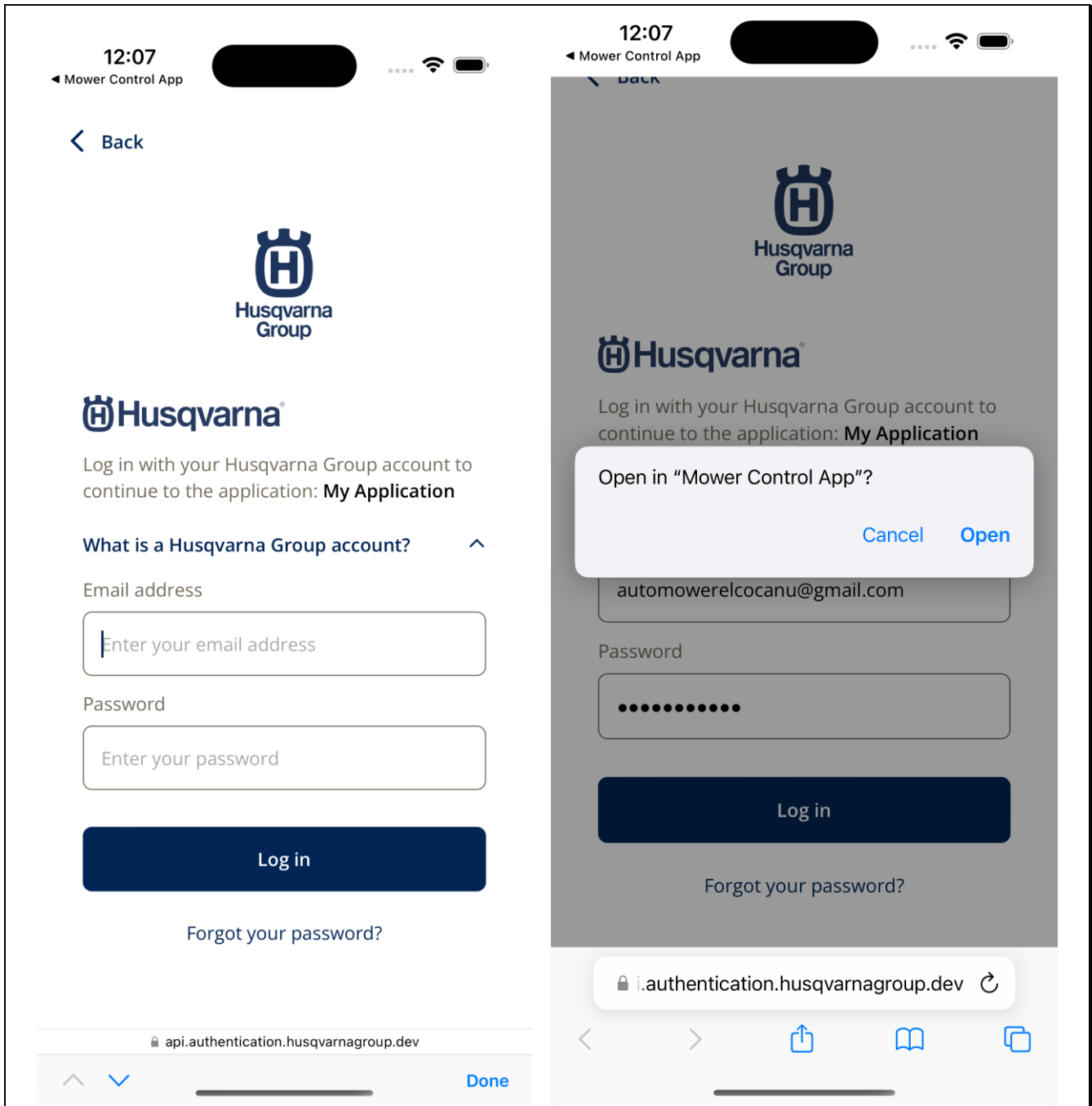


En la vista de registro, el usuario debe introducir la información relacionada con la empresa que quiere registrar así como los datos para el empleado administrador de dicha empresa. Una vez introducidos los datos, el usuario debe hacer clic en el botón de Registro, se le redirige a una vista para introducir las credenciales de su cuenta de Husqvarna. En caso de que quiera iniciar sesión, debe hacer clic en el botón de “Ya tengo una cuenta”.



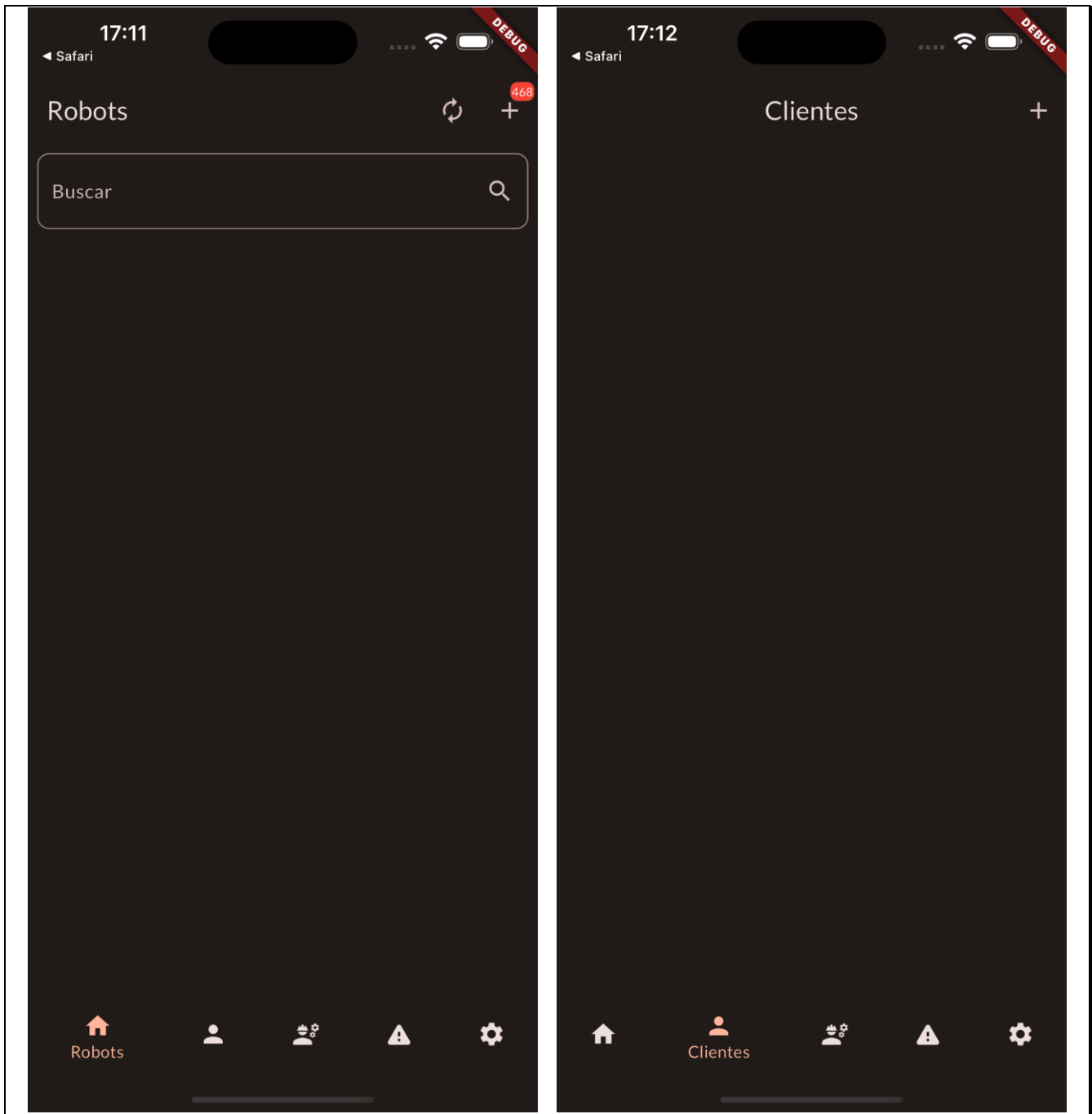
The image displays two screenshots of the Mower Control mobile application registration screen. Both screens feature a grey background with a white status bar at the top showing the time, signal strength, and battery level. A red 'DEBUG' banner is visible in the top right corner of each screen. The app's logo, an orange circle with a white mower icon, is centered above the title 'Mower Control' in a bold, orange font. Below the title is a dark grey registration form with white text labels and input fields. The form includes fields for 'Nombre de la empresa', 'CIF', 'Nombre', 'Primer apellido', 'Segundo apellido (opcional)', 'Usuario', and 'Contraseña'. At the bottom of the form are two buttons: an orange 'Registrarse' button and a white 'Ya tengo una cuenta' button with an orange border. The left screenshot (17:07) shows the form with all fields empty. The right screenshot (17:08) shows the form filled with the following data: 'Nombre de la empresa: El Cocanu', 'CIF: 987654321', 'Nombre: Alvaro', 'Primer apellido: Rodriguez', 'Segundo apellido (opcional): Gonzalez', 'Usuario: alvarorg', and 'Contraseña: [masked with dots]'. The 'Registrarse' button is highlighted in orange, indicating it has been pressed.

El usuario es enviado a la web app de Husqvarna donde debe introducir sus credenciales de la cuenta de Husqvarna. Estas credenciales se piden para que la aplicación MowerControl sea capaz de obtener la información de los robots de manera autónoma. Una vez introducido el correo electrónico y la contraseña, en caso de que sean correctos, se vuelve a redirigir a la aplicación móvil MowerControl. Ahí empezará una carga, de aproximadamente 10 segundos, donde la aplicación está inicializando toda la información para la nueva empresa, cuando acabe se muestra la lista de robots vacía.

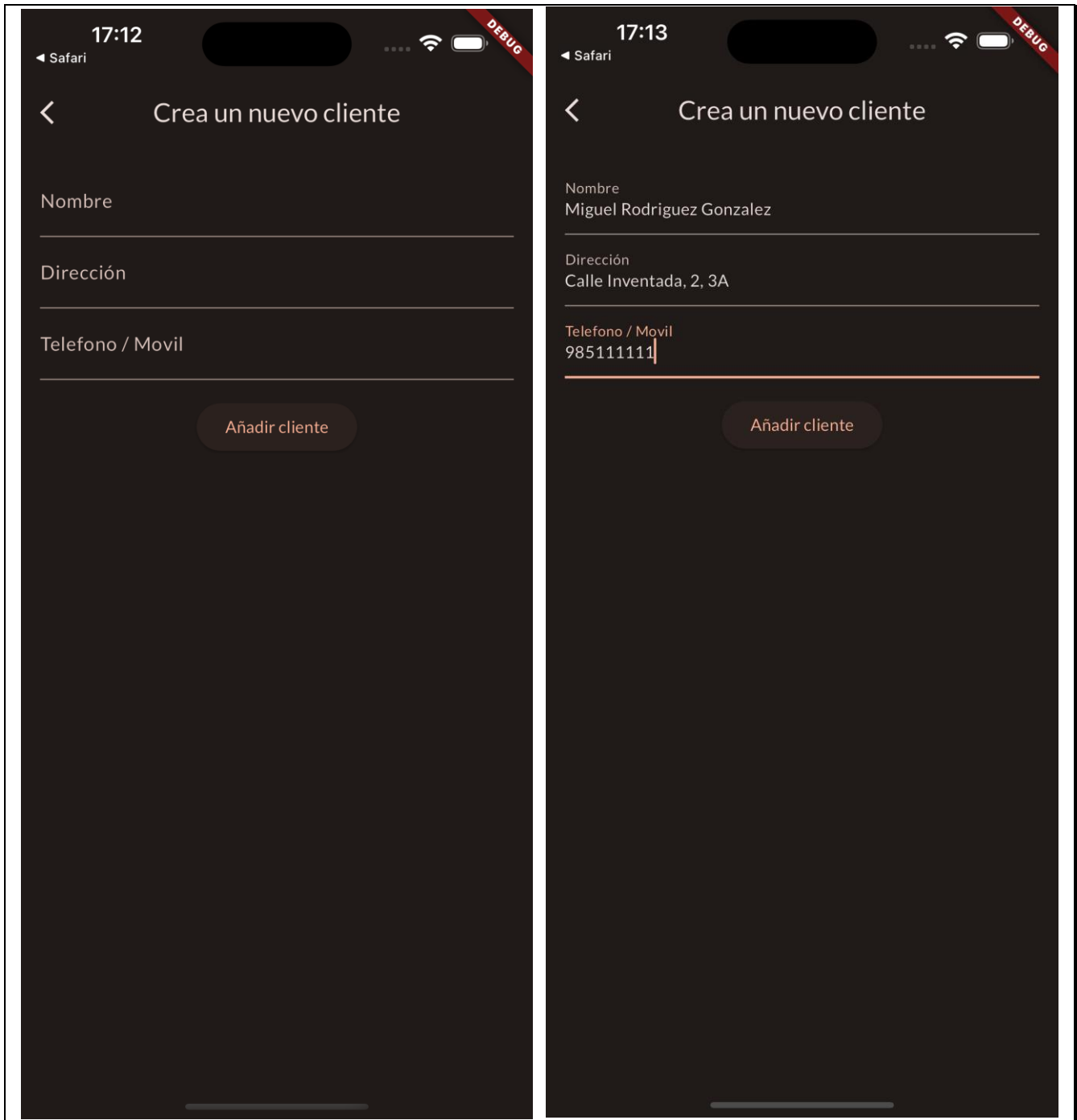


8.5.2.2 Asignación de un robot a cliente y empleado

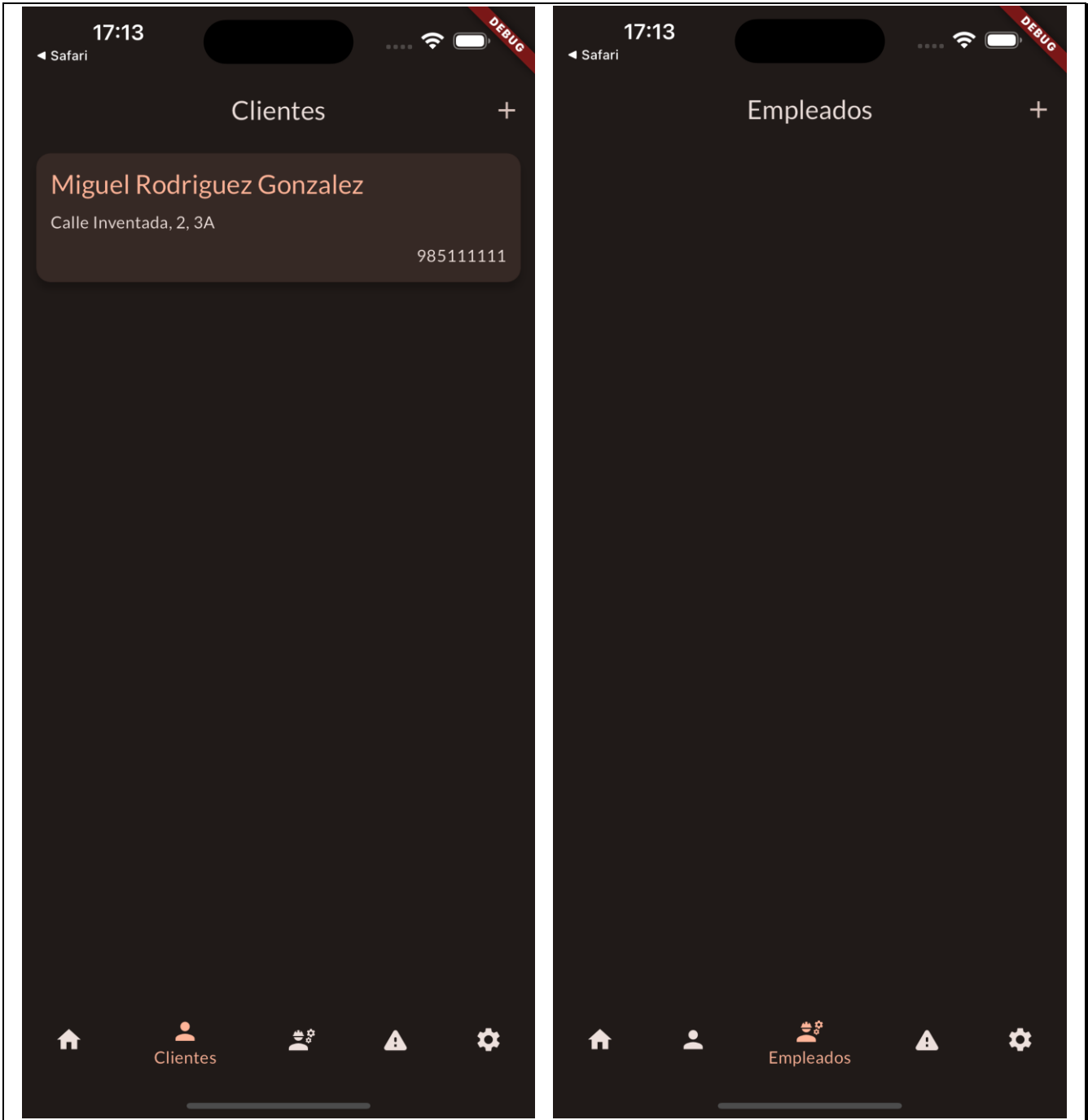
Una vez registrada la empresa, se nos muestra la lista de robots principal, al principio siempre estará vacía ya que no tenemos ningún robot asignado y en dicha solo se muestran los robots asignados. Para asignar un nuevo robot, se hace clic en el botón “+” de arriba a la derecha, pero antes necesitamos añadir un cliente y un empleado, usando la barra de navegación de la parte inferior podemos ir a las diferentes vistas de la aplicación. Con la segunda opción de dicha barra nos vamos a la vista de clientes.



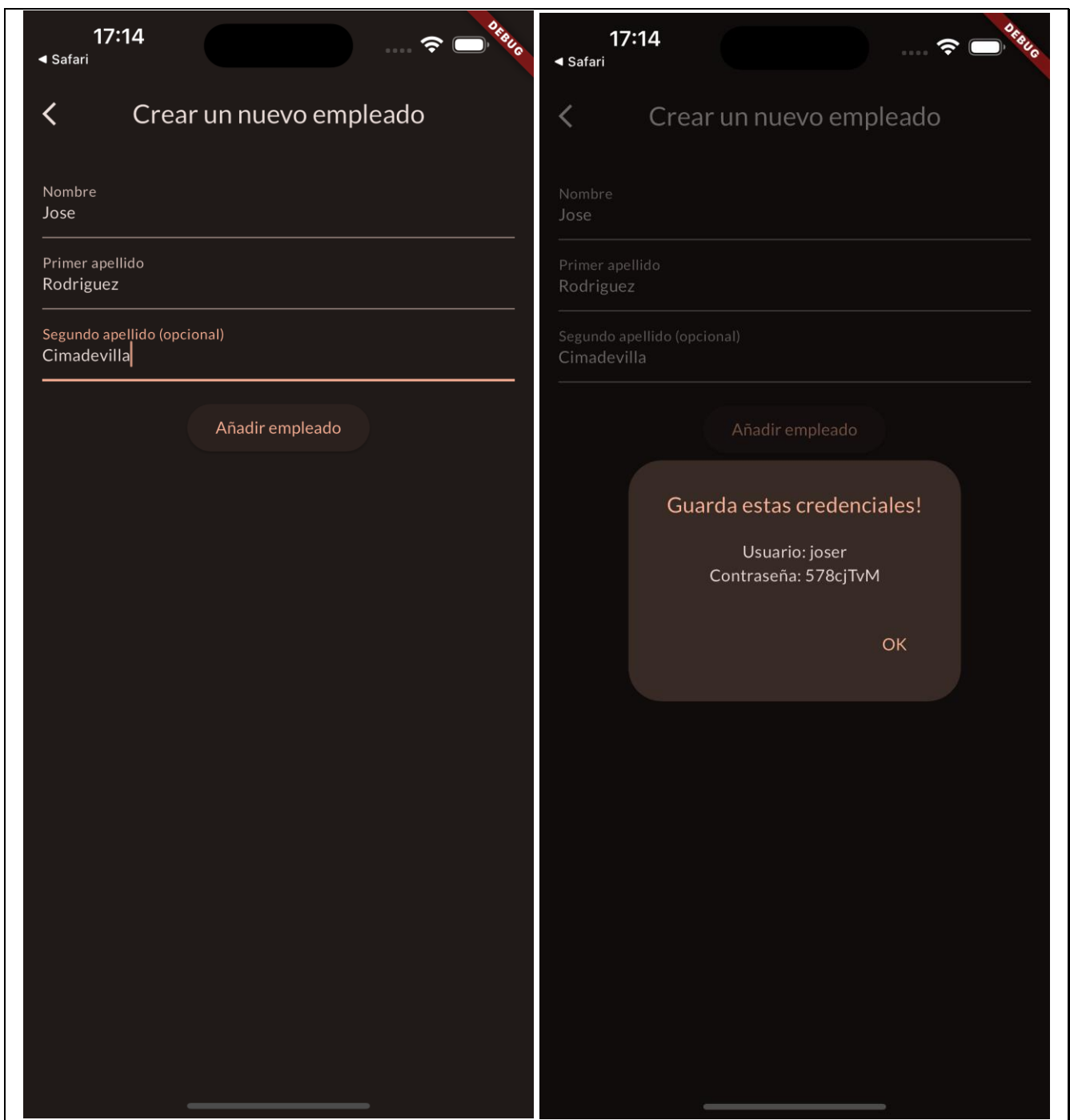
Para añadir un nuevo cliente, hacemos clic en el botón “+” de arriba a la derecha e introducimos los datos necesarios. Los datos obligatorios son nombre, dirección y número de teléfono. Una vez introducidos los datos, hacemos clic en el botón “Añadir cliente”. En caso de que los datos sean válidos, se crea el cliente y la aplicación nos lleva de nuevo a la lista de clientes.



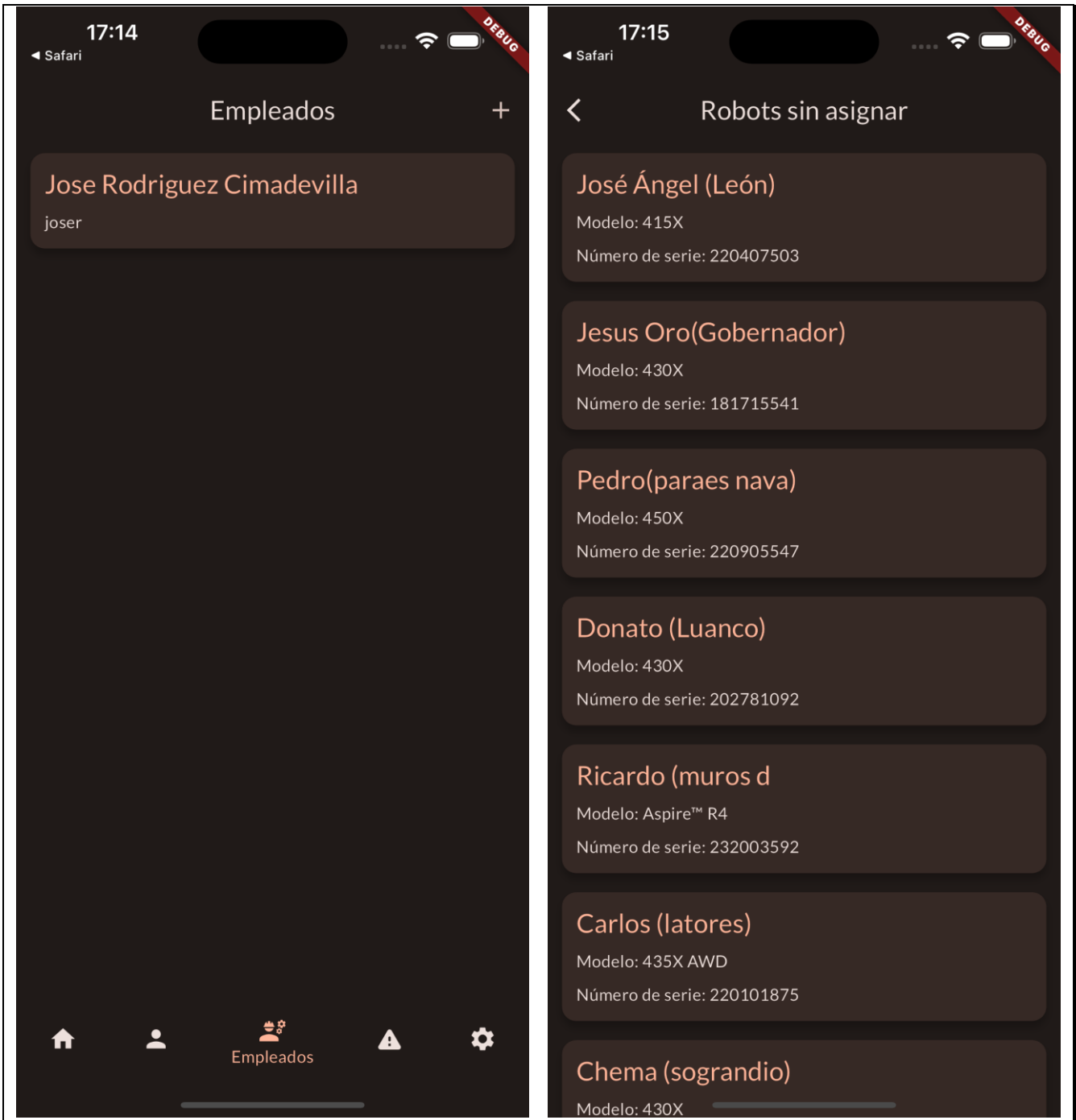
En la lista de clientes, ya nos aparece el cliente que acabamos de crear. Ahora, vamos a crear un nuevo empleado. El procedimiento es bastante similar. Primero, haciendo clic en la tercera opción de la barra de navegación, nos vamos a la vista de empleados, que actualmente se encuentra vacía, ya que el propio empleado administrador no aparece en dicha lista.



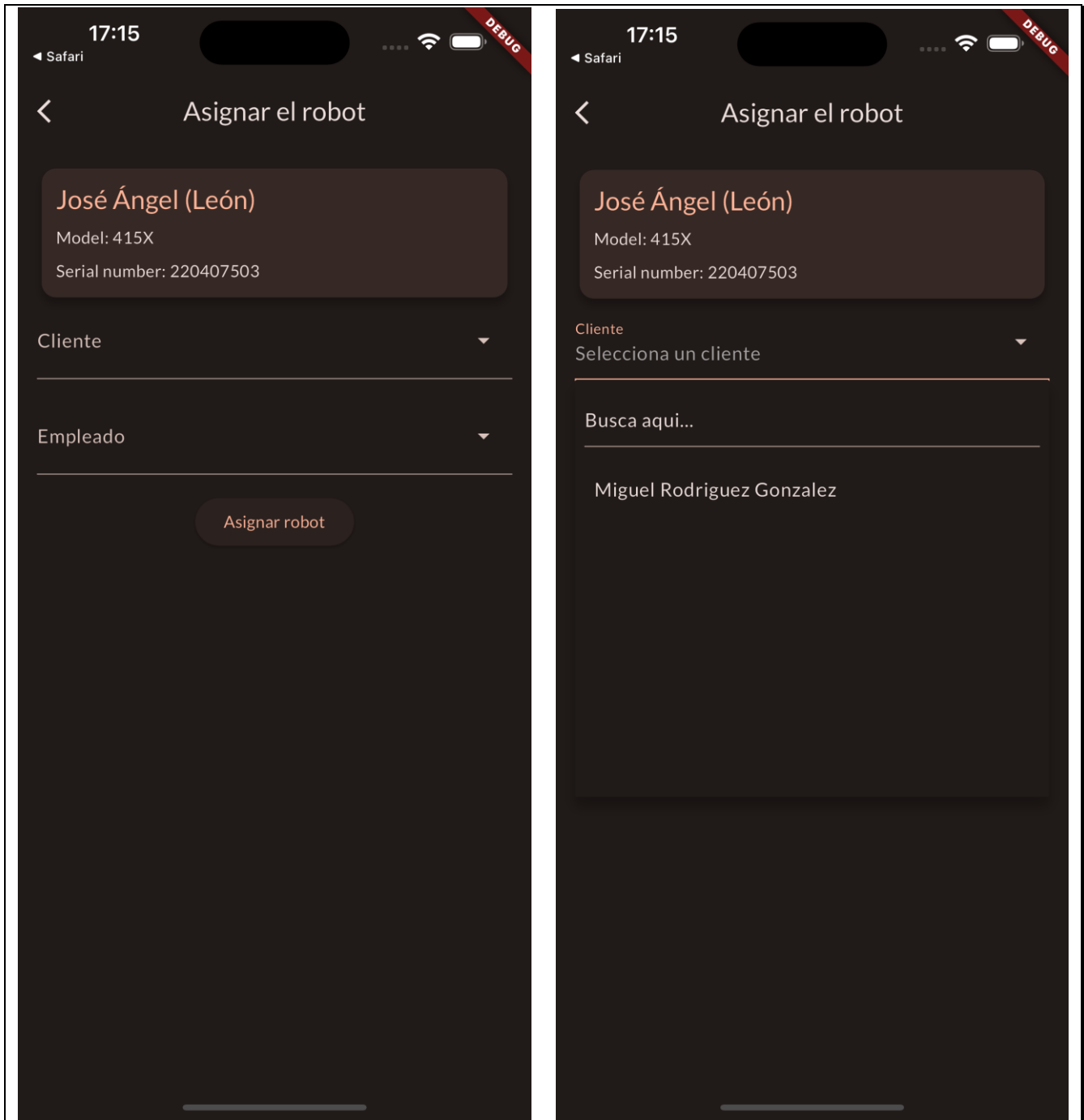
Al igual que para los clientes, hacemos clic en el botón “+” de arriba a la derecha, el cual nos lleva al formulario para añadir un nuevo empleado. En este caso los datos que tenemos que introducir, son nombre, primer apellido y segundo apellido. Este último es opcional, por lo tanto no hace falta que lo añadamos si no es necesario. Una vez introducidos los datos, hacemos clic en el botón “Añadir empleado”. El sistema valida los datos y genera las credenciales para el nuevo empleado. Estas credenciales son únicas y deben ser guardadas por el usuario administrador para comunicárselas al empleado. Cuando se confirma que se han guardado las credenciales, la aplicación muestra la lista de empleados.



En la lista de empleados, ya aparece el nuevo empleado creado. Ahora ya podemos asignar un robot, ya que tenemos un cliente y empleado creados. Para esto volvemos a la vista principal, haciendo clic en la primera opción de la barra de navegación. En dicha vista, usando el botón “+” de arriba a la derecha, el sistema nos muestra la lista de robots sin asignar. Buscamos el robot que queremos asignar y hacemos clic en él.

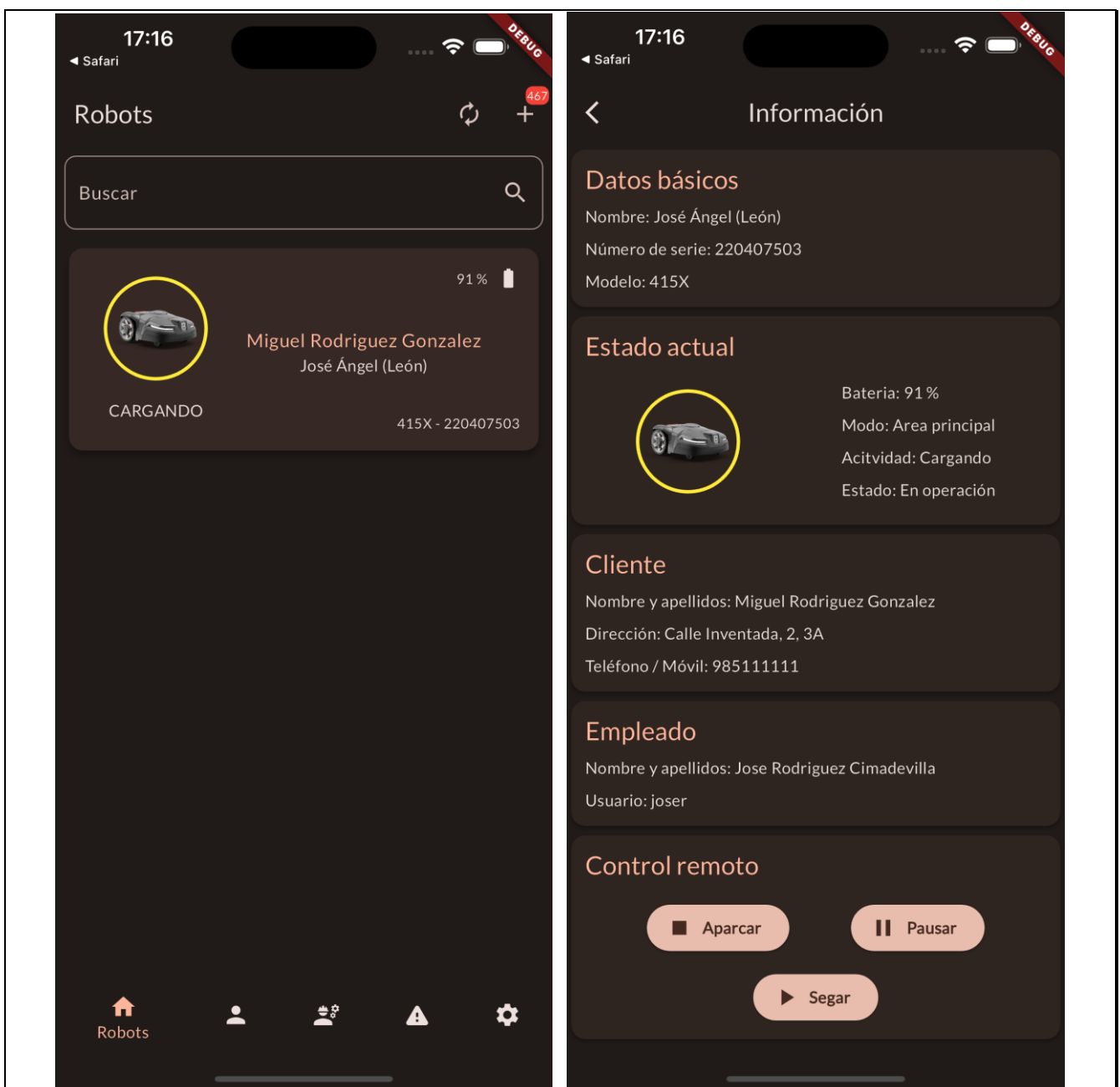


Una vez que elegimos un robot para asignar, la aplicación nos muestra la vista para asignar un robot, donde podemos seleccionar el cliente y empleado que queremos asignar. Usando los selectores que aparecen en la vista, los cuales disponen de buscador para en caso de tener muchos clientes y/o empleados se haga más fácil encontrar el que queremos. Una vez seleccionados el cliente y empleado, hacemos clic en el botón “Asignar robot”. El sistema asigna el robot al cliente y empleado, y muestra la lista principal de robots. Ahí ya podemos ver el robot que acabamos de asignar e información básica del mismo.



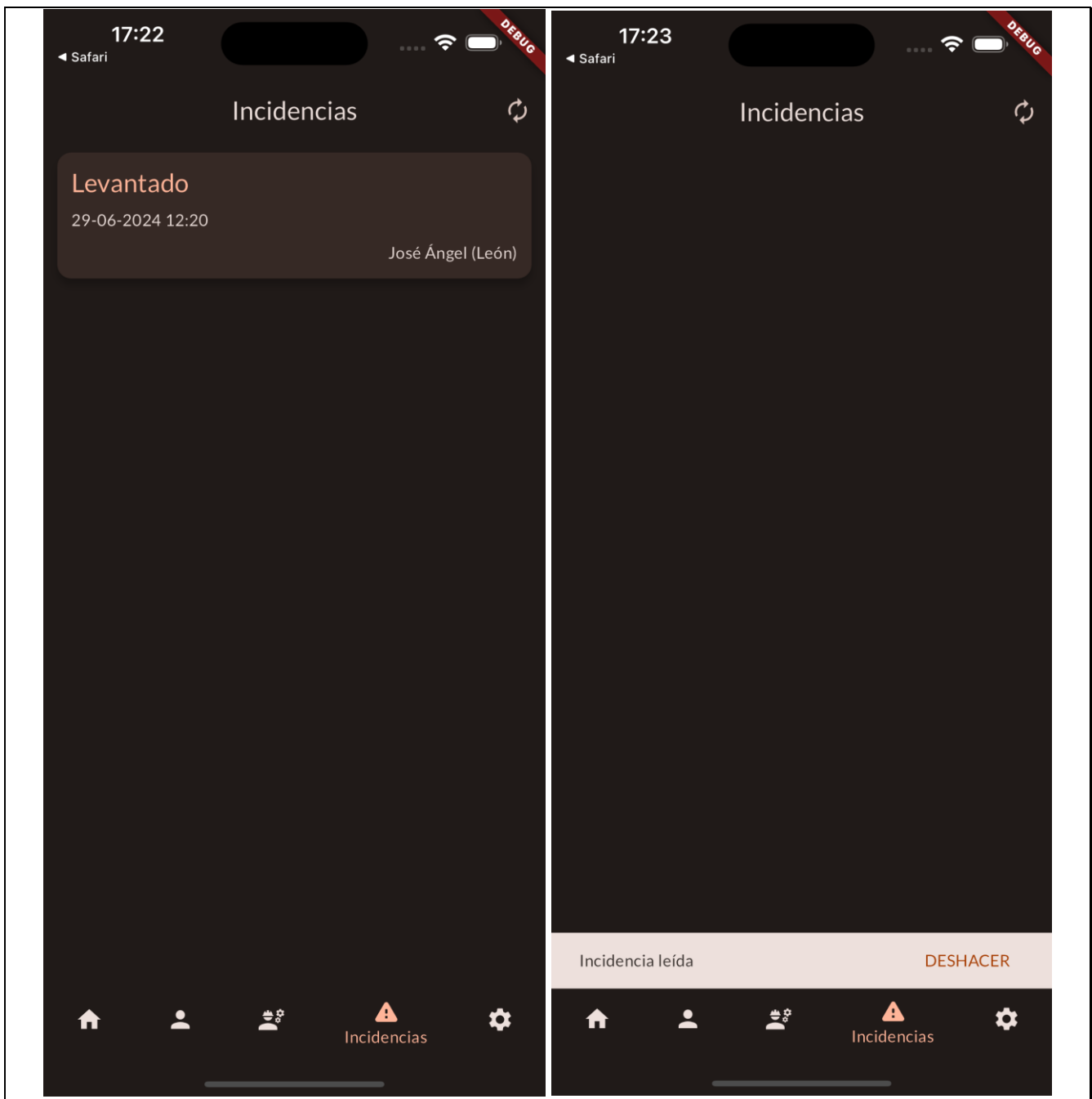
8.5.2.3 Información detallada y control remoto

Los robots asignados se pueden visualizar en la lista principal de robots, primera opción de la barra de navegación. Para cada robot se muestra información básica, como el estado actual, la batería, número de serie y modelo, así como el nombre del cliente y del robot. Para ver más información de dicho robot, hacemos clic en el robot y la aplicación nos muestra la vista de información detallada. Aquí se nos muestra toda la información del robot, un estado actual más detallado, el cliente y el empleado asignados. Por último, tenemos la zona de control de remoto, la cual tiene 3 operaciones distintas. Aparcar, la cual manda al robot a cargar a su base; Pausar, la cual hace que el robot se pare si está segando y la última, Segar, la cual hace que el robot empiece a segar, ya esté parado o en la base de carga.



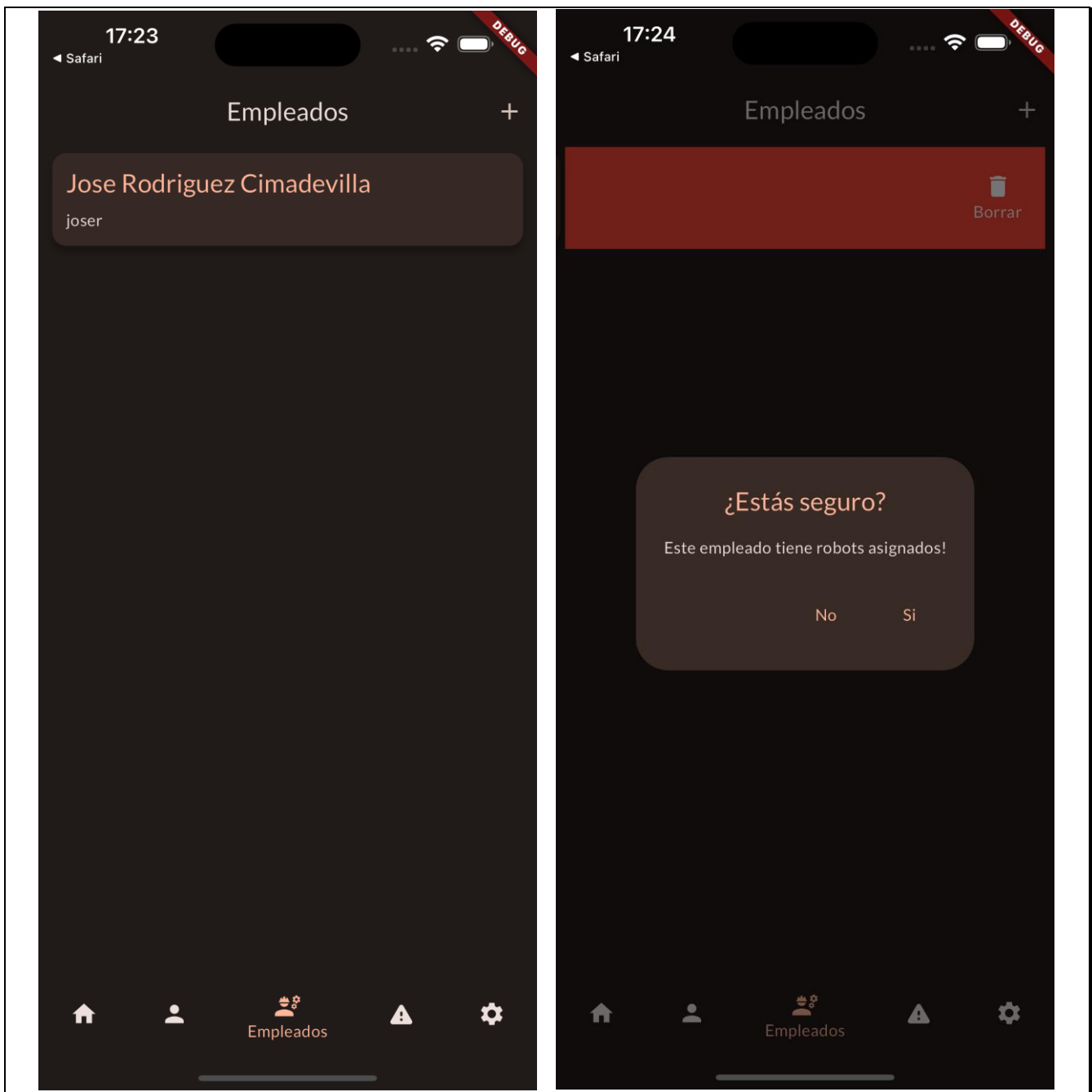
8.5.2.4 Incidencias

Para ver la lista de incidencias, usamos la cuarta opción de la barra de navegación. Ahí se nos muestra la vista de incidencias, para cada una podemos ver el mensaje de error, la fecha y la hora y el nombre del robot que causó la incidencia. Podemos marcar como leída una incidencia, para esto debemos deslizar hacia la derecha la que queramos marcar. Una vez, hagamos esto la incidencia se marca como leída, pero en caso de querer deshacer la operación, en la parte inferior, nos sale un botón para realizar dicha acción de deshacer. En caso de que se marque como leída, la incidencia dejará de aparecer en la lista.



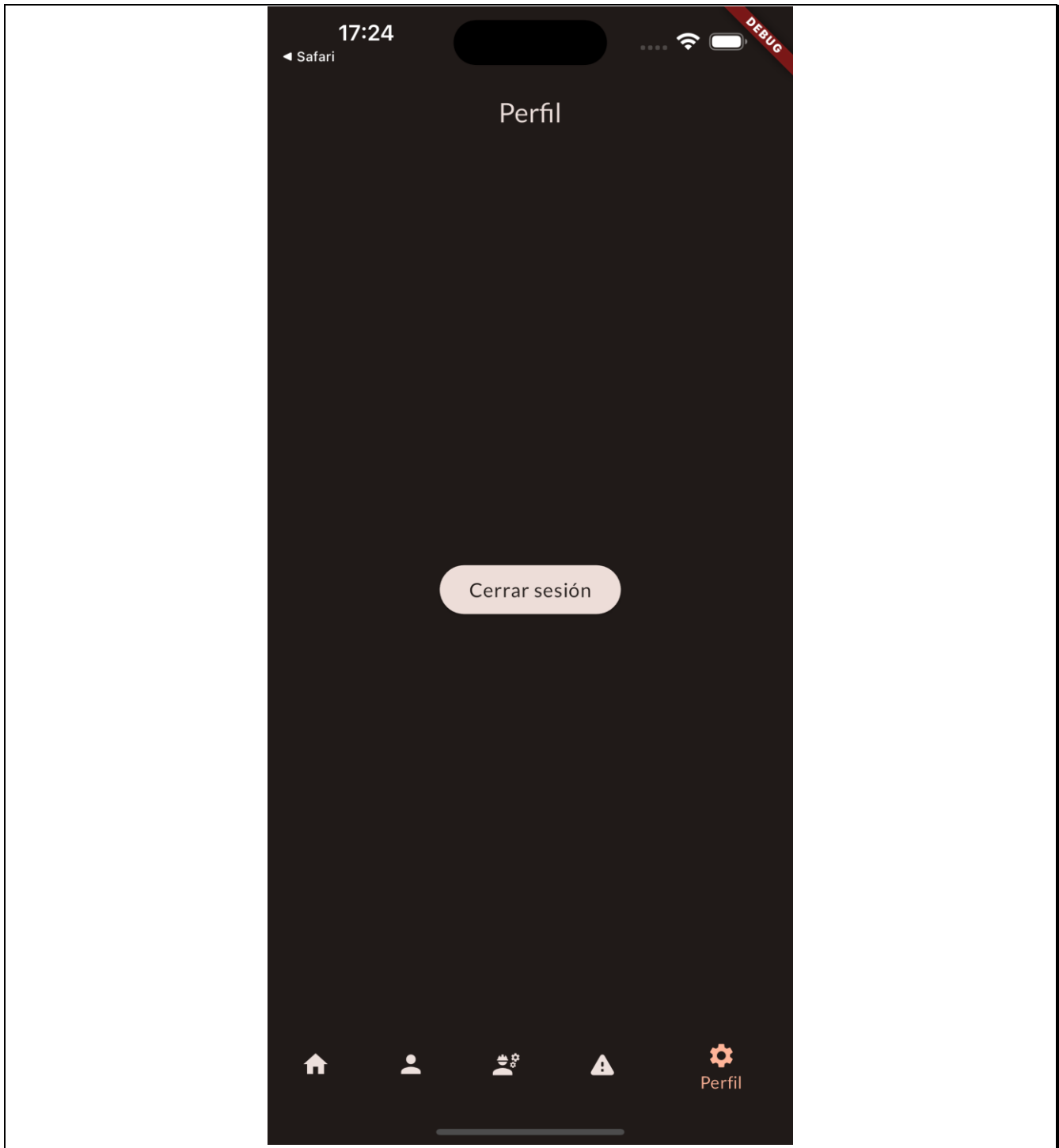
8.5.2.5 Borrar cliente y/o empleado

El procedimiento para borrar un cliente y empleado es el mismo, por lo que en este manual solo se enseñará como borrar un empleado. Para esto debemos irnos a la vista de empleados, tercera opción de la barra de navegación. Una vez ahí, buscamos el empleado que queremos borrar y lo deslizamos para la izquierda. A continuación, se abre un dialogo de confirmación, que nos indica si el empleado tiene robots asignados, ya que en caso de borrar un empleado o cliente con robots asignados, dichos robots se desasignarán. Si se confirma el borrado, el empleado desaparece de la lista.



8.5.2.6 Cerrar sesión

Por último, para poder cerrar sesión de la aplicación, hacemos clic en la última opción de la barra de navegación. Esto nos muestra la vista de Perfil y haciendo clic en el botón de “Cerrar sesión”, la sesión es cerrada y la aplicación nos muestra la vista inicial de iniciar sesión.



8.5.3 Manual del Programador

En este apartado, se redacta el manual del programador, que contiene información básica pero relevante para que un programador externo pueda empezar a trabajar con el sistema.

8.5.3.1 REST API – Backend

Para empezar a trabajar y a desarrollar nuevas funcionalidades en la API, se necesita tener instalado Node.js en el sistema donde se vaya a trabajar, así como un IDE para la edición del código, se recomienda Visual Studio Code.

Además, para poder probar la API de manera local, se necesita tener una instancia de MySQL corriendo en el sistema. Para facilitar esto, hay que tener instalado Docker y en el proyecto se entrega un archivo *docker-compose.yml*, que nos ejecuta una instancia de la API así como una instancia de MySQL.

A continuación, se detalla la estructura básica y archivos más relevantes de la REST API:

- Archivo *app.ts* – Contiene la lógica de inicialización de la API, se definen las rutas que tiene, así como un global error handler, que gestiona todas las excepciones de la aplicación y devuelve el código de error adecuado.
- Directorio */src* – Contiene todo el código de los diferentes componentes de la API, se dividen en diferentes carpetas.
 - Directorio */controllers* – Contiene todos los controladores de la API
 - Directorio */db* – Contiene los scripts de inicialización de la DB, así como la lógica de conexión.
 - Directorio */errors* – Se definen los diferentes errores o excepciones de la API.
 - Directorio */helpers* – Contiene métodos auxiliares que se reutilizan en diferentes partes del código.
 - Directorio */middleware* – Contiene las clases que actúan como middleware entre la petición y el controlador.
 - Directorio */models* – Contiene las clases que definen los modelos que se usan en el sistema.
 - Directorio */repositories* – Contiene las clases que ejecutan la lógica de consulta o modificación de la base de datos, a través de consultas SQL.
 - Directorio */routes* – Contiene las clases que definen las rutas o endpoints que tiene la API, así como el método de cada una.
 - Directorio */services* – Contiene las clases que definen la lógica más compleja del sistema y que se hace en un servicio aparte.
 - Directorio */tests* – Contiene los tests unitarios y de integración.

8.5.3.2 Aplicación móvil – Frontend

Es necesario tener instalado Flutter en el sistema y un IDE para el desarrollo del código, se recomienda al igual que para el backend Visual Studio Code. Además, para ejecutar la aplicación de manera local, se exige un simulador, ya sea de Android o de iOS.

Para poder conectarse a la API y probar las funcionalidades de manera completa, es necesario modificar la variable de entorno `MOWER_CONTROL_API_URL` en el archivo `.env`, con la dirección donde se encuentre ejecutándose la REST API.

A continuación, se detalla la estructura básica y archivos más relevantes de la aplicación:

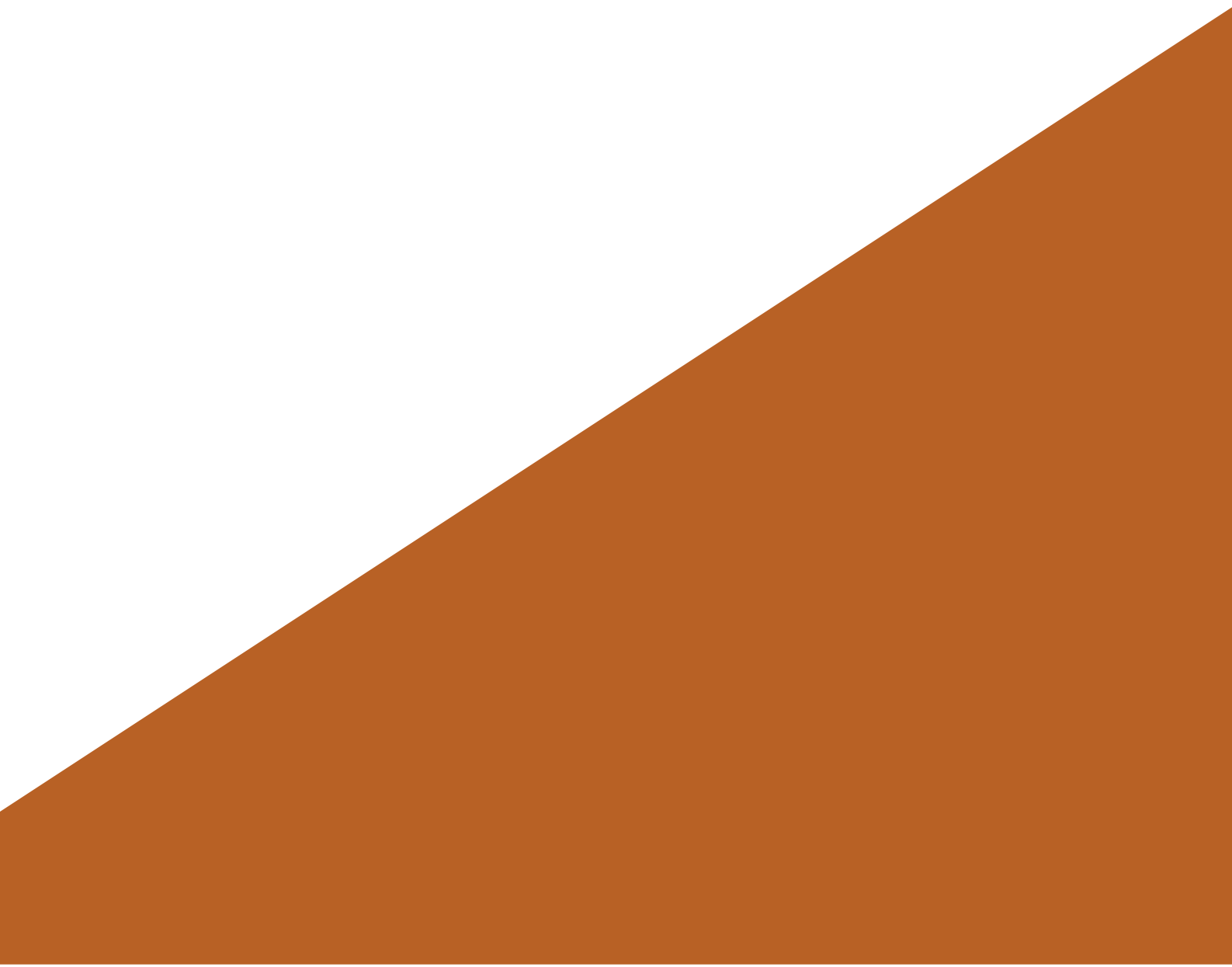
- Directorio `/lib` – Contiene todo el código y archivos más importantes de la aplicación móvil. Se divide en varias carpetas y archivos:
 - Archivo `main.dart` – Contiene la lógica de inicialización de la aplicación.
 - Directorio `/api` – Contiene toda la lógica de llamadas a la API.
 - Directorio `/models` – Contiene todas las clases que definen los modelos que usa la aplicación.
 - Directorio `/providers` – Contiene los providers que almacenan los datos de la aplicación y lanzan eventos cuando se actualizan.
 - Directorio `/screens` – Contiene los widgets que actúan como pantallas en la aplicación, se entiende por pantalla, cada una de las vistas completas de la aplicación.
 - Directorio `/utils` – Contiene clases de utilidad con métodos auxiliares.
 - Directorio `/widgets` – Contiene algunos de los widgets que se utilizan en las pantallas y que se reutilizan entre ellas.

8.5.3.3 Repositorios de GitHub

Como ya se explicó en el apartado [8.1.2.4 Documentación y control de versiones](#), se utilizó GitHub para almacenar tanto el código de la API como de la aplicación móvil. A continuación, se encuentran la dirección web a dichos repositorios:

- REST API – Backend: https://github.com/alvarorg14/Mower_Control_API
- Aplicación móvil – Frontend: https://github.com/alvarorg14/Mower_Control_App

Capítulo 9 CONCLUSIONES Y AMPLIACIONES





9.1 CONCLUSIONES

El proyecto se ha completado satisfactoriamente pese a los retrasos en la planificación. Una mejor planificación inicial y la elección de otra tecnología para la parte de frontend, podrían haber acortado el tiempo necesario para desarrollar el sistema. Aun así, el sistema es totalmente funcional y ha cumplido todos los requisitos obtenidos inicialmente.

Más concretamente, se ha desarrollado una aplicación móvil plataforma que permitirá a las empresas distribuidoras de robots cortacésped, llevar un control y una gestión mucho más sencilla e informatizada de todos los robots instalados a sus clientes. Además, es perfectamente usable por cualquier empresa, no solo para la empresa con la que surgió la idea, ya que la aplicación soporta múltiples empresas.

En la construcción de este proyecto, se han utilizado y aplicado los conocimientos de muchas de las asignaturas del grado. Dirección y Planificación de Proyectos Informáticos, para el proceso general del proyecto, así como la planificación, presupuestos y riesgos. Ingeniería de Requisitos, para la obtención de requisitos, casos de uso y escenarios. Arquitectura del software, para el desarrollo completo de un sistema, diagramas de secuencia y despliegue, así como el desarrollo de una documentación extensa. Sistema Distribuidos e Internet, para el desarrollo de APIs y uso de Node.js. Calidad, Validación y Verificación del Software, para todo el plan de pruebas, así como su ejecución. Además, del resto de asignaturas del grado relacionadas con la programación y diseño de software. Por último, Aspectos Sociales, Legales, Éticos y Profesionales de la Informática, para el tema de protección de datos y normativa a cumplir.



9.2 AMPLIACIONES

La siguiente ampliación del sistema más prioritaria, sería la implantación al 100% en una empresa real, ya que ahora mismo solo se ha usado para pruebas en casos concretos. Esta implantación y uso en una empresa, seguramente haga que salgan fallos y mejoras que se puedan hacer, así como obtener un feedback de los usuarios que estén usando la aplicación.

En cuanto a nuevas funcionalidades, existen bastantes que se podrían llevar a cabo para hacer una aplicación de gestión más completa. La API externa permite muchas más acciones para funcionar con los robots, aparte de las tres que ya se han implementado. Nos permitiría modificar el calendario de siega de cada robot, así como otros ajustes como la altura de corte y ver más estadísticas del funcionamiento total de cada robot.

Una mejora bastante importante para la aplicación sería la gestión de reparaciones. Esta nueva funcionalidad permitiría a los empleados registrar las reparaciones que ha tenido cada robot, para así tener disponible un historial de reparaciones y ver de manera fácil los problemas que ha podido tener un robot, así como el arreglo que se ha llevado a cabo.

Otra mejora relacionada con la anterior sería la gestión de recambios. Esto permitiría a las empresas llevar un control de las piezas que tienen disponibles, ya que tendrían el stock y el precio de cada recambio dentro de la aplicación. Con esto la gestión de reparaciones sería mucho más detallada, ya que cada empleado podría seleccionar las piezas o recambios que ha usado para una reparación, el stock se actualizaría automáticamente y se podría también generar el importe final que tendría que abonar el cliente por la reparación.

Por último, una mejora que no lleva relacionado una nueva funcionalidad es el cambio de cómo se lee la información de la API externa. Esta API está empezando a proveer la funcionalidad de leer la información a través de WebSockets y no con llamadas HTTPS. Esto mejoraría el tiempo de actualización del estado de los robots y posibles incidencias, ya que se estaría escuchando siempre por nuevas actualizaciones y no hay que esperar hasta que se haga la llamada a la API.

Capítulo 10 APÉNDICES





10.1 PLAN DE GESTIÓN DE RIESGOS

El siguiente es el plan de gestión de riesgos diseñado para proteger el proyecto de posibles eventos adversos y maximizar las oportunidades positivas.

Para su elaboración, se empleará la metodología de Boehm [3], complementada con una matriz de probabilidad e impacto extraída del PMBOK [4] de 2013.

10.1.1 Valoración de riesgos

La evaluación de riesgos se llevará a cabo antes de iniciar la fase de desarrollo y se continuará actualizando conforme se identifiquen nuevos riesgos. Este proceso se divide en tres secciones: la identificación de riesgos, su análisis y, finalmente, su priorización.

10.1.1.1 *Identificación de riesgos*

Existen múltiples técnicas para identificar riesgos en un proyecto informático. Entre las más utilizadas se encuentran:

- **Tormenta de ideas** (brainstorming): Se organiza una reunión entre los miembros del proyecto donde cada uno presenta varios riesgos identificados. A partir de estos casos, los participantes discuten y generan nuevos riesgos potenciales.
- **Análisis FODA** (Fortalezas, Oportunidades, Debilidades, Amenazas): Evaluar las fortalezas, oportunidades, debilidades y amenazas del proyecto para identificar riesgos inherentes y externos.
- **Entrevistas y talleres**: Se llevan a cabo conversaciones con expertos en la materia, miembros del equipo de proyecto y otros interesados, para recopilar información sobre posibles riesgos.
- **Listas de verificación** (checklists): Utilizar listas predefinidas de riesgos comunes en proyectos similares para asegurarse de que no se pasan por alto riesgos importantes.

10.1.1.2 *Análisis de riesgos*

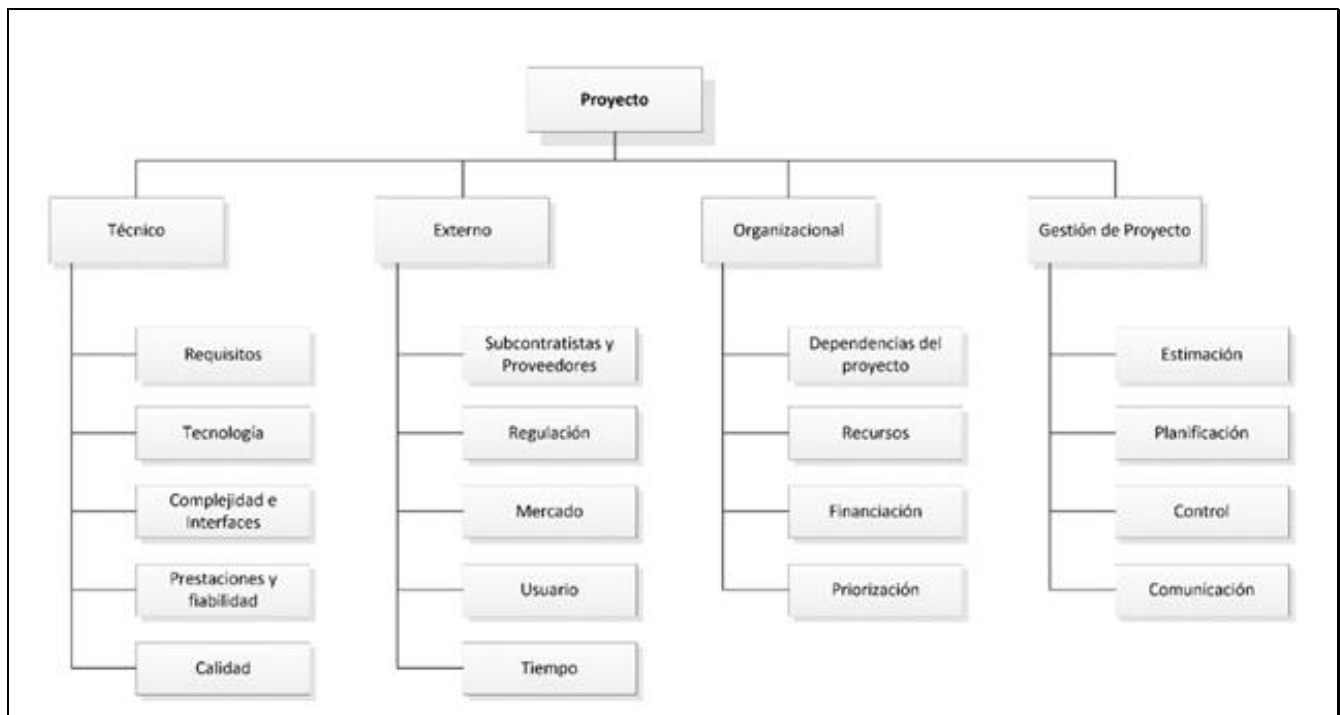
El análisis de riesgos es fundamental para asegurar el éxito de un proyecto. Realizar un análisis de riesgos exhaustivo permite evaluar la probabilidad y el impacto de los riesgos identificados, facilitando la toma de decisiones informadas sobre cómo gestionarlos. Hay diversos tipos de análisis de riesgos:

- **Análisis cuantitativo de riesgos:** El análisis cuantitativo emplea datos numéricos y técnicas estadísticas para estimar la probabilidad y el impacto de los riesgos. Este enfoque puede proporcionar una visión más objetiva y precisa de los riesgos, aunque su implementación puede ser más compleja y costosa.
- **Análisis cualitativo de riesgos:** Al contrario que el anterior, este método se basa en una evaluación subjetiva de la probabilidad y el impacto de los riesgos, utilizando la experiencia y el conocimiento de expertos en el campo. Aunque no es tan preciso como el análisis cuantitativo, es útil para identificar rápidamente los riesgos más críticos que necesitan atención inmediata.

Las actividades esenciales en el análisis de riesgos incluyen la evaluación y clasificación de estos. Es necesario cuantificar varios aspectos de un riesgo: la probabilidad, el impacto, la importancia, la duración y el marco de tiempo. Además, es fundamental identificar y mejorar los factores de riesgo e indicadores asociados.

- **Probabilidad:** Representa el porcentaje de posibilidades de que un riesgo ocurra.
- **Impacto:** Mide el efecto que el riesgo puede tener en el proyecto, ya sea negativo (amenaza) o positivo (oportunidad).
- **Importancia:** Combina la probabilidad y el impacto para determinar la relevancia del riesgo en el proyecto.
- **Duración:** Considera el momento en que el impacto ocurre y su duración.
- **Marco de tiempo:** Establece el periodo en el cual se deben tomar acciones para mitigar el riesgo.

Los riesgos también se pueden clasificar en las siguientes categorías:



10.1.1.3 Probabilidad e impacto

Para cada riesgo identificado, es necesario evaluar tanto su probabilidad de ocurrencia como su impacto potencial en el proyecto en caso de materializarse. La probabilidad se indica en porcentaje y puede fundamentarse en experiencias anteriores, datos históricos o la opinión de expertos en la materia. El impacto, en cambio, puede cuantificarse en términos de costo, tiempo o calidad, dependiendo del aspecto del proyecto que podría verse comprometido.

El cálculo de la probabilidad se basará en la siguiente tabla, dependiendo de cada característica del proyecto:

| Condiciones definidas para las escalas de impacto de un riesgo sobre los objetivos principales del proyecto (Se muestran ejemplos para impactos negativos únicamente) | | | | | |
|--|--|--|---|--|--|
| Objetivos de proyecto | Escala relativa o numérica | | | | |
| | Muy bajo / 5% | Bajo / 10% | Moderado / 20% | Alto / 40% | Muy alto / 80% |
| Coste | Incremento del coste insignificante | Incremento del coste <10% | Incremento del coste entre el 10-20% | Incremento del coste entre el 20-40% | Incremento del coste >40% |
| Tiempo | Incremento de tiempo insignificante | Incremento de tiempo <5% | Incremento de tiempo entre el 5-10% | Incremento de tiempo entre el 10-20% | Incremento de tiempo >20% |
| Alcance | Reducciones del alcance inapreciables | Afectadas áreas poco importantes del alcance | Afectadas áreas importantes del alcance | Reducciones del alcance inaceptables para el cliente | El resultado final del proyecto no es realmente útil |
| Calidad | La degradación de la calidad es inapreciable | Sólo las aplicaciones muy exigentes se ven afectadas | La reducción de la calidad requiere la aceptación del cliente | Reducción de la calidad inaceptable para el cliente | El resultado final del proyecto no es realmente útil |

Esta tabla presenta ejemplos de definiciones de impacto para cuatro objetivos de proyecto diferentes. Debe ser ajustada en el proceso de elaboración del Plan de Riesgos a cada proyecto concreto y a los umbrales de riesgo de la organización. Las definiciones del impacto deben ser desarrolladas para los riesgos positivos (oportunidades) de una manera similar.

Una vez calculada la probabilidad, se puede emplear la siguiente matriz de probabilidad e impacto para priorizar los riesgos:

| Matriz de Probabilidad e Impacto | | | | | | | | | | |
|---|-------------------------|---------------------|-------------------------|---------------------|-------------------------|-------------------------|---------------------|-------------------------|---------------------|-------------------------|
| Probabilidad | Amenazas | | | | | Oportunidades | | | | |
| 0,90 | 0,05 | 0,09 | 0,18 | 0,36 | 0,72 | 0,72 | 0,36 | 0,18 | 0,09 | 0,05 |
| 0,70 | 0,04 | 0,07 | 0,14 | 0,28 | 0,56 | 0,56 | 0,28 | 0,14 | 0,07 | 0,04 |
| 0,50 | 0,03 | 0,05 | 0,10 | 0,20 | 0,40 | 0,40 | 0,20 | 0,10 | 0,05 | 0,03 |
| 0,30 | 0,02 | 0,03 | 0,06 | 0,12 | 0,24 | 0,24 | 0,12 | 0,06 | 0,03 | 0,02 |
| 0,10 | 0,01 | 0,01 | 0,02 | 0,04 | 0,08 | 0,08 | 0,04 | 0,02 | 0,01 | 0,01 |
| | 0,05 Muy bajo | 0,10 Bajo | 0,20 Moderado | 0,40 Alto | 0,80 Muy Alto | 0,80 Muy Alto | 0,40 Alto | 0,20 Moderado | 0,10 Bajo | 0,05 Muy Bajo |
| | Impacto Negativo | | | | | Impacto Positivo | | | | |

Cada riesgo es evaluado de acuerdo a la probabilidad de que ocurra y al impacto en algún objetivo si ocurriera. Los umbrales de tolerancia de cada organización se trasladan a la matriz, de manera que las áreas verde, amarilla y roja indiquen estos umbrales para la priorización de riesgos.

10.1.2 Gestión de riesgos

En este apartado, se definirán estrategias y acciones concretas para gestionar los riesgos priorizados identificados en las fases previas.

10.1.2.1 Priorización de Riesgos

A continuación, se priorizan los riesgos definidos en el apartado [5.1.4.2 Identificación de Riesgos](#) siguiendo la metodología explicada en el apartado anterior.

| Nombre del Riesgo | Categoría | Probabilidad | Impacto | | | | Impacto |
|---------------------------------------|---------------------------|--------------|---------|----------|---------|---------|---------|
| | | | Presup. | Planifc. | Alcance | Calidad | |
| <i>Estimación inexacta</i> | Estimación | Alta | Alto | Crítico | Medio | Medio | 0,63 |
| <i>Ambigüedad en los requisitos</i> | Requisitos | Alta | Medio | Alto | Alto | Alto | 0,39 |
| <i>Falta de capacitación</i> | Usuario | Alta | Medio | Medio | Alto | Alto | 0,39 |
| <i>Cambio en software de terceros</i> | Proveedores | Media | Medio | Alto | Alto | Alto | 0,28 |
| <i>Brecha de seguridad</i> | Prestaciones y fiabilidad | Media | Alto | Medio | Bajo | Medio | 0,28 |
| <i>Mala integración de sistemas</i> | Complejidad e interfaces | Baja | Alto | Crítico | Medio | Alto | 0,27 |
| <i>Inmadurez tecnológica</i> | Tecnología | Baja | Medio | Alto | Medio | Alto | 0,17 |
| <i>Cambios regulatorios</i> | Regulación | Baja | Medio | Alto | Bajo | Bajo | 0,17 |
| <i>Adopción baja</i> | Usuario | Media | Medio | Bajo | Medio | Medio | 0,15 |
| <i>Nuevos competidores</i> | Mercado | Media | Medio | Bajo | Medio | Medio | 0,15 |

10.1.2.2 Resolución de Riesgos

A continuación, se explican la estrategia que se seguirá para cada riesgo, así como una pequeña explicación de una posible respuesta contra el riesgo.

| Nombre del Riesgo | Estrategia | Respuesta / Acción |
|--|--------------------|---|
| <i>Estimación inexacta</i> | Mitigar el riesgo | Involucrar a todo el equipo en el proceso de estimación para obtener diferentes perspectivas y mejorar la aplicación. Para así poder realizar una planificación detallada y con revisiones frecuentes. |
| <i>Ambigüedad en los requisitos</i> | Mitigar el riesgo | Establecer un proceso de gestión de requisitos claro y continuo, con validaciones regulares y comunicación constante con los stakeholders. |
| <i>Falta de capacitación</i> | Eliminar el riesgo | Desarrollar un programa de formación para los usuarios finales. Crear tutoriales y manuales de usuario que cubran todas las funcionalidades de la aplicación. |
| <i>Cambio en el software de terceros</i> | Mitigar el riesgo | Abstraer lo máximo posible el sistema desarrollado de la posible fuente de los datos, así en caso de cambio de este software de terceros afecte lo menos posible al sistema desarrollado. |
| <i>Brecha de seguridad</i> | Mitigar el riesgo | Implementar practicas robustas de seguridad durante el desarrollo, así como realizar pruebas extensas sobre este tema. Además, llevar un control sobre las vulnerabilidades que pueda tener tanto el sistema como los proveedores externos. |
| <i>Mala integración de sistemas</i> | Mitigar el riesgo | Planificar y realizar pruebas de integración exhaustivas desde las primeras etapas del desarrollo. Utilizar estándares de comunicación y protocolos comunes para facilitar la integración. |
| <i>Inmadurez tecnológica</i> | Mitigar el riesgo | Capacitación y formación en las nuevas tecnologías. Realizar pruebas exhaustivas y crear prototipos para validar la viabilidad de las tecnologías en el proyecto. |
| <i>Cambios regulatorios</i> | Mitigar el riesgo | Mantenerse informado sobre las leyes y regulaciones relevantes y planificar revisiones periódicas del cumplimiento. Incorporar flexibilidad en el diseño de la aplicación para facilitar ajustes en caso de cambios de normativa. |



| | | |
|--------------------------------|----------------------|---|
| <i>Adopción baja</i> | Mitigar el riesgo | Desarrollar campañas de marketing efectivas. Implementar un programa de feedback continuo para mejorar la aplicación según las opiniones de los usuarios y aumentar su atractivo y utilidad. |
| <i>Nuevos competidores</i> | Asumir el riesgo | Monitorear continuamente el mercado y realizar análisis de la competencia. Estar preparado para desarrollar nuevas funcionalidades. |



10.2 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. M. Redondo, «Documentos-modelo para Trabajos de Fin de Grado/Máster de la Escuela de Informática de Oviedo,» 17 6 2019. [En línea]. Available: https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo.
- [2] J. Redondo, «Creación y evaluación de plantillas para trabajos de fin de grado como buena práctica docente.,» *Revista de Innovación y Buenas Prácticas Docentes*, p. pp, 2020.
- [3] B. Boehm, "A Spiral Model of Software Development and Enhancement", *IEEE Computer*. 21 (5), p. 61–72, 05/1988.
- [4] Project Management Institute, "PMBOK Guide, 5th edition", 2013. ISBN 978-1-935589-67-9

10.3 CONTENIDO ENTREGADO EN LOS ANEXOS

En este apartado, se detalle la estructura y los contenidos del proyecto entregados junto a esta memoria.

10.3.1 Contenidos

| Directorio | Contenido |
|------------|---|
| ./ | <p>Contiene un fichero README.TXT explicando toda esta estructura e instrucciones para instalar la aplicación.</p> <p>Contiene un archivo Mower_Control.zip, el cual contiene todo el código desarrollador para el proyecto, especificado con más detalle en el apartado 10.2.1.1 Contenido del archivo comprimido del código.</p> <p>Contiene un archivo Mower_Control.apk, para instalar la aplicación en un dispositivo móvil con sistema operativo Android.</p> |

10.3.2 Contenido del archivo comprimido del código

| Directorio | Contenido |
|---------------------|---|
| ./ | Contiene un fichero README.TXT explicando toda esta estructura. |
| ./Mower_Control_API | Contiene todo el código desarrollado para la REST API, explicado con más detalle en el apartado 10.2.2.1 Contenido del directorio backend . |
| ./Mower_Control_App | Contiene todo el código desarrollado para la App explicado con más detalle en el apartado 10.2.2.2 Contenido del directorio frontend . |



10.3.2.1 Contenido del directorio backend

| Directorio | Contenido |
|---|--|
| <code>./github/workflows</code> | Contiene las GitHub Actions que se usan para ejecutar los tests y desplegar la API en el servidor remoto. |
| <code>./src</code> | Contiene todo el código de la API, dividido en varias subcarpetas dependiendo de las responsabilidades, las principales son <i>controllers</i> , <i>services</i> y <i>repositories</i> . |
| <code>Dockerfile</code> y <code>docker-compose.yml</code> | Ficheros para la creación de la imagen de Docker, así como para levantar la API junto a una instancia de MySQL de manera local. |
| <code>Package.json</code> | Fichero que contiene las dependencias que se utilizan en la API. |

10.3.2.2 Contenido del directorio frontend

| Directorio | Contenido |
|---------------------------|---|
| <code>./android</code> | Contiene archivos necesarios para la generación de la aplicación para el sistema operativo Android. |
| <code>./assets</code> | Contiene archivos como imágenes que se usan en la aplicación. |
| <code>./ios</code> | Contiene archivos necesarios para la generación de la aplicación para el sistema operativo iOS. |
| <code>./lib</code> | Contiene todo el código desarrollador para la APP. Divido en diferentes directorios, el principal es <i>screens</i> , el cual define las vistas principales de la aplicación móvil. |
| <code>Pubspec.yaml</code> | Fichero que contiene las dependencias que se utilizan en la APP. |