



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Mención en Ingeniería de Software

**Aplicación web de gestión de recursos humanos relativo a la
administración de horarios, equipos y permisos**

Alumno: Jorge De Bustos Pérez

Tutor: Margarita Gonzalo Tasis

Agradecimientos

A mi familia y pareja, que ha sabido apoyarme y darme fuerzas durante tanto durante mis años de estudio como durante la elaboración del proyecto.

A mis amigos, por su apoyo, ánimo y preocupación de diversas formas a lo largo de este periodo.

A mi tutora Margarita, por su entusiasmo, sus consejos y su implicación para conseguir este proyecto.

Gracias a todos

Resumen

Este proyecto está enfocado en el ámbito de la gestión de recursos humanos como una solución que permite centralizar y facilitar, por un lado, la creación de equipos y de ubicaciones de trabajo, permitiendo así a los usuarios asignados a estas, realizar fichajes laborales fiables y geolocalizados, y por otro lado, la configuración de horarios laborales personalizables, con su posterior sistema de cambio de horarios, permitiendo así a los propios trabajadores gestionarlo de una manera sencilla y rápida.

Esta plataforma se ha visto motivada por la escasez de soluciones que permitan realizar estos procesos a empresas con un sistema de trabajo que permita a sus trabajadores a abandonar presencialmente la sede de esta para realizar el trabajo, como pueden ser, empresas de la construcción, profesionales de ventas etc.

Dicho proyecto se ha desarrollado como una WebApp compuesta por un frontend hecho en Angular y por un backend en Spring Boot.

Abstract

This project is focused on the field of human resources management as a solution that centralizes and facilitates, on the one hand, the creation of teams and work locations, allowing users assigned to these, making reliable and geolocated work records, and on the other hand, the configuration of customizable work schedules, with its subsequent system of schedule changes, allowing workers themselves to manage it in a simple and fast way.

This platform has been motivated by the absence of solutions that allow these processes to companies with a work system that requires their workers to leave their headquarters in person, such as construction companies, sales professionals, etc.

This project has been developed as a WebApp composed of a frontend made in Angular and a backend in Spring Boot.

Índice

1.	INTRODUCCIÓN.....	15
1.1.	CONTEXTO.....	15
1.2.	MOTIVACIÓN.....	15
1.3.	OBJETIVOS.....	15
1.4.	PLATAFORMAS SIMILARES.....	16
2.	PLANIFICACIÓN.....	19
2.1.	SCRUM.....	19
2.1.1.	<i>Definición.....</i>	19
2.1.2.	<i>Características de Scrum.....</i>	19
2.1.3.	<i>Aplicación al proyecto.....</i>	21
2.2.	PLANIFICACIÓN IDEAL.....	21
2.3.	ANÁLISIS DE RIESGOS.....	22
2.3.1.	<i>Tipos de riesgos.....</i>	22
2.3.2.	<i>Tipos de planes.....</i>	23
2.3.3.	<i>Categorización de los riesgos, probabilidades e impactos.....</i>	23
2.3.4.	<i>Riesgos.....</i>	25
2.4.	PRESUPUESTO.....	29
2.4.1.	<i>Presupuesto Simulado.....</i>	29
2.4.2.	<i>Presupuesto real.....</i>	30
3.	ANÁLISIS.....	31
3.1.	ANÁLISIS DE REQUISITOS.....	31
3.1.1.	<i>Historias de usuario.....</i>	31
3.1.2.	<i>Requisitos Funcionales.....</i>	32
3.1.3.	<i>Requisitos No Funcionales.....</i>	33
3.1.4.	<i>Requisitos de Información.....</i>	34
3.1.5.	<i>Reglas de Negocio.....</i>	35
3.2.	CASOS DE USO.....	36
3.2.1.	<i>Diagrama de casos de uso.....</i>	36
3.2.2.	<i>Especificación de casos de uso.....</i>	38
3.3.	MODELO DE DOMINIO.....	48
3.4.	DIAGRAMAS DE SECUENCIA.....	50
4.	DISEÑO.....	55
4.1.	PATRÓN ARQUITECTÓNICO MVVM.....	55
4.2.	PATRONES DE DISEÑO.....	56
4.2.1.	<i>Patrón DAO/DTO.....</i>	56
4.2.2.	<i>Patrón Repository.....</i>	56
4.3.	DESARROLLO BASADO EN COMPONENTES.....	57
4.3.1.	<i>Componentes en Angular.....</i>	57
4.3.2.	<i>Componentes creados en el frontend.....</i>	58
4.4.	DISEÑO ARQUITECTÓNICO DEL BACKEND.....	61
4.5.	DISEÑO DE BASE DE DATOS.....	65
4.6.	DESPLIEGUE DE PLATAFORMA.....	67
4.7.	PRIVACY BY DESIGN.....	67
4.8.	INTERFAZ DE USUARIO.....	68
4.8.1.	<i>Atributos de usabilidad.....</i>	68
4.8.2.	<i>Bocetos de interfaz de usuario.....</i>	69
5.	TECNOLOGÍA UTILIZADA.....	71

5.1.	MICROSOFT 365	71
5.2.	VISUAL STUDIO CODE	71
5.3.	ASTAH PROFESIONAL	71
5.4.	GIT	71
5.5.	POSTMAN.....	72
5.6.	ANGULAR	72
5.7.	SPRING BOOT.....	73
5.8.	JWT.....	73
5.9.	HEIDISQL.....	73
5.10.	MARIADB	73
6.	IMPLEMENTACIÓN Y PRUEBAS.....	75
6.1.	ESTRUCTURA DEL CÓDIGO FRONTEND.....	75
6.2.	ESTRUCTURA DEL CÓDIGO BACKEND.....	77
6.3.	PRUEBAS DE CAJA NEGRA.....	79
6.4.	EVALUACIÓN DE USABILIDAD	86
6.4.1.	<i>Responsable</i>	86
6.4.2.	<i>Trabajador</i>	87
7.	SEGUIMIENTO DEL PROYECTO	89
8.	CONCLUSIONES	91
8.1.	CONCLUSIONES.....	91
8.2.	LÍNEAS DE TRABAJO FUTURAS	91
	BIBLIOGRAFÍA.....	93
I.	MANUAL DE INSTALACIÓN.....	97
I.I.	MANUAL DE DESPLIEGUE DE BACKEND	97
I.II.	MANUAL DE DESPLIEGUE DE FRONTEND	98
II.	MANUAL DE USUARIO.....	99
II.I.	INICIO DE SESIÓN	99
II.II.	INICIO	99
II.III.	USUARIOS.....	101
II.IV.	EQUIPOS	103
II.V.	UBICACIONES	105
II.VI.	TURNOS	108
II.VII.	FICHAJE.....	111
II.VIII.	HORARIO.....	112
II.IX.	INFORMES.....	116
III.	ENLACES DE CÓDIGO	117

Índice de Figuras

Figura 1.1: Logotipo Zoho People.....	16
Figura 1.2: Logotipo Workforce Now	17
Figura 3.1: Diagrama De Casos De Uso.....	37
Figura 3.2: Diagrama De Modelo De Dominio	49
Figura 3.3: Diagrama De Secuencia Del Cu Registrar Usuarios	50
Figura 3.4: Diagrama De Secuencia Del Cu Editar Perfil	50
Figura 3.5: Diagrama De Secuencia Del Cu Iniciar Sesión	51
Figura 3.6: Diagrama De Secuencia Del Cu Cerrar Sesión	51
Figura 3.7: Diagrama De Secuencia Del Cu Inhabilitar Usuario.....	52
Figura 3.8: Diagrama De Secuencia Del Cu Registrar Equipo.....	52
Figura 3.9: Diagrama De Secuencia Del Cu Editar Equipo.....	53
Figura 3.10: Diagrama De Secuencia Del Cu Eliminar Equipo	53
Figura 4.1: Componentes Patrón Mvvm [15].....	55
Figura 4.2: Patrón Dao/Dto	56
Figura 4.3: Patrón Repository.....	57
Figura 4.4: Estructura De Un Componente Angular	57
Figura 4.5: Estructura De La Reutilización De Un Componente Angular En Otro.....	58
Figura 4.6: Estructura De Un Componente Angular Que Requiere Un Servicio	58
Figura 4.7: Estructura De Componentes Del Frontend.....	60
Figura 4.8: Arquitectura General Del Backend	62
Figura 4.9: Modules&Uses Style Del Paquete Controller	62
Figura 4.10: Modules&Uses Style Del Paquete Repository	63
Figura 4.11: Modules&Uses Style Del Paquete Model Y Entities	63
Figura 4.12: Modules&Uses Style Del Paquete Requests	64
Figura 4.13: Modules&Uses Style Del Paquete Response	64
Figura 4.14: Modules&Uses Style Del Paquete Security	65
Figura 4.15: Modelo De Datos De La Aplicación	66
Figura 4.16: Diagrama De Despliegue	67
Figura 4.17: Paleta De Colores Utilizada En La Plataforma	69
Figura 4.18: Boceto De Interfaz De La Pantalla De Login.....	69
Figura 4.19: Boceto De Interfaz De La Pantalla De Escritorio	70
Figura 5.1: Esquema Funcionamiento Ramas Git [26].....	72
Figura 6.1: Estructura Código Frontend	76
Figura 6.2: Estructura Código Backend (General, Controllers, Entities, Repository)	78
Figura 6.3: Estructura Código Backend (Requests, Response, Security)	79
Figura A.1: Código Del Archivo Server.Js Del Proyecto Angular	98

Figura A.2: Pantalla De Inicio De Sesión.....	99
Figura A.3: Página De Inicio.....	100
Figura A.4: Pop-Up De Edición Del Perfil.....	100
Figura A.5: Página De Usuarios.....	101
Figura A.6: Página De Usuarios Con Usuario Inhabilitado.....	102
Figura A.7: Importación De Usuarios.....	102
Figura A.8: Página De Equipos.....	103
Figura A.9: Página De Creación De Un Equipo.....	103
Figura A.10: Página De Edición De Un Equipo.....	104
Figura A.11: Pop-Up De Eliminación De Un Equipo.....	104
Figura A.12: Página De Ubicaciones.....	105
Figura A.13: Página De Creación De Una Ubicación (1).....	105
Figura A.14: Página De Creación De Una Ubicación (2).....	106
Figura A.15: Página De Edición De Una Ubicación (1).....	106
Figura A.16: Página De Edición De Una Ubicación (2).....	107
Figura A.17: Pop-Up De Eliminación De Una Ubicación.....	108
Figura A.18: Página De Turnos.....	108
Figura A.19: Página De Creación De Turnos (1).....	109
Figura A.20: Página De Creación De Turnos (2).....	109
Figura A.20: Página De Edición De Turnos (1).....	110
Figura A.21: Página De Edición De Turnos (2).....	110
Figura A.22: Pop-Up De Eliminación De Turno.....	111
Figura A.23: Página De Fichaje.....	111
Figura A.24: Página De Horario.....	112
Figura A.25: Página De Mi Horario (1).....	113
Figura A.26: Página De Mi Horario (2).....	113
Figura A.27: Página De Cambios.....	114
Figura A.28: Página De Registrar Cambio.....	114
Figura A.29: Página De Contestación De Solicitud.....	115
Figura A.30: Solicitud En Tramite.....	115
Figura A.31: Pop-Up De Respuesta A Cambio.....	116
Figura A.32: Informes.....	116

Índice de Tablas

Tabla 2.1: Planificación Inicial.....	22
Tabla 2.2: Probabilidades De Los Riesgos [11]	24
Tabla 2.3: Impactos De Los Riesgos [11]	24
Tabla 2.4: Probabilidades X Impactos Pmbok [11].....	24
Tabla 2.5: Niveles De Riesgos [11].....	25
Tabla 2.6: R001 – No Disponibilidad Del Equipo Hardware	25
Tabla 2.7: R002 – Falta De Conocimiento O Experiencia Con Las Herramientas Utilizadas	26
Tabla 2.8: R003 – Falta De Disponibilidad De Miembro/S Del Proyecto	26
Tabla 2.9: R004 – Cambios En Los Requisitos	27
Tabla 2.10: R005 – Interpretación Errónea De Los Requisitos	27
Tabla 2.11: R006 – Error En Las Estimaciones	28
Tabla 2.12: R007 – Interfaz Gráfica De Usuario Poco Usable.....	28
Tabla 2.13: Resumen De Presupuesto Simulado.....	29
Tabla 2.14: Resumen De Presupuesto Real.....	30
Tabla 3.1: Requisitos Funcionales.....	33
Tabla 3.2: Requisitos No Funcionales	34
Tabla 3.3: Requisitos De Información.....	35
Tabla 3.4: Cu001 Iniciar Sesión	38
Tabla 3.5: Cu002 Cerrar Sesión	38
Tabla 3.6: Cu003 Editar Perfil.....	39
Tabla 3.7: Cu004 Inhabilitar Perfil.....	39
Tabla 3.8: Cu005 Importar Usuarios	40
Tabla 3.9: Cu006 Solicitar Cambio Jornada.....	40
Tabla 3.10: Cu007 Aceptar Solicitud De Cambio De Jornada	41
Tabla 3.11: Cu008 Ver Solicitud De Cambio De Jornada.....	41
Tabla 3.12: Cu009 Aceptar Respuesta De Cambio De Jornada	42
Tabla 3.13: Cu010 Realizar Fichaje	42
Tabla 3.14: Cu011 Registrar Equipo	43
Tabla 3.15: Cu012 Editar Equipo	44
Tabla 3.16: Cu013 Eliminar Equipo.....	44
Tabla 3.17: Cu014 Registrar Turno	45
Tabla 3.18: Cu015 Editar Turno.....	45
Tabla 3.19: Cu016 Eliminar Turno.....	46
Tabla 3.20: Cu017 Registrar Ubicación.....	46
Tabla 3.21: Cu018 Editar Ubicación	47
Tabla 3.22: Cu019 Eliminar Ubicación.....	47

Tabla 3.23: Cu020 Obtención De Informes.....	48
Tabla 6.1: Pcn-001 Iniciar Sesión.....	80
Tabla 6.2: Pcn-002 Cerrar Sesión.....	81
Tabla 6.3: Pcn-003 Editar Perfil.....	81
Tabla 6.4: Pcn-004 Importar Usuarios.....	81
Tabla 6.5: Pcn-005 Inhabilitar Perfil.....	82
Tabla 6.6: Pcn-006 Registrar Equipo.....	82
Tabla 6.7: Pcn-007 Editar Equipo.....	83
Tabla 6.8: Pcn-008 Eliminar Equipo.....	83
Tabla 6.9: Pcn-009 Registrar Fichaje.....	84
Tabla 6.10: Pcn-010 Solicitar Cambio De Turno.....	84
Tabla 6.11: Pcn-011 Aceptar Solicitud De Cambio De Turno.....	85
Tabla 6.12: Pcn-012 Aceptar Respuesta De Cambio De Turno.....	85
Tabla 6.13: Pcn-013 Obtención De Informes.....	86

Capítulo 1

1. Introducción

En este capítulo se expondrá el contexto de la aplicación, la motivación tras este proyecto, que aplicaciones similares existen ya en el mercado y la estructura de la memoria.

1.1. Contexto

En las últimas décadas, el mundo laboral ha experimentado cambios significativos en el modo en que se gestionan los recursos humanos en las organizaciones. Con el incremento de la tecnología en la digitalización de los procesos, las empresas han buscado mejorar la eficiencia y la productividad en la gestión del talento humano de la empresa. En este sentido, las aplicaciones de gestión de recursos humanos han puesto el énfasis en facilitar y agilizar tareas como la programación de horarios, control horario, la asignación de equipos y ubicaciones de trabajo.

Se pretende crear una solución web que permita centralizar, agilizar y facilitar procesos diarios relacionados con los recursos humanos de las empresas, y a su vez, demostrar así los conocimientos adquiridos durante los años de estudio realizando un proyecto software que pasará por todas sus fases de vida.

1.2. Motivación

La motivación de este proyecto viene dada porque nos hemos percatado que hay empresas de diversos sectores que utilizan un promedio de más de 15 herramientas software para gestionar procesos relacionados con el sector de recursos humanos [\[1\]](#), como pueden ser sistemas de gestión de nóminas, de fichajes laborales, de planificación horaria etc., obligadas así a adaptarse a plataformas de diferente ámbitos, funcionamiento y requisitos.

Por otro lado, hay empresas que dada su naturaleza o sector que cubren, requieren que su personal realice su jornada laboral fuera de sus instalaciones, viéndose más diseminadas y obligadas a llevar a cabo nuevas dinámicas de trabajo. Hemos podido apreciar que estas funcionalidades no están en la mayoría de las herramientas actuales.

Todos estos motivos hacen que se plantee LaborFlix, una plataforma web que pretende concentrar y mejorar los procesos de gestión de recursos humanos.

1.3. Objetivos

El objetivo que se persigue con este proyecto es realizar una plataforma web que cumpla con las siguientes funcionalidades:

- Elaboración, gestión y personalización de equipos de trabajos asociados a una ubicación física.
- Elaboración y personalización de turnos laborales para diferentes trabajadores.
- Realización de fichajes laborales geolocalizados en base a la ubicación previamente asignada.
- Posibilidad de visualización del horario laboral individual previamente asignado.
- Posibilidad de intercambio de días laborales incluidos en el horario con otro usuario de la plataforma.
- Obtención de informes de los fichajes realizados en la plataforma.

Por otro lado, debido al alcance actual del proyecto y problemas en los tiempos de elaboración de la funcionalidad indicada, hay una característica que no se ha podido realizar, aunque se encontraba en nuestros objetivos iniciales.

- Sistema de permisos laborales

1.4. Plataformas similares

Este proyecto se ha inspirado en otras plataformas existentes del ámbito del HRM (Human Resource Management) que reunían algunas de las funcionalidades planteadas, seleccionando las más esenciales de esta materia y reuniéndolas en una sola, a la par que adaptándolas para que la plataforma resultante pueda resultar más atractiva a empresas con trabajadores/equipos de campo.

- **Zoho People:** Zoho People se destaca como una plataforma de gestión de recursos humanos en la nube que brinda una amplia gama de funciones esenciales. Entre ellas se incluyen el registro de asistencia, la gestión de permisos y el seguimiento de horarios laborales.

Este software permite a los empleados llevar un registro preciso de sus horas de trabajo, así como solicitar permisos y administrar sus horarios de manera eficiente. Aunque cabe mencionar, que esta plataforma carece de un sistema de cambio de horario, presenta una curva de aprendizaje pronunciada y su enfoque en la funcionalidad puede resultar algo rígido y poco adaptable a ciertas necesidades específicas [2].



Figura 1.1: Logotipo Zoho People

- **ADP Workforce Now:** ADP Workforce Now es plataforma en la nube gestión de recursos humanos que ofrece características como registro de tiempo y asistencia, gestión de ausencias y planificación de turnos. Los empleados pueden registrar horas de trabajo, solicitar permisos y consultar su saldo de días libres. Los gerentes pueden supervisar asistencia, aprobar solicitudes de ausencia y programar turnos.

También se integra con otras aplicaciones empresariales, especialmente en la gestión de nóminas. Aunque su curva de aprendizaje es pronunciada y puede no ser ideal para organizaciones más pequeñas que necesitan funciones específicas de control horario y asignación de turnos [3].



Figura 1.2: Logotipo Workforce Now

Estas plataformas seleccionadas presentan una serie de ventajas y puntos fuertes que las aporta una cierta ventaja competitiva sobre el resto de las del sector, como pueden ser, la elaboración de horarios y turnos personalizados, registro de asistencia y revisión dichas asistencias por parte de los gerentes (rol superior al de los trabajadores). Sin embargo, también presentan ciertas debilidades, como curvas de aprendizaje muy pronunciadas y poca flexibilidad en la adaptación de los horarios.

LaborFlix se ha desarrollado intentando incorporar las fortalezas más notables de estas aplicaciones y evitando sus debilidades, para así poder ofrecer una funcionalidad distinguible.

Capítulo 2

2. Planificación

En este capítulo se desarrolla la metodología escogida para el desarrollo del proyecto, la planificación ideal, riesgos y planificación de estos.

2.1. Scrum

La metodología utilizada en este proyecto es Ágil y el marco de trabajo es Scrum, por lo que a continuación se procede a contextualizarla, definirla y explicar que caracteriza a este marco de trabajo.

2.1.1. Definición

Scrum es una metodología de trabajo iterativa e incremental para la gestión de proyectos, desplegado principalmente en el desarrollo ágil de software. La metodología scrum hace hincapié en el software funcional, la flexibilidad para cambiar junto con las realidades de negocio emergentes como páginas web, la comunicación y la colaboración [\[4\]](#).

El enfoque central de Scrum es la entrega iterativa e incremental de valor al cliente. En lugar de planificar y ejecutar todo el proyecto de una vez, Scrum divide el trabajo en incrementos más pequeños llamados "sprints". Cada sprint tiene una duración fija, generalmente de dos a cuatro semanas, y al final de cada sprint, se entrega un incremento de producto potencialmente utilizable, es decir, un producto mínimo viable.

La metodología Scrum se basa en principios fundamentales que incluyen la transparencia, la inspección y la adaptación. Estos principios proporcionan una estructura para que los equipos de trabajo colaboren y se adapten continuamente a medida que avanzan en el proyecto [\[5\]](#).

2.1.2. Características de Scrum

La característica principal de Scrum es que está formado por varios componentes característicos que trabajan juntos para facilitar la gestión y el desarrollo de proyectos:

2.1.2.1. Artefactos

Scrum utiliza varios artefactos para facilitar la gestión del trabajo y la transparencia [\[6\]](#):

- **Backlog del producto:** Es una lista priorizada de todos los elementos que se desean en el producto, incluyendo funcionalidades, requisitos y mejoras. El Product Owner es responsable de mantener y actualizar el backlog del producto.
- **Backlog del sprint:** Es una selección de elementos del backlog del producto que el equipo se compromete a completar durante el sprint actual. El backlog del sprint se crea en la planificación del sprint y se ajusta a medida que el equipo avanza en su trabajo.
- **Incremento del producto:** Es el resultado tangible del sprint, es decir, el trabajo completado y potencialmente entregable al finalizar cada iteración. El incremento debe ser de alta calidad y estar en condiciones de ser utilizado por el cliente.

2.1.2.2. Eventos

Scrum establece una serie de eventos para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum [7]. Estos eventos incluyen:

- **Planificación del sprint:** Es una reunión en la que el equipo selecciona los elementos del backlog del producto que se abordarán durante el próximo sprint y define cómo se realizará el trabajo.
- **Daily Scrum:** Es una reunión diaria de seguimiento de 15 minutos en la que el equipo sincroniza su trabajo, discute el progreso y los obstáculos, y planifica las próximas actividades.
- **Revisión del sprint:** Al final de cada sprint, se realiza una reunión en la que el equipo muestra el incremento de producto completado y recibe retroalimentación de los stakeholders.
- **Retrospectiva del sprint:** Es una reunión en la que el equipo reflexiona sobre el sprint pasado, identifica mejoras y define acciones para implementar en el próximo sprint.

2.1.2.3. Roles

Scrum define tres roles principales [8].

- **Product Owner:** responsable de representar las necesidades de los clientes y stakeholders, así como de gestionar el backlog del producto.
- **Scrum Master:** es el facilitador del proceso, se encarga de eliminar obstáculos, guiar al equipo y asegurarse de que se apliquen correctamente los principios y prácticas de Scrum.
- **Equipo de Desarrollo:** está compuesto por profesionales multidisciplinarios que son responsables de entregar el trabajo y alcanzar los objetivos del proyecto.

Estos componentes característicos de Scrum trabajan en conjunto para fomentar la colaboración, la transparencia y la adaptabilidad en el desarrollo de proyectos. Scrum se enfoca en la entrega de valor de manera iterativa y continua, lo que permite a los equipos responder de manera ágil a los cambios y obtener retroalimentación temprana del cliente, lo que a su vez conduce a un producto final más ajustado a las necesidades y expectativas del cliente.

2.1.3. Aplicación al proyecto

Dado que este proyecto es un Trabajo Fin de Grado donde las personas involucradas en él son el alumno y la tutora de dicho proyecto es imposible aplicar fielmente las características Scrum, por lo que para este caso concreto se ha hecho una readaptación de los diferentes elementos de la metodología, pero manteniendo su esencia y finalidad.

Los roles se reasignan dando lugar a que:

- el estudiante asume el papel del **Equipo de Desarrollo**,
- el papel del **Scrum Master** lo asume Margarita, la tutora de dicho TFG,
- y el rol de **Product Owner** se ha compartido por ambos, ya que han sido algunas ideas comunes las que se han aplicado al product backlog.

En cuanto a los eventos, se sustituirán las **Daily Scrum** por **Weekly Scrum**, es decir, las reuniones diarias serán sustituidas por reuniones semanales cada viernes, cuyo objetivo será revisar el avance del proyecto, correcciones y necesidades futuras.

2.2. Planificación Ideal

Se establecerá una duración de sprint de 2 semanas, coincidiendo el final de este con la entrega y revisión de los avances llevados a cabo en dicho sprint junto con la planificación del siguiente. A su vez, todos los viernes de cada semana se llevará a cabo un seguimiento semanal (Weekly Scrum, mencionado anteriormente) del desarrollo del proyecto hasta la fecha.

El trabajo de fin de grado cuenta con un desarrollo total comprendido en 300 horas, teniendo como fecha de finalización del proyecto el 19 de junio obtenemos un total de 17 semanas de trabajo, lo cual componen un total de 8 sprints, compuesto cada uno por aproximadamente 36 horas, obteniendo como resultado, 18 horas semanales.

A continuación, observamos cómo quedaría la planificación mencionada en los apartados anteriores desde el punto de vista mensual

Número de Sprint	Fecha Inicio	Fecha Fin
Sprint 1	17/02/2023	03/03/2023
Revisión semanal y planificación	24/02/2023	
Sprint 2	03/03/2023	17/03/2023
Revisión semanal y planificación	10/03/2023	
Sprint 3	17/03/2023	31/03/2023
Revisión semanal y planificación	24/03/2023	
Sprint 4	31/03/2023	14/04/2023
Revisión semanal y planificación	07/02/2023	
Sprint 5	14/04/2023	28/04/2023
Revisión semanal y planificación	21/04/2023	
Sprint 6	28/04/2023	12/05/2023
Revisión semanal y planificación	05/05/2023	
Sprint 7	12/05/2023	26/05/2023
Revisión semanal y planificación	19/05/2023	
Sprint 8	26/05/2023	09/06/2023
Revisión semanal y planificación	02/06/2023	
Semana Revisión	09/06/2023	19/06/2023
Revisiones y retoques finales		

Tabla 2.1: Planificación Inicial

En esta sección presentaremos el análisis del proyecto dividido en el análisis de riesgos, análisis de requisitos, identificación y presentación de casos de uso y modelo de dominio.

2.3. Análisis de Riesgos

En cualquier proyecto de cualquier índole pueden surgir siempre imprevistos, situaciones no contempladas que, por su naturaleza, pueden provocar tanto la ralentización o el frenado en el avance de dicho proyecto como el resultado del producto final obtenido, llevando consigo contratiempos tales como que los plazos y fechas previstas puedan verse comprometidas o como que el producto final no cumpla con ciertas características previamente establecidas.

2.3.1. Tipos de riesgos

Como cualquier otra métrica, los riesgos pueden ser definidos, es por esto por lo que dependiendo de su naturaleza distinguimos tres tipos bien marcados [\[9\]](#).

- **Riesgos conocidos:** aquellos riesgos que pueden ser identificados y analizados, para los cuales puede elaborarse un plan de mitigación y uno de respuesta. Por otro lado, a estos riesgos es necesario asignarles; una categoría para poder acotar su naturaleza, una probabilidad de que este riesgo se haga presente, el impacto que tendría este al aparecer y, por último, que consecuencias conllevarían para con el proyecto al producirse.
- **Riesgos predecibles:** son los riesgos que pueden extrapolarse por propia experiencia personal en proyectos anteriores.

- **Riesgos impredecibles:** como su nombre indica, este tipo de riesgo recoge a aquellos que son extremadamente difíciles de identificar con antelación, llevando consigo que si uno se materializase sus efectos serían difíciles de gestionar, puesto que no se habría podido elaborar un plan de respuesta previo.

2.3.2. Tipos de planes

Como se ha descrito con anterioridad, para los riesgos conocidos existen una serie de planes de actuación y prevención que es necesario conocer.

- **Plan de mitigación:** conjunto de estrategias definidas cuyo objetivo es reducir o eliminar la posibilidad de que un riesgo se materialice.
- **Plan de respuesta:** conjunto de procedimientos y acciones que ayudan a responder de manera efectiva ante la aparición de un riesgo previamente identificado y analizado. Su objetivo principal es reducir en la medida de lo posible el impacto generado por dicho riesgo.

2.3.3. Categorización de los riesgos, probabilidades e impactos

Por último, cabe desarrollar cómo se podrán clasificar las categorías, impactos y probabilidades que se le asignen a cada riesgo identificado [\[9\]\[10\]](#).

En primer lugar, definimos cuales son y en que se basa cada categoría:

- **Riesgos técnicos:** riesgos relacionados con la complejidad del software y las tecnologías a utilizar en el proyecto.
- **Riesgos de negocio:** estos afectan al contexto empresarial, de definición, financiación, mercado y requisitos de un proyecto.
- **Riesgos humanos:** aquellos riesgos relacionados con los miembros del equipo del proyecto y sus situaciones personales que impidan que estos realicen un trabajo óptimo.
- **Riesgos de gestión:** riesgos relacionados con la gestión, planificación, organización, seguimiento y control de un proyecto.
- **Riesgos de seguridad:** incluyen a aquellos relacionados con la seguridad del software y la documentación, englobando no solo a los riesgos de vulnerabilidades y ataques, sino también a los que contemplan pérdida sensible de información o código por fallos de sistema.

A continuación, listamos cuales son y en que se basa cada probabilidad:

Nivel	Probabilidad	Descripción
1	Raro	El riesgo puede darse solo en circunstancias excepcionales
2	Improbable	El riesgo puede ocurrir en circunstancias no tan excepcionales
3	Posible	El riesgo podría ocurrir en algún momento
4	Probable	El riesgo probablemente ocurrirá en alguna circunstancia
5	Casi seguro	Se espera que el riesgo tenga lugar en la mayoría de las circunstancias

Tabla 2.2: Probabilidades de los riesgos [11]

Y, por último, qué impactos existen y cuáles son sus significados:

Nivel	Impacto	Descripción
1	Insignificante	Si el hecho llegara a presentarse, tendría consecuencias o efectos mínimos
2	Menor	Si el hecho llegara a presentarse, tendría bajo impacto
3	Moderado	Si el riesgo llegara a presentarse, tendría un impacto medio
4	Grave	Si el hecho llegara a presentarse, tendría consecuencias considerables
5	Extremo	Si el hecho llegara a presentarse, tendría consecuencias desastrosas

Tabla 2.3: Impactos de los riesgos [11]

Resultante de estas dos tablas obtenemos una matriz de probabilidad e impacto que permite cuantificar cada riesgo detectado, esta matriz se ha elaborado siguiendo la guía PMBOK.

Probabilidad	Impacto				
	Insignificante (1)	Menor (2)	Moderado (3)	Grave (4)	Extremo (5)
Raro (1)					
Improbable (2)					
Posible (3)					
Probable (4)					
Casi seguro (5)					

Tabla 2.4: Probabilidades x Impactos PMBOK [11]

Con el resultado de esta matriz se propone una escala numérica que permite clasificar el riesgo según su nivel.

Nivel de riesgo	Probabilidad X Impacto
Muy alto	>80
Alto	51-80
Medio	31-50
Bajo	11-30
Muy Bajo	<10

Tabla 2.5: Niveles de riesgos [11]

2.3.4. Riesgos

Una vez en contexto del análisis de riesgos, debemos identificar y analizar aquellos relacionados con este proyecto, listándolos de manera ascendente en su probabilidad e impacto.

Riesgo R001		No disponibilidad de equipo hardware de trabajo			
Descripción		El desarrollador puede sufrir una situación que comprometa la integridad tanto de su equipo de trabajo como de la información que este contenía			
Categoría	Riesgo de seguridad	Probabilidad	Raro	Impacto	Extremo
Plan de mitigación		<ul style="list-style-type: none"> • Para la documentación, utilización de plataformas en la nube que permitan a su vez la compartición de dichos documentos (SharePoint) • Para el código del proyecto, utilización de plataformas en la nube de control de versiones (Git) • Para el equipo físico, realizar un diagnóstico inicial que asegure su correcto funcionamiento 			
Plan de respuesta		<ul style="list-style-type: none"> • Tanto para la documentación como para el código habría que recuperar la copia de seguridad más reciente posible • Disponer de otro equipo que ofrezca las funcionalidades y herramientas necesarias para el correcto desarrollo del proyecto 			
Consecuencias		Los problemas en la información relativa al proyecto conllevarían la pérdida de la información no guardada, con la consecuente reelaboración del progreso perdido, la cual dependería de la cantidad de información perdida			

Tabla 2.6: R001 – No disponibilidad del equipo hardware

Riesgo R002		Falta de conocimientos o experiencia con las herramientas utilizadas			
Descripción		El desarrollador puede no conocer en diferentes grados la utilización o funcionamiento de las diferentes herramientas utilizadas, como los lenguajes de programación, <i>frameworks</i> , bases de datos etc.			
Categoría	Riesgo técnico	Probabilidad	Improbable	Impacto	Moderado
Plan de mitigación		<ul style="list-style-type: none"> Realizar tanto un repaso inicial de las herramientas a utilizar como una continua formación de estas 			
Plan de respuesta		<ul style="list-style-type: none"> Consulta de la información de una manera más activa y urgente hasta resolver las dudas aparecidas Consulta de la información no conocida con un profesional con experiencia en la herramienta 			
Consecuencias		Un desconocimiento en alguna de las herramientas podría ralentizar la productividad del desarrollador en mayor o menor grado			

Tabla 2.7: R002 – Falta de conocimiento o experiencia con las herramientas utilizadas

Riesgo R003		Falta de disponibilidad de miembro/s del proyecto			
Descripción		Tanto el desarrollador como la tutora del proyecto pueden sufrir circunstancias personales o laborales que les impida su implicación activa en el proyecto			
Categoría	Riesgo humano	Probabilidad	Improbable	Impacto	Grave
Plan de mitigación		<ul style="list-style-type: none"> Realizar reuniones semanales para que la actualización del trabajo realizado hasta el momento sea lo más reciente posible 			
Plan de respuesta		<ul style="list-style-type: none"> Utilización de herramientas comunicativas online para el seguimiento y/o dudas puntuales con una frecuencia que el afectado considere aceptable en su circunstancia 			
Consecuencias		La ausencia prolongada de un miembro del proyecto podría conllevar la reestimación en el alcance y planificación del proyecto			

Tabla 2.8: R003 – Falta de disponibilidad de miembro/s del proyecto

Riesgo R004		Cambios en los requisitos			
Descripción		Los requisitos del proyecto pueden cambiar a medida que se desarrolla, llegando a producirse cuando dichos cambios comprometan las fechas y estimaciones del proyecto.			
Categoría	Riesgo de negocio	Probabilidad	Posible	Impacto	Grave
Plan de mitigación		<ul style="list-style-type: none"> Definir la funcionalidad del proyecto de la forma más clara y concisa posible 			
Plan de respuesta		<ul style="list-style-type: none"> Decidir, si por el alcance y la fecha de aparición de la modificación o surgimiento de dicho requisito, hay que realizar un cambio de alcance o por el contrario descartarlo 			
Consecuencias		En caso de que el requisito se acepte habría que replantear la estimación en el alcance y planificación del proyecto			

Tabla 2.9: R004 – Cambios en los requisitos

Riesgo R005		Interpretación errónea de los requisitos			
Descripción		Una incorrecta interpretación en los requisitos por confusión o mala redacción de estos, repercutiendo en su elaboración u objetivo final			
Categoría	Riesgo de negocio	Probabilidad	Posible	Impacto	Grave
Plan de mitigación		<ul style="list-style-type: none"> Confirmación bidireccional con la tutora del proyecto de la elaboración y objetivos a alcanzar con cada funcionalidad Revisión activa y reelaboración (si es necesaria) de los requisitos al añadir nueva funcionalidad 			
Plan de respuesta		<ul style="list-style-type: none"> Readaptación o bien de la estimación del requisito o del propio requisito en sí para adaptarlo a la estimación original 			
Consecuencias		Una interpretación errónea de los requisitos puede conllevar la necesidad de reducir un requisito funcionalmente básico para el aplicativo o bien, un aumento en el tiempo de elaboración del proyecto			

Tabla 2.10: R005 – Interpretación errónea de los requisitos

Riesgo R006		Error en las estimaciones			
Descripción		Una incorrecta estimación de tiempo de una funcionalidad del proyecto			
Categoría	Riesgo de gestión	Probabilidad	Probable	Impacto	Moderado
Plan de mitigación		<ul style="list-style-type: none"> Asignar estimaciones muy ligeramente sobreestimadas para contar con un margen de tiempo y maniobra 			
Plan de respuesta		<ul style="list-style-type: none"> Realizar la funcionalidad restante en el siguiente sprint junto con otra funcionalidad de baja estimación para poder completar ambas 			
Consecuencias		El errar en las estimaciones podría provocar retrasos en cadena de la realización de las diferentes funcionalidades estimadas			

Tabla 2.11: R006 – Error en las estimaciones

Riesgo R007		Interfaz gráfica de usuario poco usable			
Descripción		El resultado final del proyecto no cumple con unos deseos estéticos o funcionales mínimos			
Categoría	Riesgo de gestión	Probabilidad	Probable	Impacto	Grave
Plan de mitigación		<ul style="list-style-type: none"> Crear prototipos de la IU (interfaz de usuario) antes de desarrollar el componente deseado para aproximar lo más posible el aspecto final 			
Plan de respuesta		<ul style="list-style-type: none"> Realizar todos los posibles cambios visuales que no afecten significativamente a la funcionalidad 			
Consecuencias		Si la IU final es incómoda o poco usable para el usuario el atractivo y la agilidad de la funcionalidad puede verse comprometida			

Tabla 2.12: R007 – Interfaz gráfica de usuario poco usable

2.4. Presupuesto

2.4.1. Presupuesto Simulado

Para elaborar el presupuesto del proyecto software se deben tener en cuenta varios factores que influyen en el presupuesto final.

En primer lugar, el sueldo del desarrollador, a quien tras consultar diversas fuentes y estadísticas le correspondería un sueldo medio de 25.000€ anuales estimados en 12.82€ la hora de trabajo [12]. Dados todos estos datos concluimos que para las 300 horas estimadas para este proyecto el sueldo correspondiente al desarrollador de este rondaría los 3846€ netos, ya que se ha descontado previamente el 33% correspondiente dirigido a la Seguridad Social (1893,55€, calculado del porcentaje descontado, con un total bruto de 5739,55€).

A continuación, hay que contemplar el equipo de trabajo a utilizar, en este caso el desarrollo se ha realizado en un ordenador de sobremesa customizado valorado en unos 1100€ con una vida útil de 5 años, por lo que el precio aproximado es de 18,33€/mes y dado que el proyecto cuenta con una duración de 4 meses obtenemos un coste amortizado de 73,32€. Cabe mencionar también que se han de utilizar dispositivos de distintas resoluciones para validar que el software se adapta a distintas resoluciones. En este contexto se ha utilizado un dispositivo móvil Xiaomi Redmi Note 11 (247,92€) y un iPad Air 2 (399,98€), contando que ambos dispositivos cuentan con una vida útil de 4 años los costes mensuales obtenidos serían de 5,16€/mes y 8,33€/mes, con el consecuente precio dedicado a los 4 meses del proyecto, 20,64€ y 33,32€, respectivamente.

Por último, las licencias del software derivado de dicho desarrollo, entre las que se encuentran:

- **Microsoft 365:** herramienta de ofimática que entre los planes que ofrece escogeremos el plan básico por 5,60€/mes, obteniendo así 22,4€ en la duración del proyecto.
- **Astah Professional:** herramienta de modelado UML con un coste de licencia total de 146€, siendo así de 12,17€/mes y un total de 48,67€ en los 4 meses de trabajo.

Si sumamos todos los precios desarrollados anteriormente obtenemos un precio total estimado para el proyecto de 5937,9€, teniendo en cuenta que este valor está obtenido de una estimación y que debemos contemplar posibles gastos imprevistos durante el desarrollo, aumentaremos un 30% el coste total, con un resultante de **7710,27€**.

Elemento	Coste	Tiempo	Coste Total (€)
Sueldo desarrollador	12,82 €/h	300 horas	3846
Seguridad Social	6,31 €/h	300 horas	1893,55
Equipo de trabajo (Ordenador)	18,33/mes	4 meses	73,32
Equipo de trabajo (iPad)	5,16 €/mes	4 meses	20,64
Equipo de trabajo (Móvil)	8,33 €/mes	4 meses	33,32
Licencia Microsoft 365	5,6 €/mes	4 meses	22,4
Licencia Astah Professional	12,17 €/mes	4 meses	48,67
Total			5937,9
Total + % gastos imprevistos			7710,27

Tabla 2.13: Resumen de presupuesto simulado

2.4.2. Presupuesto real

Dado que nos encontramos ante un Trabajo de Fin de Grado, los costes asociados con el sueldo del empleado y pago a Seguridad Social no tienen cabida puesto que es el propio alumno quien realiza el desarrollo.

En lo relacionado al equipo de trabajo, sí que aplica mantener los precios explicados en el apartado anterior, ya que están basados en precios reales del equipo del alumno (73,32€, 20,64€ y 33,32€).

Y, por último, los costes de las licencias derivadas del software mencionado (Microsoft365 y Astah Profesional) tampoco se aplican pues es la propia Universidad de Valladolid quien suministra licencias de estudiantes gratuitas a todos sus alumnos.

Si tenemos en cuenta todos estos factores, el desarrollo total del proyecto resulta en **124,28€**.

Elemento	Coste	Tiempo	Coste Total (€)
Equipo de trabajo (Ordenador)	18,33/mes	4 meses	73,32
Equipo de trabajo (iPad)	5,16 €/mes	4 meses	20,64
Equipo de trabajo (Móvil)	8,33 €/mes	4 meses	33,32
Total			124,28

Tabla 2.14: Resumen de presupuesto real

Capítulo 3

3. Análisis

En este capítulo abordaremos la fase de análisis del proyecto, elaborando y redactando el análisis de requisitos, casos de uso, modelo de dominio y diagramas de secuencia en análisis de este.

3.1. Análisis de Requisitos

En primer lugar, presentaremos las historias de usuario y los requisitos presentes en el proyecto (funcionales, no funcionales, de información y reglas de negocio)

3.1.1. Historias de usuario

Como hemos mencionado anteriormente, seguimos la metodología Scrum, por lo que se nos presentan historias de usuarios, es decir, peticiones por parte del cliente, que siguen el formato [\[13\]](#):

“Como [perfil], quiero [objetivo] para [interés]”

Se aprecia en la estructura que para poder realizar de forma correcta dichas historias primero debemos identificar los parámetros necesarios

- Perfil: usuario o grupo de usuarios que comparten ciertas características que los hacen agrupables, para los que se plantea la historia
- Objetivo: intencionalidad final del perfil, y la que promueve que este use el aplicativo
- Interés: motivo que aporta valor y utilidad al objetivo del producto

Las historias de usuario solicitadas para este aplicativo son las siguientes

1. Como responsable quiero visualizar información resumida de los datos registrados en la plataforma, gestionar a mis equipos de trabajo, ubicaciones, trabajadores y los turnos relacionados con estos para así poder optimizar mis recursos empresariales y organizar la vida laboral en la entidad.
 - 1.1. Como responsable quiero poder asignar trabajadores a la plataforma para poder poblarla y hacerla funcional.
 - 1.2. Como responsable quiero poder visualizar informes para poder obtener información relevante de los usuarios registrados.
 - 1.3. Como responsable quiero poder dar de alta turnos laborales en la plataforma para poder administrar las jornadas de los trabajadores.

- 1.4. Como responsable quiero poder dar de alta equipos de trabajo en la plataforma para poder organizar a mis trabajadores.
 - 1.5. Como responsable quiero poder asignar turnos a trabajadores dentro de los turnos registrados en la plataforma para que los trabajadores conozcan su jornada laboral.
 - 1.6. Como responsable quiero dar de alta ubicaciones reales asignadas a equipos, para acotar su zona de fichaje.
2. Como trabajador y como responsable quiero poder conocer mi jornada laboral actualizada, realizar mis fichajes de trabajo y poder intercambiar días laborales con otro usuario, para así poder administrar mi información laboral de una forma más accesible y eficaz.
- 2.1. Como usuario quiero poder conocer mi jornada laboral para estar al tanto de los cambios de esta y poder anticiparme y organizarme en mi vida privada.
 - 2.2. Como usuario quiero poder realizar fichajes laborales para que queden registradas mis horas de trabajo.
 - 2.3. Como usuario quiero poder conocer y editar mi propio perfil para así controlar mis datos personales.
 - 2.4. Como usuario quiero poder solicitar cambios puntuales en mi turno laboral para amoldarlo a necesidades específicas personales.

3.1.2. Requisitos Funcionales

Del análisis de las historias de usuario mencionadas con antelación obtenemos los siguientes requisitos funcionales, aquellos que describen las funcionalidades que debe aportar el sistema.

Los requisitos RF007, RF008 y RF009 se han eliminado por parte del Product Owner debido a problemas en los tiempos de desarrollo.

Código	Nombre	Descripción
RF001	Registrar usuario	El sistema debe permitir al responsable dar de alta otros responsables y trabajadores en la plataforma.
RF002	Editar usuario	El sistema debe permitir a un usuario modificar su información personal.
RF003	Inhabilitar usuario	El sistema debe permitir a un responsable inhabilitar a otro usuario de la plataforma, impidiendo su acceso a esta.
RF004	Registrar turno	El sistema debe permitir a un responsable registrar un nuevo horario laboral.
RF005	Editar turno	El sistema debe permitir a un responsable editar un horario registrado en la plataforma.
RF006	Eliminar turno	El sistema debe permitir a un responsable borrar un horario registrado en la plataforma.
RF007	Registrar Disposición de Días	El sistema debe permitir al responsable añadir la cantidad y la naturaleza de días libres de los trabajadores.
RF008	Editar Disposición de Días	El sistema debe permitir al responsable editar las disposiciones de días registradas en la plataforma.
RF009	Eliminar Disposición de Días	El sistema debe permitir al responsable eliminar las disposiciones de días registradas en la plataforma.
RF010	Registrar equipos de trabajo	El sistema debe permitir al responsable añadir equipos de trabajos.
RF011	Editar equipos de trabajos	El sistema debe permitir al responsable editar los equipos de trabajos
RF012	Eliminar equipos de trabajos	El sistema debe permitir al responsable eliminar los equipos de trabajos
RF013	Asignar usuarios	El sistema debe permitir a un responsable asignar usuarios a un equipo.
RF014	Asignar turnos	El sistema debe permitir a un responsable asignar turnos a los usuarios.
RF015	Obtener turnos	El sistema debe permitir a cada usuario conocer sus turnos laborales actuales.
RF016	Realizar fichaje	El sistema debe permitir a un usuario realizar fichajes horarios en su jornada laboral.
RF017	Registrar cambio de jornada	El sistema debe permitir a un usuario registrar una solicitud de cambio de jornada laboral
RF018	Aceptar cambio de jornada	El sistema debe permitir a otro usuario de la plataforma visualizar y aceptar la petición
RF019	Obtención de informes	El sistema debe permitir a un responsable obtener estadísticas e informes de los datos laborales de la entidad.

Tabla 3.1: Requisitos Funcionales

3.1.3. Requisitos No Funcionales

Los requisitos no funcionales son aquellos requisitos que describen una característica o propiedad que no está relacionada directamente con la funcionalidad del sistema, sino que se refieren a otros aspectos vinculados a las características del funcionamiento.

Código	Nombre	Descripción
RNF001	Plataforma de la aplicación	El sistema se podrá ejecutar en cualquier dispositivo con conexión a internet mediante un navegador web.
RNF002	Seguridad de credenciales	El sistema almacenará las contraseñas de los usuarios encriptadas en la base de datos.
RNF003	Usabilidad	La interfaz de usuario debe ser intuitiva y fácil de aprender para mejorar la experiencia del usuario
RNF004	Actualización de la información	El sistema mantendrá la información actualizada en todo momento
RNF005	Tratamiento de errores	El sistema contará con validaciones tanto de datos como de flujos, contemplando cualquier posible error y proporcionando una respuesta contemplada a este.

Tabla 3.2: Requisitos No Funcionales

3.1.4. Requisitos de Información

Para continuar con el análisis de requisitos nos encontramos con los requisitos de información, los cuales describen la información que el sistema debe capturar y almacenar.

Código	Descripción
RI001	El sistema guardará información relacionada con los usuarios: <ul style="list-style-type: none"> • Nombre • Apellidos • DNI • Correo electrónico • Contraseña • Rol
RI002	El sistema guardará información relacionada con los turnos laborales: <ul style="list-style-type: none"> • Nombre • Descripción • Días laborales • Usuarios asignados
RI003	El sistema guardará información relacionada con los equipos de trabajo: <ul style="list-style-type: none"> • Nombre • Descripción • Usuarios asignados
RI004	El sistema guardará información relacionada con los fichajes laborales: <ul style="list-style-type: none"> • Fecha de inicio • Fecha de fin • Coordenadas de inicio • Coordenadas de fin • Fichaje realizado dentro/fuera de ubicación • Usuario
RI005	El sistema guardará información relacionada con los cambios de jornadas: <ul style="list-style-type: none"> • Estado • Fecha de publicación • Fecha de contestación • Usuario emisor • Días laborales del usuario emisor • Usuario aceptante • Días laborales del usuario aceptante

Tabla 3.3: Requisitos de Información

3.1.5. Reglas de Negocio

Por último, cabe describir las reglas de negocio de la plataforma, aquellas directrices que definen cómo debe funcionar y comportarse el software desde el punto de vista de la lógica de negocio o funcionalidad específica.

- RN01: Los usuarios que hagan uso de la plataforma estarán ya registrados en el sistema a consecuencia de una exportación de datos que la propia empresa tendrá previamente al uso de esta.
- RN02: Se permite una importación a mayores de usuarios directamente en la plataforma mediante una plantilla Excel por parte de un responsable.

- RN03: Los equipos de trabajo solo pueden tener asignada una ubicación.
- RN04: Los usuarios pueden tener más de un turno asignado.
- RN05: Cuando un usuario realiza un fichaje, se reflejará si lo ha hecho dentro del radio de la ubicación que tiene asignada su equipo. Si por el contrario no tiene asignado un equipo o ubicación o la realiza fuera de esta, también queda reflejado.
- RN06: Si se completa una solicitud de cambio de turno, los días intercambiados entre usuarios quedan permanentemente asignados a estos.

3.2. Casos de uso

A continuación, veremos los diferentes casos de uso que se plantean en la plataforma junto con su especificación

3.2.1. Diagrama de casos de uso

El diagrama de casos de uso resultante del análisis del proyecto es el siguiente:

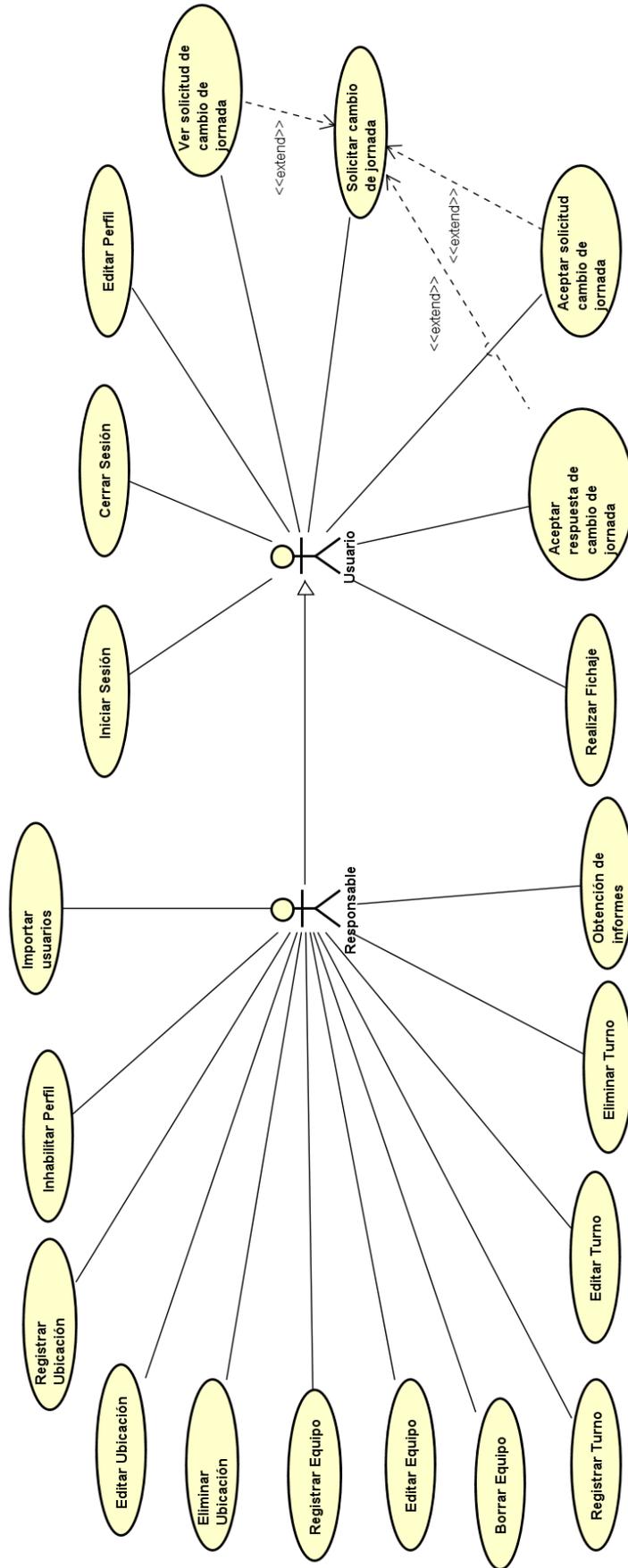


Figura 3.1: Diagrama de casos de uso

3.2.2. Especificación de casos de uso

Una vez visto los diferentes casos de uso que se presentan en el sistema detallaremos de forma específica cada uno de ellos.

CU001	Iniciar sesión	
Actor	Usuario	
Descripción	El usuario quiere iniciar sesión en el sistema	
Precondición	El usuario está registrado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción “Iniciar sesión”
	2	El sistema muestra un formulario con email y contraseña
	3	El usuario rellena el formulario y pulsa el botón de iniciar sesión
	4	El sistema comprueba la validez del formulario
	5	El caso de uso finaliza
Postcondición	El usuario queda autenticado en el sistema y es redirigido a la pestaña de Inicio	
Excepciones	Paso	Acción
	4'	Si algún campo es incorrecto o está vacío se indica el tipo de error y se vuelve al paso 3

Tabla 3.4: CU001 Iniciar Sesión

CU002	Cerrar sesión	
Actor	Usuario	
Descripción	El usuario quiere cerrar su sesión activa	
Precondición	El usuario está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción “Cerrar sesión”
	2	El caso de uso finaliza
Postcondición	El usuario sale de su sesión y es redirigido a la pestaña de Iniciar sesión	
Excepciones	Paso	Acción

Tabla 3.5: CU002 Cerrar Sesión

CU003	Editar perfil	
Actor	Usuario	
Descripción	El usuario quiere editar la información de su perfil	
Precondición	El usuario está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona su nombre y apellido en la cabecera
	2	El sistema muestra un formulario con la información del usuario junto con la opción de Editar Perfil
	3	El usuario edita en el formulario la información deseada y pulsa el botón editar perfil
	4	El sistema comprueba la validez del formulario
	5	El caso de uso finaliza
Postcondición	El sistema actualiza la información del usuario y este es redirigido a la pestaña de Perfil	
Excepciones	Paso	Acción
	4'	Si algún campo es incorrecto o está vacío se indica el tipo de error y se vuelve al paso 3
	3'	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.6: CU003 Editar perfil

CU004	Inhabilitar Perfil	
Actor	Responsable	
Descripción	El responsable quiere inhabilitar un usuario del sistema	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la pestaña usuarios
	2	El sistema muestra la información de los usuarios registrados en el sistema
	3	El usuario selecciona encima del usuario que desea inhabilitar
	4	El caso de uso finaliza
Postcondición	El sistema inhabilita al usuario y este es redirigido a la pestaña de Iniciar sesión	
Excepciones	Paso	Acción
	2'	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.7: CU004 Inhabilitar perfil

CU005	Importar Usuarios	
Actor	Responsable	
Descripción	El responsable quiere importar usuarios en la plataforma	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la pestaña usuarios
	2	El sistema muestra la información de los usuarios registrados en el sistema junto con el botón importar usuarios
	3	El responsable selecciona la opción importar usuarios
	4	El sistema muestra un descargable de la plantilla Excel a rellenar y una sección donde añadir el archivo Excel junto con un botón importar usuarios
	5	El responsable rellena la plantilla, la añade en la sección y selecciona la opción importar usuarios
	6	El sistema comprueba que toda la información en el Excel es válida y añade los usuarios al sistema
	7	El caso de uso finaliza
Postcondición	El sistema registra a los nuevos usuarios en el sistema	
Excepciones	Paso	Acción
	5'	Si el responsable retrocede en su acción el caso de uso queda sin efecto
	6'	Si la información que ha introducido el responsable en el archivo Excel es incorrecta se informa y se vuelve al paso 5

Tabla 3.8: CU005 Importar Usuarios

CU006	Solicitar cambio jornada	
Actor	Usuario	
Descripción	El usuario quiere solicitar un cambio de jornada	
Precondición	El usuario está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña de Horario
	2	El sistema muestra la pestaña del horario del usuario y la de cambios de jornada
	3	El usuario selecciona la opción de cambios de jornada
	4	El sistema muestra las solicitudes y los cambios propios del usuario junto con la opción de Nuevo Cambio
	5	El usuario selecciona la opción de nuevo cambio
	6	El sistema muestra los días laborales de los que dispone el usuario
	7	El usuario selecciona los días que desea y selecciona el botón registrar cambio
	8	El caso de uso finaliza
Postcondición	El sistema registra la solicitud de cambio de jornada y el resto de los usuarios la podrán visualizar como una solicitud de cambio de jornada. El usuario es redirigido a la pestaña de cambios de jornada	
Excepciones	Paso	Acción
	6'	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.9: CU006 Solicitar cambio jornada

CU007	Aceptar solicitud de cambio de jornada	
Actor	Usuario	
Descripción	El usuario quiere aceptar una nueva solicitud de cambio de jornada	
Precondición	El usuario está autenticado y otro usuario ha solicitado un cambio de jornada	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña de Horario
	2	El sistema muestra la pestaña del horario del usuario y la de cambios de jornada
	3	El usuario selecciona la opción de cambios de jornada
	4	El sistema muestra las solicitudes y los cambios propios del usuario junto con la opción de Nuevo Cambio
	5	El usuario selecciona la opción de Ver en la solicitud deseada
	6	El sistema muestra los días laborales disponibles del usuario junto con los solicitados en la petición
	7	El usuario selecciona la misma cantidad de días que tiene la solicitud y selecciona el botón aceptar solicitud
	8	El sistema registra la respuesta a la solicitud
9	El caso de uso finaliza	
Postcondición	El sistema registra que el usuario ha aceptado una solicitud de cambio de jornada a expensas de que esta sea validada. El usuario es redirigido a la pestaña de cambios laborales	
Excepciones	Paso	Acción
	7'	Si el usuario no selecciona la misma cantidad de días el sistema informa y se vuelve al paso 6
	7''	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.10: CU007 Aceptar solicitud de cambio de jornada

CU008	Ver solicitud de cambio de jornada	
Actor	Usuario	
Descripción	El usuario quiere ver una de sus solicitudes de cambio de jornada	
Precondición	El usuario está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña de Horario
	2	El sistema muestra la pestaña del horario del usuario y la de cambios de jornada
	3	El usuario selecciona la opción de cambios de jornada
	4	El sistema muestra las solicitudes y los cambios propios del usuario junto con la opción de Nuevo Cambio
	5	El usuario selecciona la solicitud que desea
	6	El sistema muestra la información de la solicitud
7	El caso de uso finaliza	
Postcondición		
Excepciones		

Tabla 3.11: CU008 Ver solicitud de cambio de jornada

CU009	Aceptar respuesta de cambio de jornada	
Actor	Usuario	
Descripción	El usuario quiere aceptar una respuesta a su solicitud de cambio de jornada	
Precondición	El usuario está autenticado y ha registrado una solicitud de cambio de jornada que ha sido aceptada	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña de Horario
	2	El sistema muestra la pestaña del horario del usuario y la de cambios de jornada
	3	El usuario selecciona la opción de cambios de jornada
	4	El sistema muestra las solicitudes y los cambios propios del usuario junto con la opción de Nuevo Cambio
	5	El usuario selecciona la opción de Ver en su cambio deseado
	6	El sistema muestra los días laborales que el usuario aceptante ha ofrecido para el intercambio
	7.a	El usuario acepta el cambio de jornada
	7.b	El usuario rechaza el cambio de jornada
	8	El sistema registra la respuesta
9	El caso de uso finaliza	
Postcondición	El sistema registra que el usuario ha aceptado una solicitud de cambio de jornada. El usuario es redirigido a la pestaña de su horario laboral	
Excepciones	Paso	Acción
	7'	Si el usuario rechaza la petición esta pasa a estado pendiente y puede ser aceptada por otro usuario
	7''	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.12: CU009 Aceptar respuesta de cambio de jornada

CU010	Realizar Fichaje	
Actor	Usuario	
Descripción	El usuario quiere realizar un fichaje	
Precondición	El usuario está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la pestaña de Fichaje
	2	El sistema muestra el mapa con la ubicación actual del usuario, y la de la ubicación que tiene asignada
	3.a	El usuario selecciona la opción de iniciar fichaje al no tener ningún fichaje abierto
	3.b	El usuario selecciona la opción de finalizar fichaje al tener un fichaje abierto
	4	El sistema registra el fichaje
5	El caso de uso finaliza	
Postcondición	El sistema registra el fichaje en el sistema	
Excepciones	Paso	Acción
	4'	El sistema registra el fichaje como realizado dentro/fuera de la ubicación dependiendo de la del usuario
	3'	Si el usuario retrocede en su acción el caso de uso queda sin efecto

Tabla 3.13: CU010 Realizar fichaje

CU011	Registrar Equipo	
Actor	Responsable	
Descripción	El responsable quiere registrar un equipo de trabajo	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la pestaña de Equipos
	2	El sistema muestra todos los equipos registrados junto con la opción de editar, eliminar en cada equipo y añadir un nuevo equipo
	3	El responsable selecciona la opción de Añadir equipo
	4	El sistema muestra un formulario del nuevo equipo junto con la asignación inicial de usuarios deseados
	5	El responsable completa el formulario, la asignación y selecciona el botón crear equipo
	6	El sistema valida el formulario y la asignación inicial
	7	El caso de uso finaliza
Postcondición	El sistema registra el nuevo equipo de trabajado en el sistema junto con su asignación a los usuarios. El responsable es redirigido a la pestaña de Equipos de trabajo	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
	6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5

Tabla 3.14: CU011 Registrar Equipo

CU012	Editar Equipo	
Actor	Responsable	
Descripción	El responsable quiere editar un equipo de trabajo	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la pestaña de Equipos
	2	El sistema muestra todos los equipos registrados junto con la opción de editar, eliminar en cada equipo y añadir un nuevo equipo
	3	El responsable selecciona la opción de Editar equipo
	4	El sistema muestra un formulario con la información del equipo junto con la asignación actual de usuarios
	5	El responsable edita el formulario, la asignación y selecciona el botón editar equipo
	6	El sistema valida el formulario y la asignación
7	El caso de uso finaliza	
Postcondición	El sistema actualiza la información del equipo de trabajado en el sistema junto con su asignación a los usuarios. El responsable es redirigido a la pestaña de Equipos de trabajo	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5''	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
	6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5

Tabla 3.15: CU012 Editar Equipo

CU013	Eliminar Equipo	
Actor	Responsable	
Descripción	El responsable quiere eliminar un equipo de trabajo	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la pestaña de Equipos
	2	El sistema muestra todos los equipos registrados junto con la opción de editar, eliminar en cada equipo y añadir un nuevo equipo
	3	El responsable selecciona la opción de Eliminar equipo
	4	El sistema muestra una ventana con una confirmación
	5	El responsable selecciona el botón eliminar equipo
	6	El sistema elimina el equipo de la plataforma
7	El caso de uso finaliza	
Postcondición	El sistema elimina la información del equipo junto con sus asignaciones a usuarios. El responsable es redirigido a la pestaña de Equipos de trabajo	
Excepciones	Paso	Acción
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto

Tabla 3.16: CU013 Eliminar Equipo

CU014	Registrar Turno	
Actor	Responsable	
Descripción	El responsable quiere registrar un nuevo turno	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Turnos
	2	El sistema muestra todos los turnos junto con la opción de editar, eliminar y añadir nuevo turno
	3	El responsable selecciona la opción de Añadir nuevo turno
	4	El sistema muestra un formulario del nuevo turno junto con la asignación inicial de usuarios deseados
	5	El responsable completa el formulario, la asignación y selecciona el botón crear turno
	6	El sistema valida el formulario y la asignación inicial
	7	El caso de uso finaliza
Postcondición	El sistema registra el nuevo turno en el sistema junto con su asignación a los usuarios. El responsable es redirigido a la pestaña de Turnos	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
	6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5

Tabla 3.17: CU014 Registrar Turno

CU015	Editar Turno	
Actor	Responsable	
Descripción	El responsable quiere editar un turno	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Turnos
	2	El sistema muestra todos turnos registrados junto con la opción de editar, eliminar en cada turno y añadir un nuevo turno
	3	El responsable selecciona la opción de Editar turno
	4	El sistema muestra un formulario con la información del turno junto con la asignación actual de usuarios
	5	El responsable edita el formulario, la asignación y selecciona el botón editar turno
	6	El sistema valida el formulario y la asignación
	7	El caso de uso finaliza
Postcondición	El sistema actualiza la información del turno en el sistema junto con su asignación a los usuarios. El responsable es redirigido a la pestaña de Turnos	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
	6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5

Tabla 3.18: CU015 Editar Turno

CU016	Eliminar Turno	
Actor	Responsable	
Descripción	El responsable quiere eliminar un turno	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Turno
	2	El sistema muestra todos los turnos registrados junto con la opción de editar, eliminar en cada equipo y añadir un nuevo turno
	3	El responsable selecciona la opción de Eliminar turno
	4	El sistema muestra una ventana de confirmación
	5	El responsable selecciona el botón eliminar turno
	6	El sistema elimina el turno de la plataforma
7	El caso de uso finaliza	
Postcondición	El sistema elimina la información del turno junto con sus asignaciones a usuarios. El responsable es redirigido a la pestaña de Turnos	
Excepciones	Paso	Acción
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto

Tabla 3.19: CU016 Eliminar Turno

CU017	Registrar Ubicación	
Actor	Responsable	
Descripción	El responsable quiere registrar una nueva ubicación	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Ubicaciones
	2	El sistema muestra todas las ubicaciones junto con la opción de editar, eliminar y añadir nueva ubicación
	3	El responsable selecciona la opción de Añadir nueva ubicación
	4	El sistema muestra un formulario de la nueva ubicación junto con la asignación inicial de equipos deseados
	5	El responsable completa el formulario, la asignación y selecciona el botón crear ubicación
	6	El sistema valida el formulario y la asignación inicial
7	El caso de uso finaliza	
Postcondición	El sistema registra la nueva ubicación en el sistema junto con su asignación a los equipos. El responsable es redirigido a la pestaña de Ubicaciones	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5	

Tabla 3.20: CU017 Registrar Ubicación

CU018	Editar Ubicación	
Actor	Responsable	
Descripción	El responsable quiere editar una ubicación	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Ubicaciones
	2	El sistema muestra todas las ubicaciones junto con la opción de editar, eliminar y añadir nueva ubicación.
	3	El responsable selecciona la opción de Editar ubicación
	4	El sistema muestra un formulario con la información de la ubicación junto con la asignación actual de equipos
	5	El responsable edita el formulario, la asignación y selecciona el botón editar ubicación
	6	El sistema valida el formulario y la asignación
7	El caso de uso finaliza	
Postcondición	El sistema actualiza la información de la ubicación en el sistema junto con su asignación a los equipos. El responsable es redirigido a la pestaña de Ubicaciones	
Excepciones	Paso	Acción
	6'	Si hay algún campo incorrecto o vacío se informa del tipo de error al responsable y se vuelve al paso 5
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto
	6''	Si la asignación es errónea se informa del tipo de error al responsable y se vuelve al paso 5

Tabla 3.21: CU018 Editar Ubicación

CU019	Eliminar Ubicación	
Actor	Responsable	
Descripción	El responsable quiere eliminar una ubicación	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Ubicaciones
	2	El sistema muestra todas las ubicaciones junto con la opción de editar, eliminar y añadir nueva ubicación.
	3	El responsable selecciona la opción de Eliminar ubicación
	4	El sistema muestra una ventana de confirmación
	5	El responsable selecciona el botón eliminar ubicación
	6	El sistema elimina la ubicación de la plataforma
7	El caso de uso finaliza	
Postcondición	El sistema elimina la información de la ubicación junto con sus asignaciones a equipos. El responsable es redirigido a la pestaña de Ubicaciones	
Excepciones	Paso	Acción
	5'	Si el responsable retrocede en sus acciones el caso de uso queda sin efecto

Tabla 3.22: CU019 Eliminar Ubicación

CU020	Obtención de informes	
Actor	Responsable	
Descripción	El responsable quiere obtener informes	
Precondición	El responsable está autenticado en el sistema	
Secuencia normal	Paso	Acción
	1	El responsable selecciona la opción de Informes
	2	El sistema muestra todos los fichajes realizados con información relevante de estos junto con la opción Exportar informes
	3'	El responsable selecciona la opción Exportar informes
	4	El sistema descarga un archivo Excel con la información de los informes
	5	El caso de uso finaliza
Postcondición	El sistema elimina la información del turno junto con sus asignaciones a usuarios. El responsable es redirigido a la pestaña de Equipos de trabajo	
Excepciones	Paso	Acción

Tabla 3.23: CU020 Obtención de informes

3.3. Modelo de dominio

A continuación, se muestra el modelo conceptual del dominio, mostrándose las entidades y relaciones identificadas tras el análisis de las historias de usuario, requisitos y casos de uso

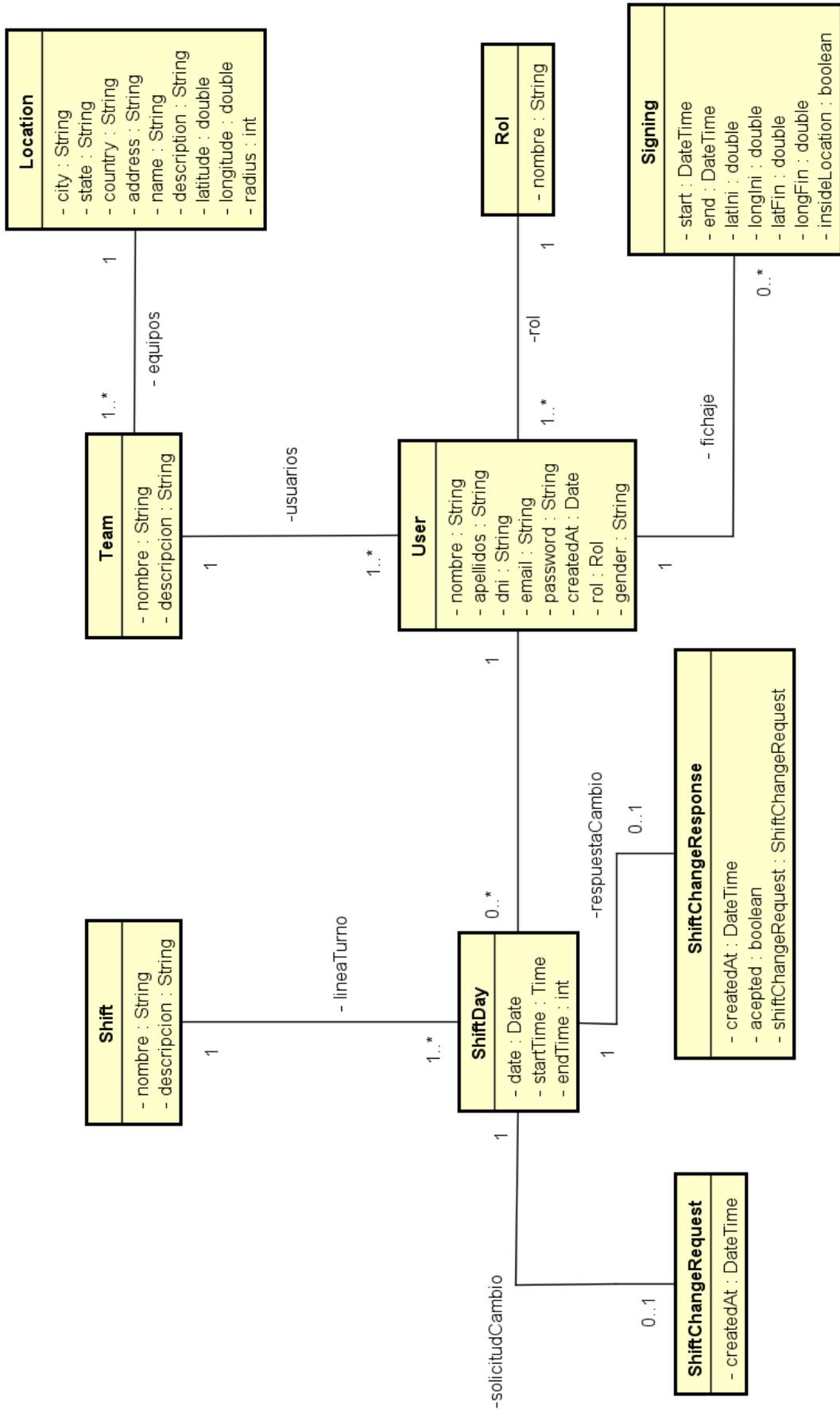


Figura 3.2: Diagrama de modelo de dominio

3.4. Diagramas de secuencia

En este apartado se mostrarán los diagramas de secuencia obtenidos del análisis a partir de los casos de uso mencionados anteriormente

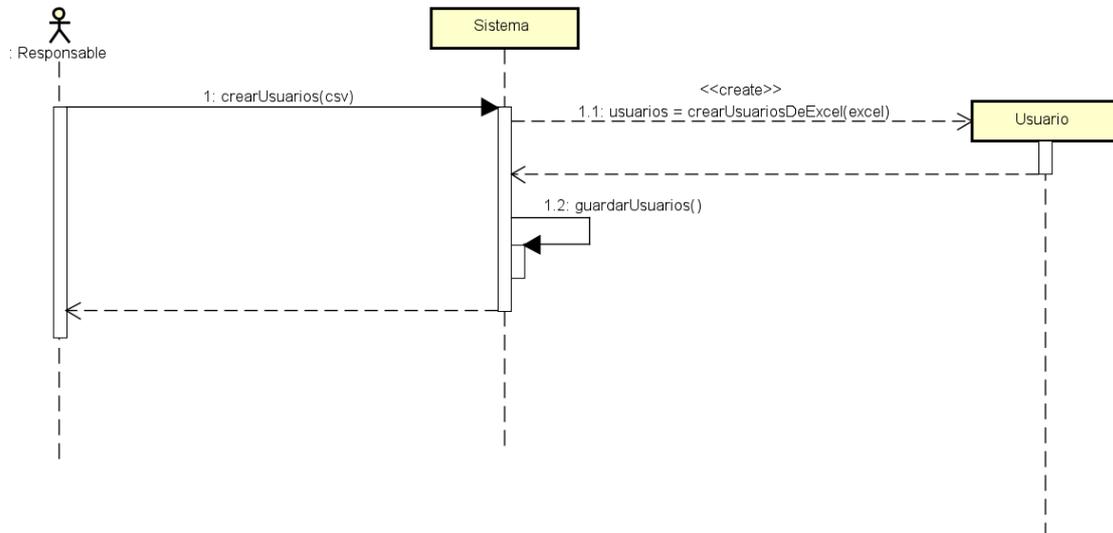


Figura 3.3: Diagrama de secuencia del CU registrar usuarios

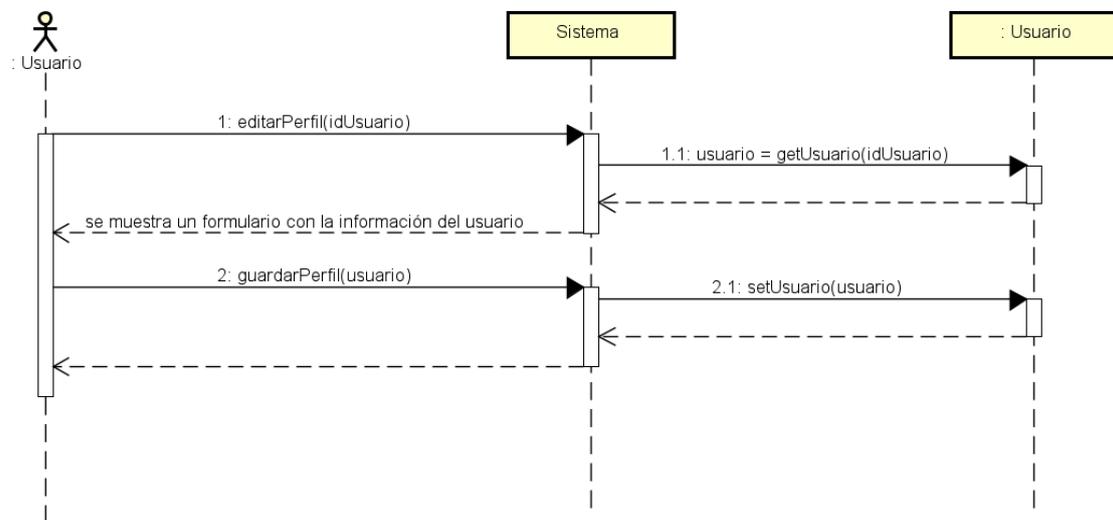


Figura 3.4: Diagrama de secuencia del CU editar perfil

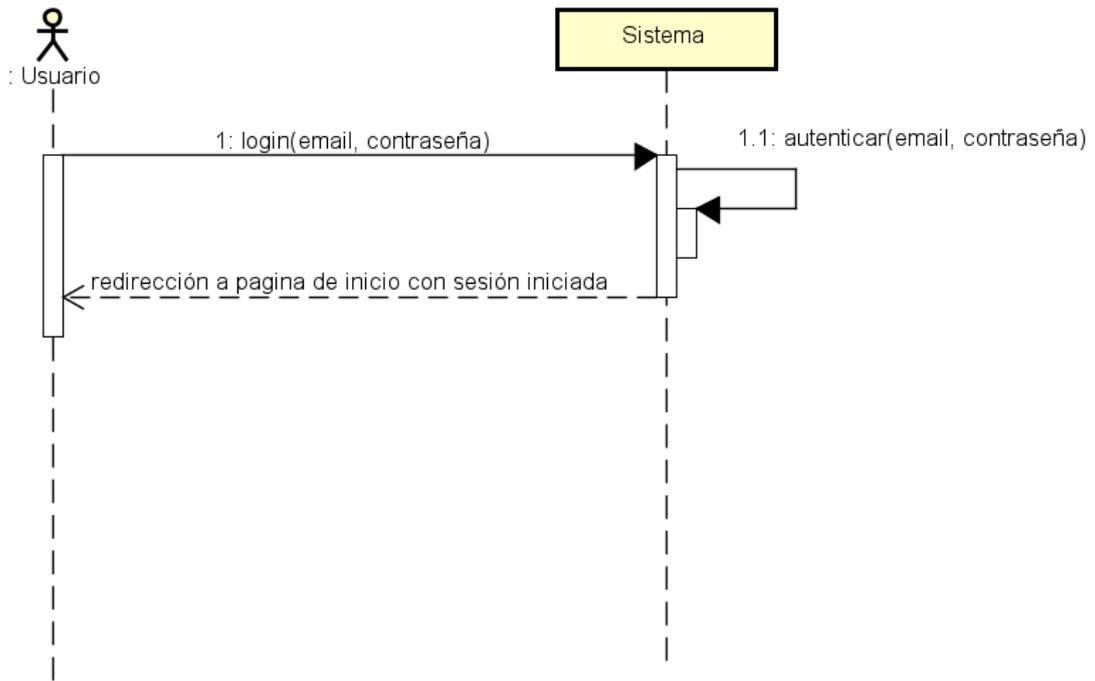


Figura 3.5: Diagrama de secuencia del CU iniciar sesión

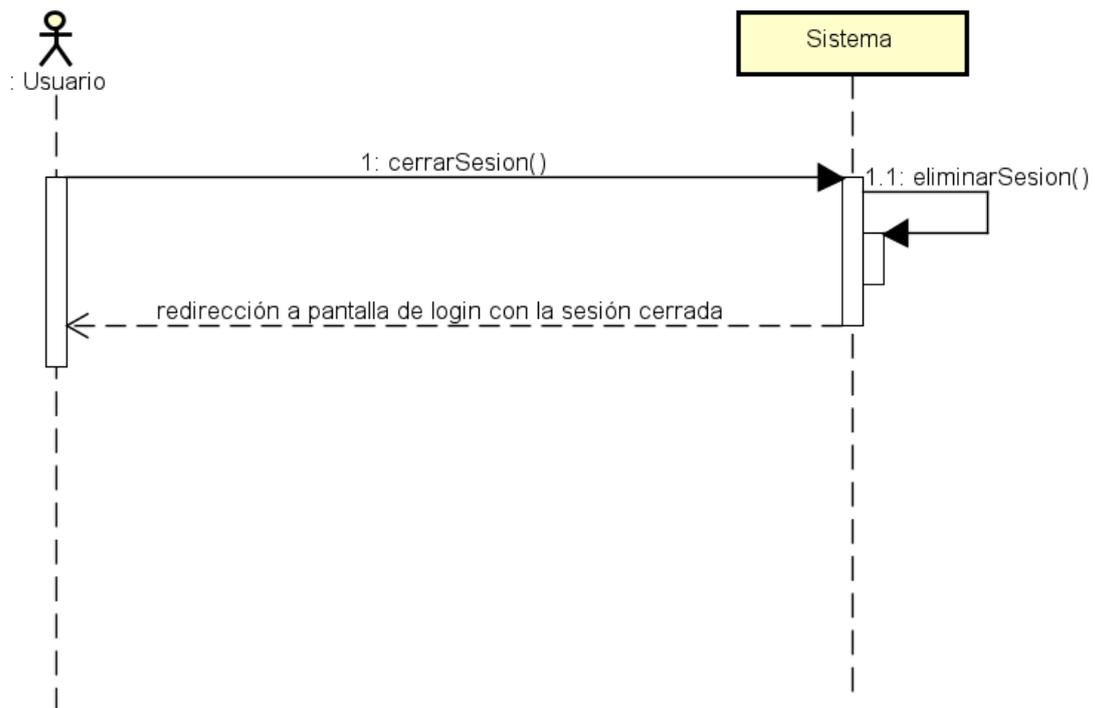


Figura 3.6: Diagrama de secuencia del CU cerrar sesión

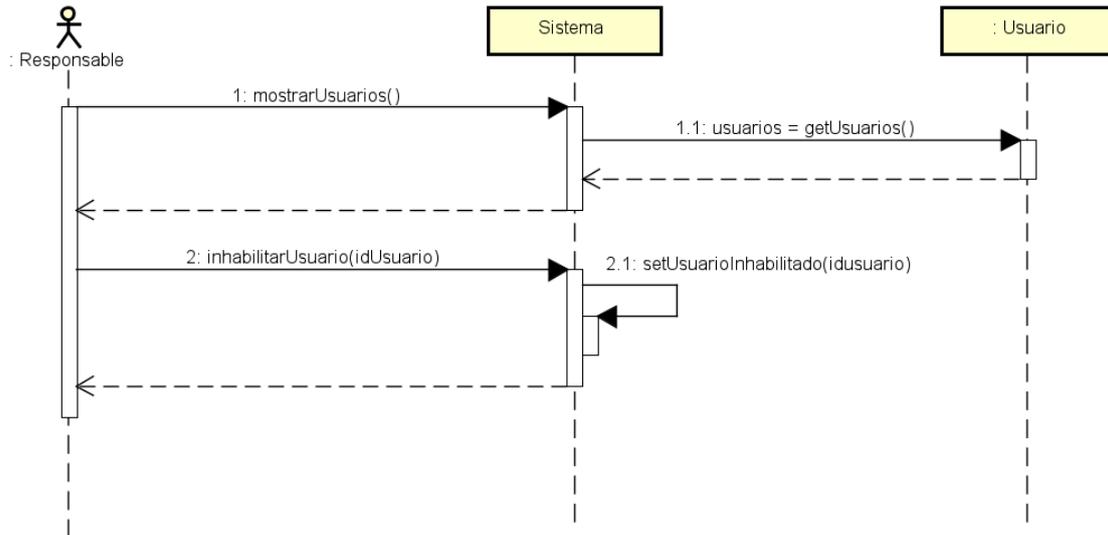


Figura 3.7: Diagrama de secuencia del CU inhabilitar usuario

Los siguientes casos de uso reflejan el registro, edición y eliminación de una entidad Equipo, sin embargo, estos diagramas son extrapolables al resto de entidades (Ubicación, Turno)

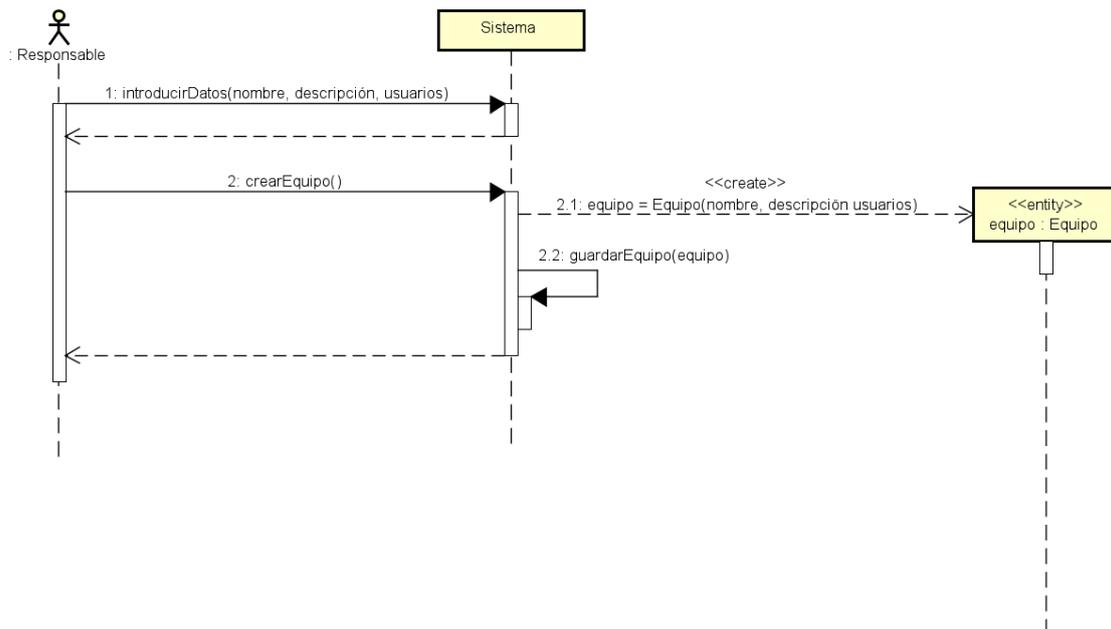


Figura 3.8: Diagrama de secuencia del CU registrar equipo

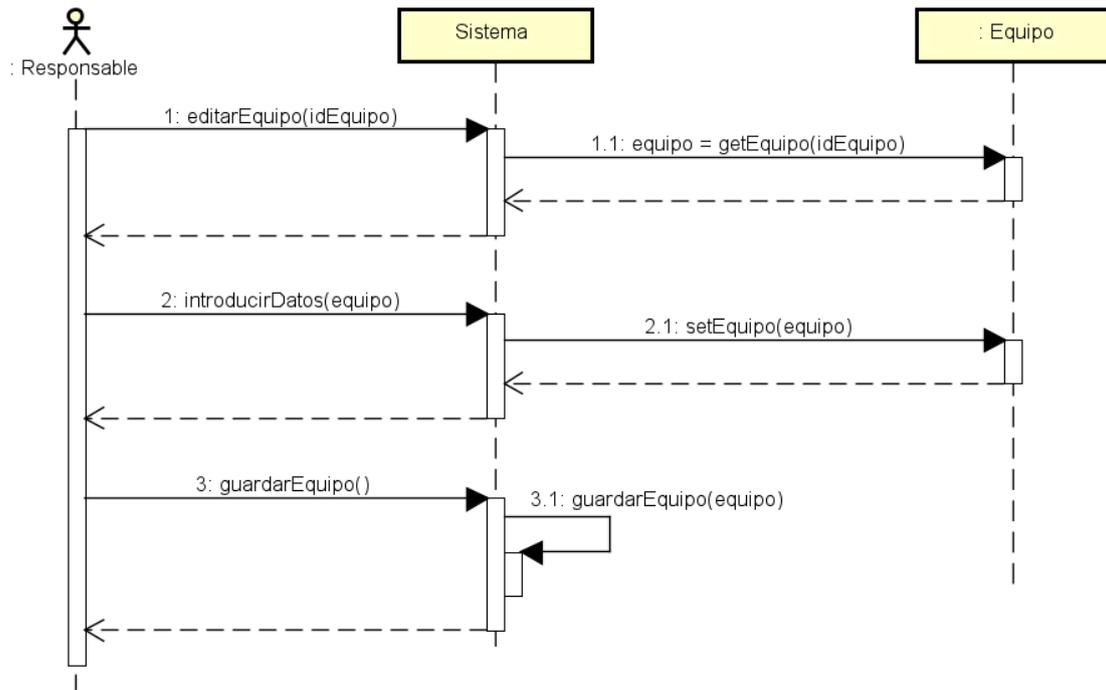


Figura 3.9: Diagrama de secuencia del CU editar equipo

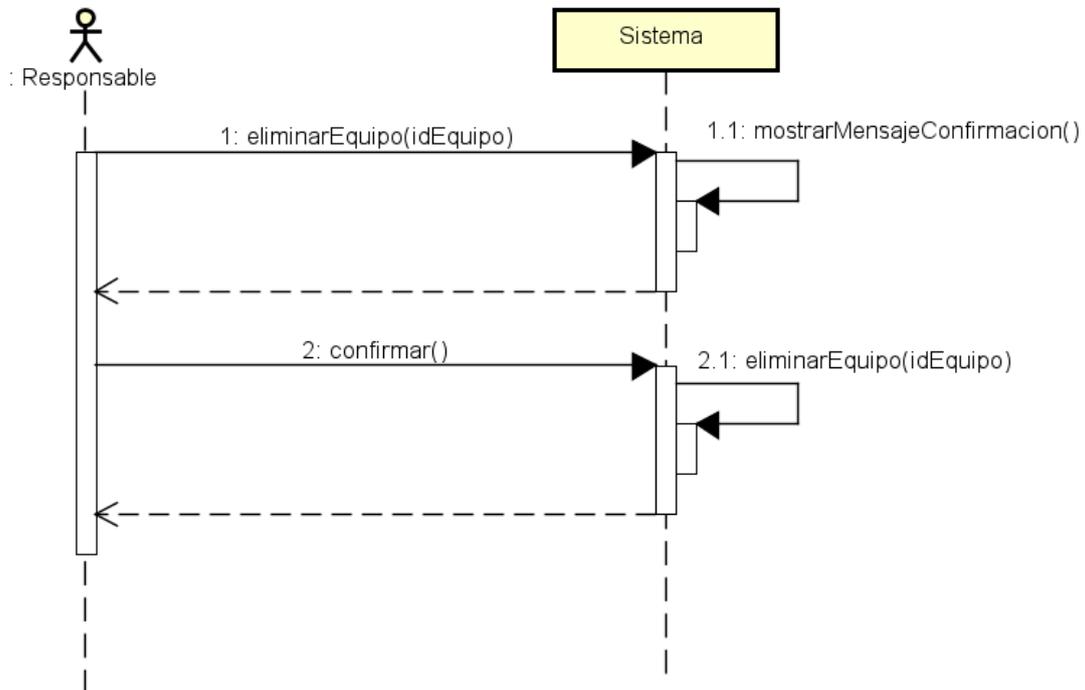


Figura 3.10: Diagrama de secuencia del CU eliminar equipo

Capítulo 4

4. Diseño

En este capítulo detallaremos la fase de diseño del proyecto software, explicando los frameworks utilizados, la arquitectura del proyecto, los patrones de diseño, el diseño de la base de datos y, por último, el diseño de la interfaz de usuario.

4.1. Patrón arquitectónico MVVM

El patrón arquitectónico MVVM ha sido utilizado dado que la parte FrontEnd de la plataforma se ha desarrollado con el framework Angular (resultante de una decisión de diseño).

Dicho patrón arquitectónico conocido como Modelo-Vista-Modelo de Vista (MVVM) se caracteriza por su objetivo de desacoplar la interfaz de usuario de la lógica (View) de la parte lógica (Model), permitiendo que la parte visual sea independiente. Este patrón consta de tres componentes principales: el Modelo, la Vista y el Modelo de Vista [14].

- Modelo: es una capa independiente que contiene la lógica del negocio.
- Vista: representa la interfaz de usuario, mostrando la información del modelo al usuario y capturando eventos de interacción. Además, puede ejecutar lógica específica de la interfaz de usuario.
- Modelo de Vista: actúa como intermediario entre la Vista y el Modelo, utilizando comandos, métodos u otros medios.

Como se observa a continuación, se ilustran las relaciones entre los componentes de este patrón, evidenciando el desacoplamiento entre ellos ya que la Vista solo conoce al Modelo de Vista, y este solo conoce al Modelo. Esta separación ofrece diversas ventajas, como la división del desarrollo de la interfaz de usuario del resto del código y la posibilidad de realizar pruebas unitarias en la lógica de presentación.

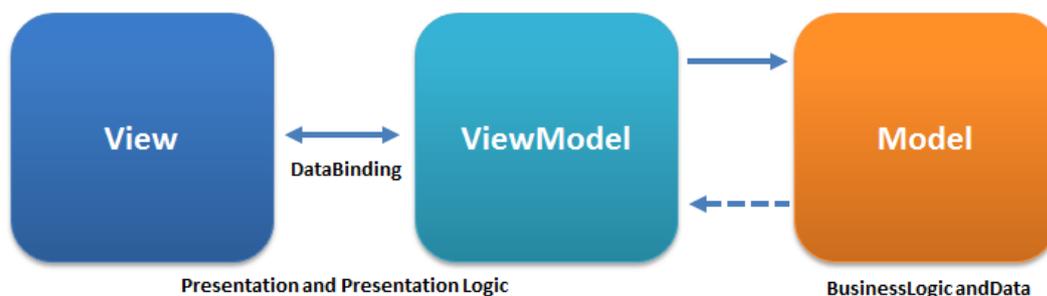


Figura 4.1: Componentes patrón MVVM [15]

4.2. Patrones de diseño

4.2.1. Patrón DAO/DTO

Este patrón hace referencia a dos patrones individuales que se usan conjuntamente:

- Patrón DAO (Data Access Object): patrón de diseño software que proporciona una abstracción de alto nivel para acceder y gestionar los datos en una base de datos u otro origen de datos. Su objetivo principal es separar la lógica de acceso a datos de la lógica de negocio, proporcionando una clase DAO que implementa la interfaz de operaciones que se pueden realizar con la información de la fuente de datos [16].
- Patrón DTO (Data Transfer Object): patrón de diseño software que proporciona un objeto DTO que transporta datos entre diferente capas o componentes. Su objetivo principal es encapsular y transportar datos de manera eficiente reduciendo el número de llamadas [17].

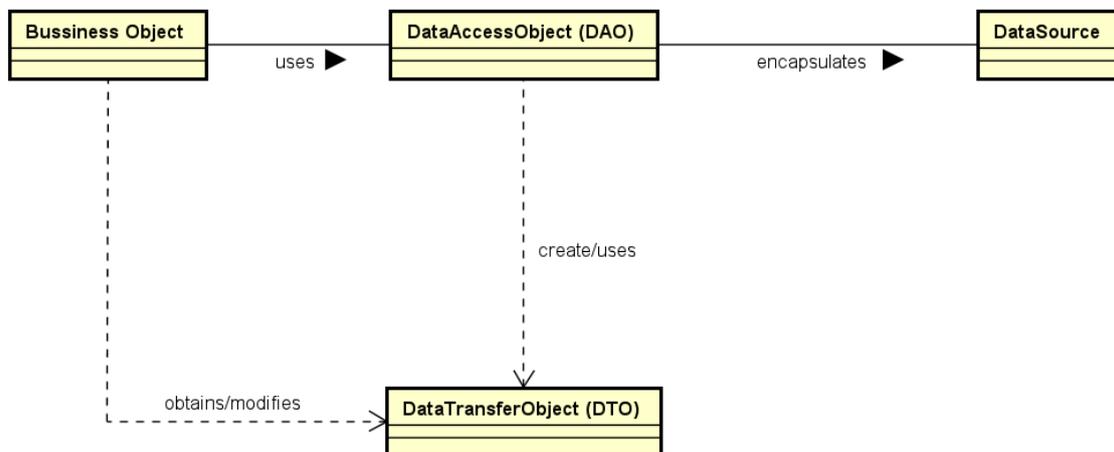


Figura 4.2: Patrón DAO/DTO

En nuestra aplicación, los objetos DAO se encuentran en el paquete *Repository*, facilitando las implementaciones básicas de acceso a datos CRUD (Create Read Update Delete).

A su vez, los objetos DTO están alojados en el paquete *Entitie* como entidades del dominio, permitiéndose así que la información se transmita entre las diferentes capas.

4.2.2. Patrón Repository

El patrón de diseño Repository gestiona y abstrae el acceso a la capa de persistencia de datos en un sistema software. El objetivo principal de este es proporcionar una interfaz consistente y orientada al dominio para interactuar con la capa de persistencia, ocultando los detalles específicos de la tecnología de almacenamiento subyacente [18].

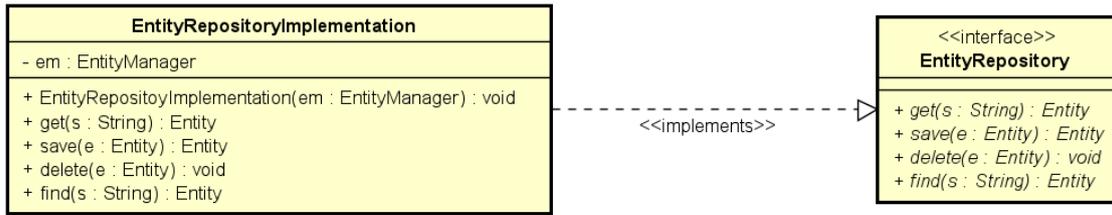


Figura 4.3: Patrón Repository

El patrón Repository lo podemos ver aplicado en los objetos DAO del paquete *Repository*, los cuales extienden *JpaRepository*, implementando así los métodos presentados en la Figura 4.3.

4.3. Desarrollo basado en componentes

La complejidad de los sistemas informáticos actuales nos hace querer reutilizar los sistemas existentes. La programación de componentes le permite reutilizar partes de código desarrollado previamente para realizar diferentes tareas, lo que brinda muchos beneficios, como una mejor administración, menos desarrollo y un mayor retorno de la inversión [19].

4.3.1. Componentes en Angular

Angular utiliza un modelo de desarrollo basado en componentes. La estructura básica de un componente se divide en dos partes, una clase TypeScript y un template o plantilla HTML [20].

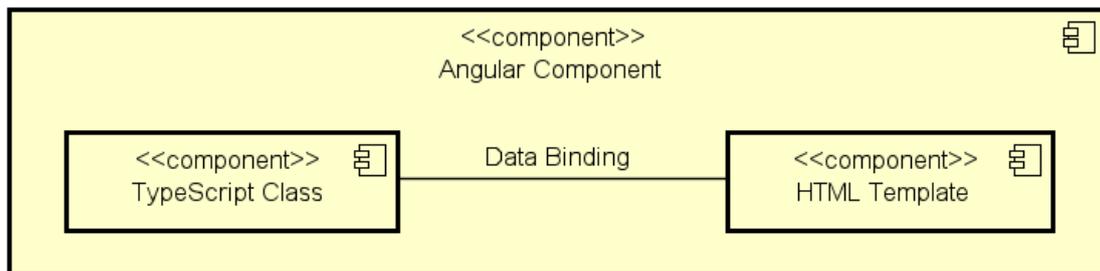


Figura 4.4: Estructura de un componente Angular

Como hemos explicado, el desarrollo basado en componentes se basa principalmente en el concepto de reutilización, por lo que un componente Angular puede reutilizarse dentro de otros componentes, creándose así una jerarquía padre-hijo, donde el componente hijo (componente reutilizado) se inserta en el HTML Template del componente padre.

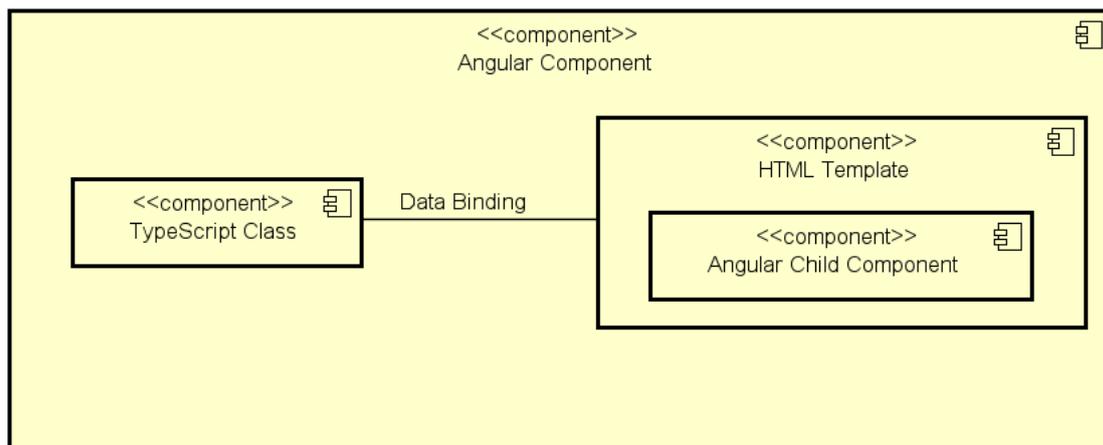


Figura 4.5: Estructura de la reutilización de un componente Angular en otro

Por último, los componentes en sí pueden necesitar de un artefacto auxiliar también reutilizable que los ayude a realizar su función, estos artefactos son los servicios.

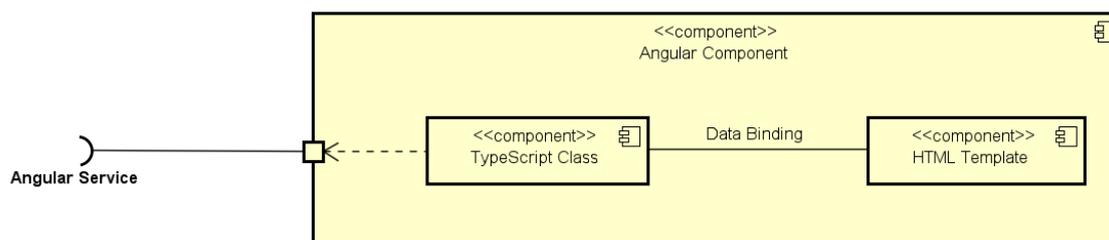


Figura 4.6: Estructura de un componente Angular que requiere un servicio

4.3.2. Componentes creados en el frontend

En esta sección se listarán los diferentes componentes Angular creados para este proyecto describiendo la funcionalidad que implementan.

- **App:** Contiene toda la aplicación.
- **Home:** Página de inicio o escritorio de la aplicación. Contiene enlaces a otras páginas.
- **Login:** Página de inicio de sesión para los usuarios ya registrados en la plataforma.
- **Header:** Cabecera de la aplicación, que contiene la edición de los datos del usuario autenticado en la plataforma junto con el cierre de sesión.
- **Users:** Página de listado de los usuarios de la plataforma e importación de nuevos.
- **Teams:** Página de listado y eliminación de los equipos registrados en la plataforma.

- **Team-detail:** Página donde se muestra el detalle de la edición de un equipo existente o se registra uno nuevo.
- **Locations:** Página de listado y eliminación de las ubicaciones registradas en la plataforma.
- **Location-detail:** Página donde se muestra el detalle de la edición de una ubicación o se registra una nueva.
- **Shifts:** Página de listado y eliminación de los turnos registrados en la plataforma.
- **Shift-detail:** Página donde se muestra el detalle de la edición de un turno o se registra uno nuevo.
- **Signing:** Página donde se muestra tanto la ubicación del usuario como la preasignada a su equipo. El usuario puede registrar el inicio y fin de fichajes laborales.
- **Schedule:** Página que contiene enlaces al horario del usuario y a la sección de solicitudes de cambio.
- **Schedule-detail:** Página que muestra en detalle el horario del usuario.
- **Schedule-change:** Página que muestra las solicitudes de cambio y las peticiones de cambio del propio usuario.
- **Schedule-change-detail:** Página que muestra en detalle una solicitud de cambio de horario para aceptarla.
- **Reports:** Página que muestra los informes de fichajes realizados en la plataforma junto con la exportación a Excel de estos.

A continuación, se muestra un diagrama que recoge los diferentes componentes mencionados.

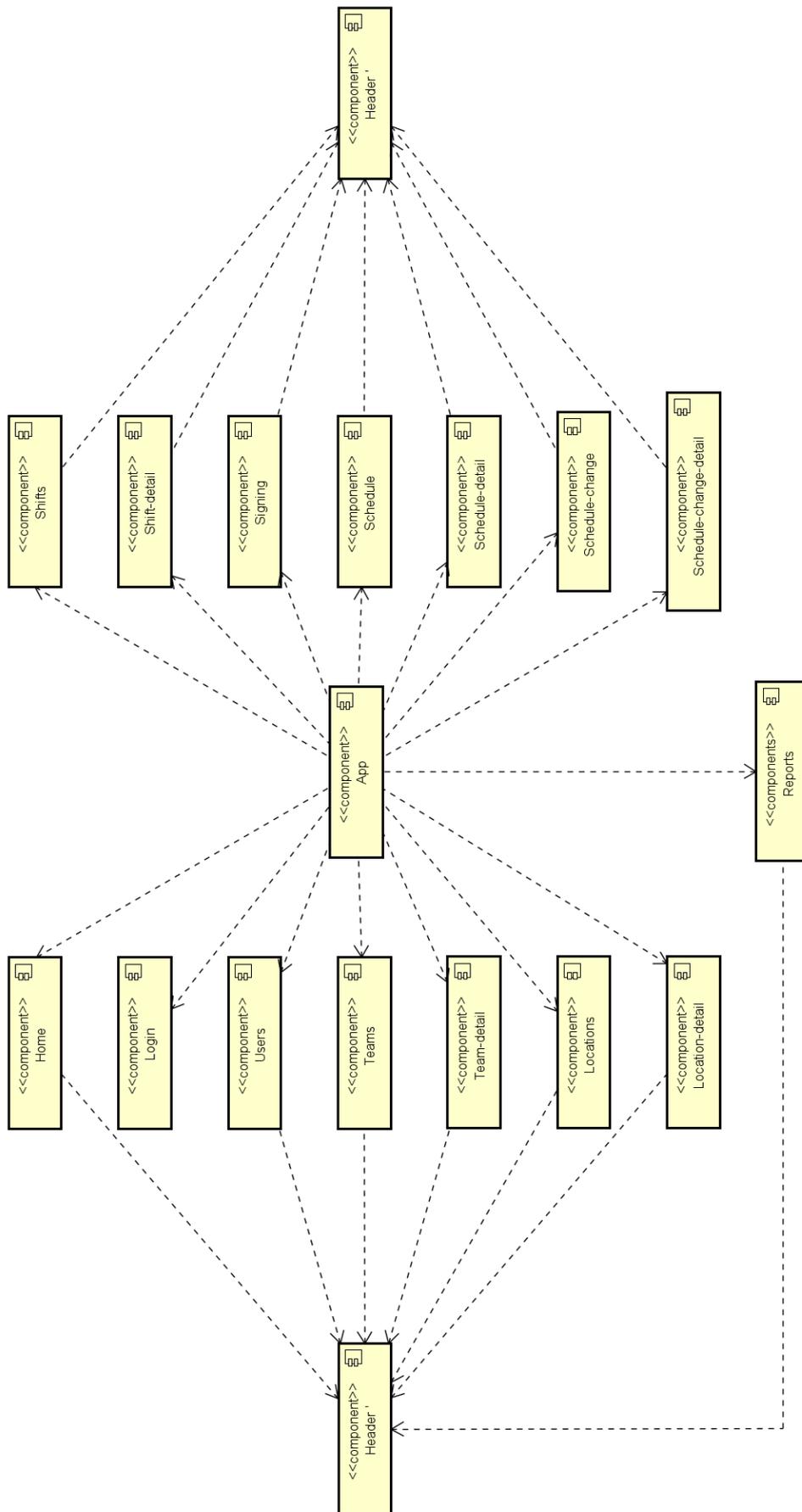


Figura 4.7: Estructura de componentes del frontend

Cabe mencionar que los componentes Header y Header ´ hacen referencia al mismo componente, sin embargo, por facilidad en la elaboración del diagrama y por una mera cuestión estética se ha representado dos veces.

4.4. Diseño arquitectónico del backend

En este apartado se listaran tanto los diferentes paquetes que están incluidos en la arquitectura del backend como diagramas que los reflejan, cabe mencionare que todos ellos se encuentran bajo el paquete *com.uva.JorgeDeBustosTfg*.

- **Controller:** En este paquete se incluyen los controladores, es decir, los objetos encargados de gestionar las distintas solicitudes HTTP que emite la parte frontend.

Cada controlador se ha hecho bajo la premisa de división de funcionalidad y entidades que manejan, como, por ejemplo, la clase *UsuariosController*, que se ha definido para gestionar todos aquellos procesos en los que se desea interactuar con la entidad *Usuario*.

- **Repository:** En este paquete se han incluido aquellas clases que son abstracciones y proporcionan las operaciones necesarias relativas a una clase de dominio para interactuar con la base de datos.
- **Model:** En este paquete se localizan las clases de las entidades que forman el dominio, más en concreto, en el paquete **Entities**, donde se encuentran los objetos.
- **Requests:** En este paquete se han incluido las clases utilizadas para validar el formato de datos que tienen que cumplir diversas peticiones HTTP para poder ser tramitas por distintos endpoints de los controladores.
- **Response:** En este paquete se ha incluido la clase que permite a los controladores dar respuesta a la petición HTTP que han recibido.

Al utilizar siempre la misma clase en la respuesta al lado del cliente se consigue gestionar de una manera unificada tanto peticiones correctas y erróneas.

- **Security:** En este paquete se han almacenado las clases que implementan la seguridad de acceso, fiabilidad de las peticiones HTTP contra el backend y generación de tokens JWT de acceso.

A continuación, podemos observar distintos diagramas que reflejan la arquitectura del backend que hemos mencionado.

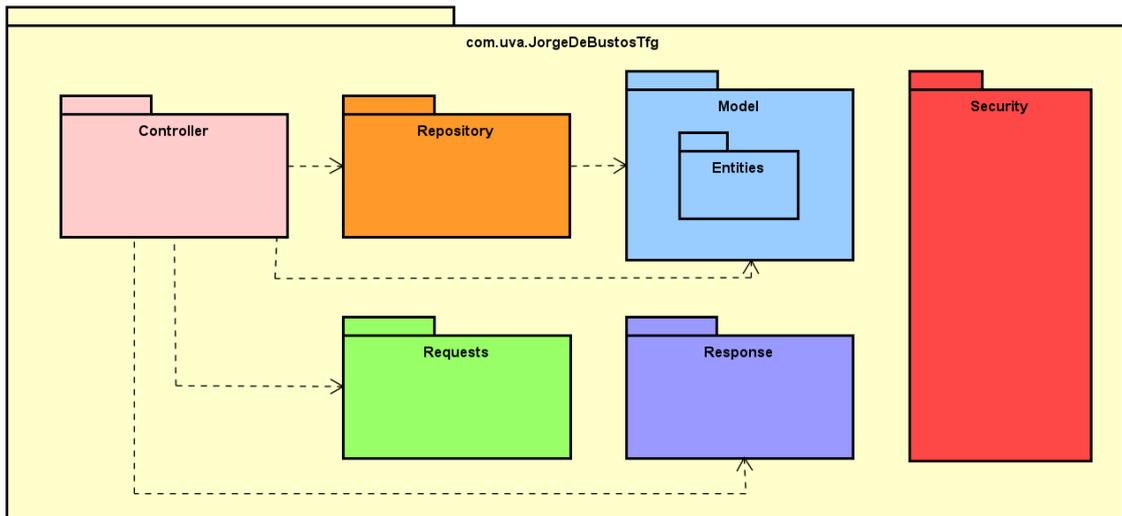


Figura 4.8: Arquitectura general del backend

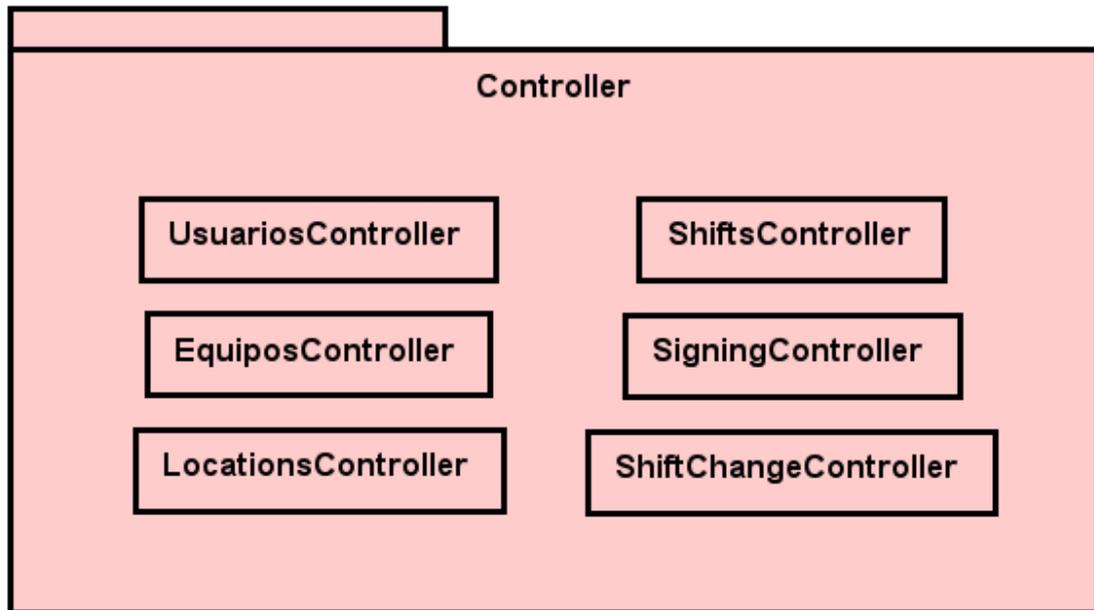


Figura 4.9: Modules&Uses Style del paquete Controller

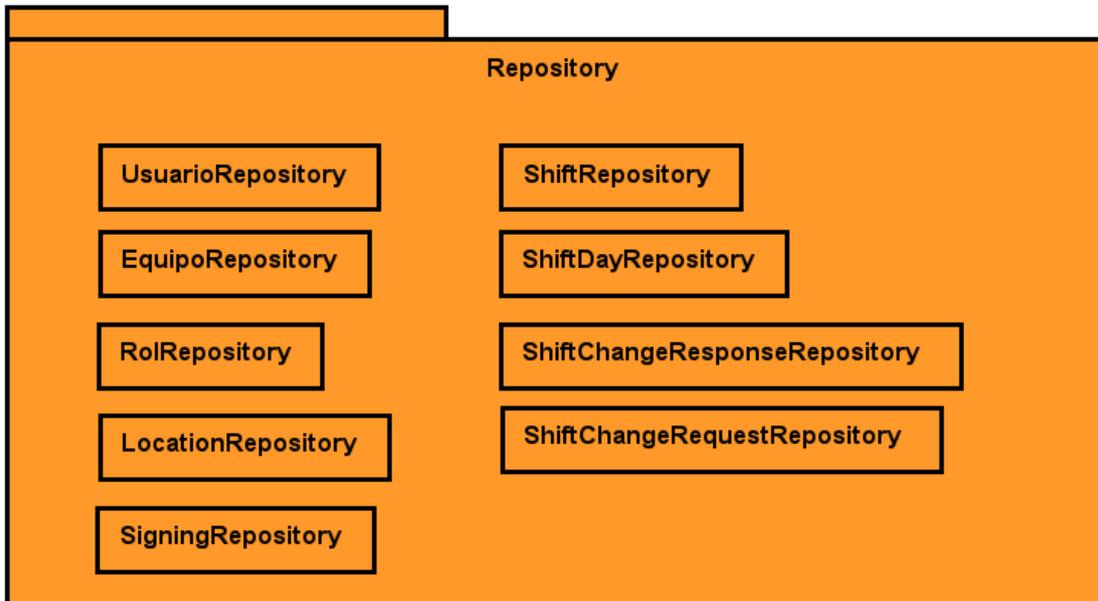


Figura 4.10: Modules&Uses Style del paquete Repository

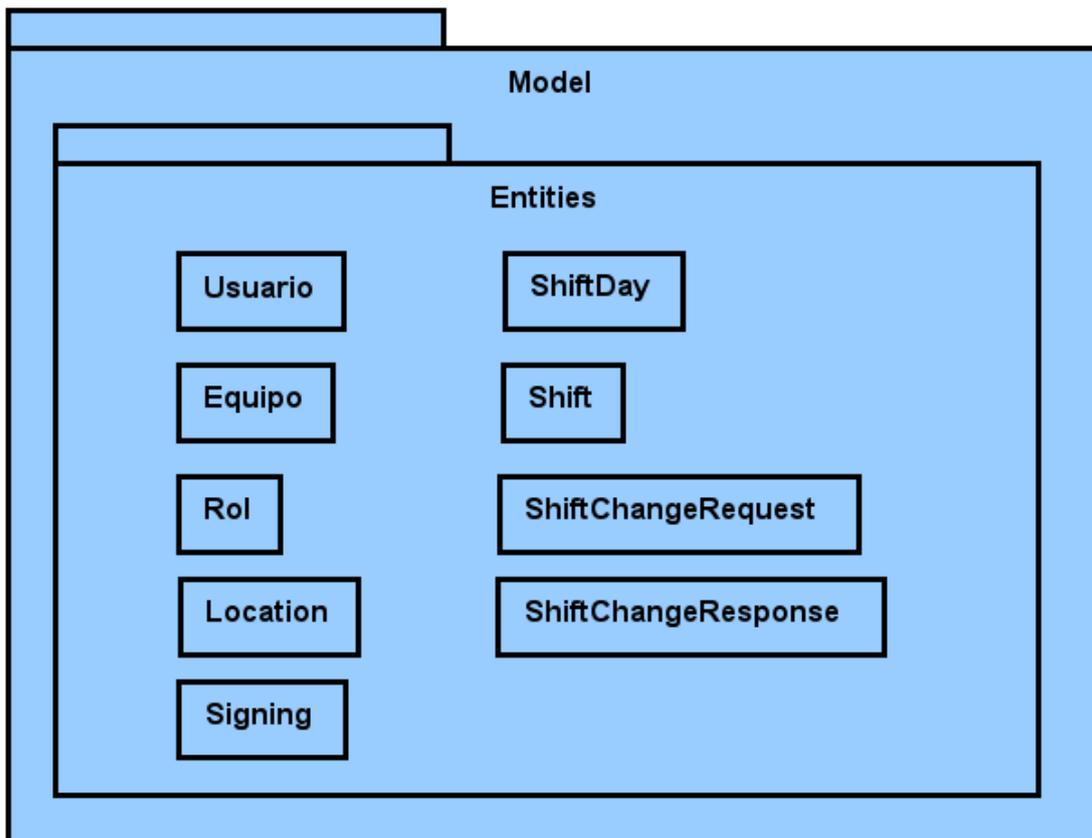


Figura 4.11: Modules&Uses Style del paquete Model y Entities

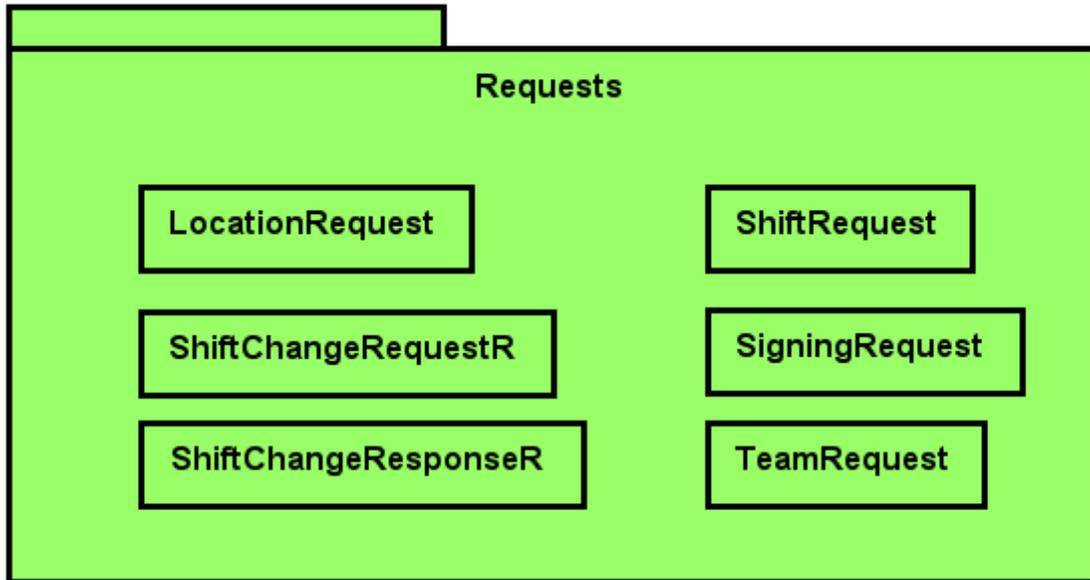


Figura 4.12: Modules&Uses Style del paquete Requests

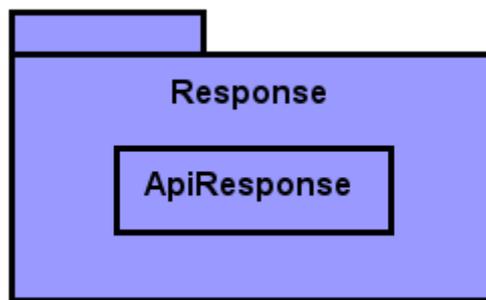


Figura 4.13: Modules&Uses Style del paquete Response

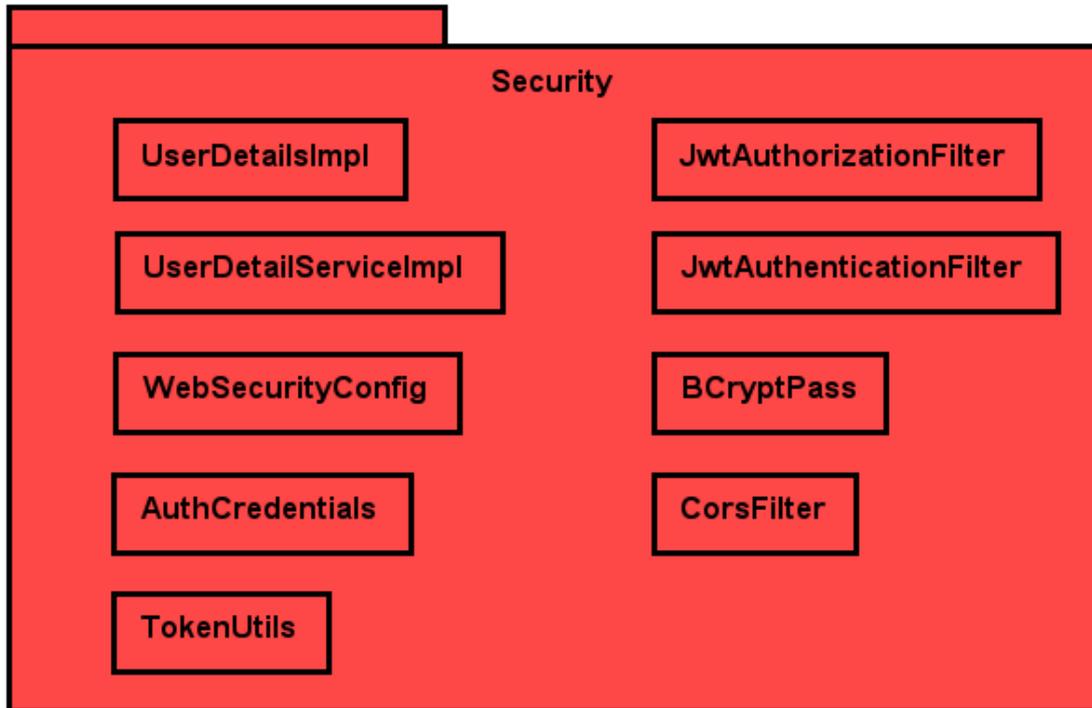


Figura 4.14: Modules&Uses Style del paquete Security

4.5. Diseño de base de datos

El diseño lógico relacional de la base de datos utilizará un sistema gestor de base de datos MariaDB, se muestra a continuación.

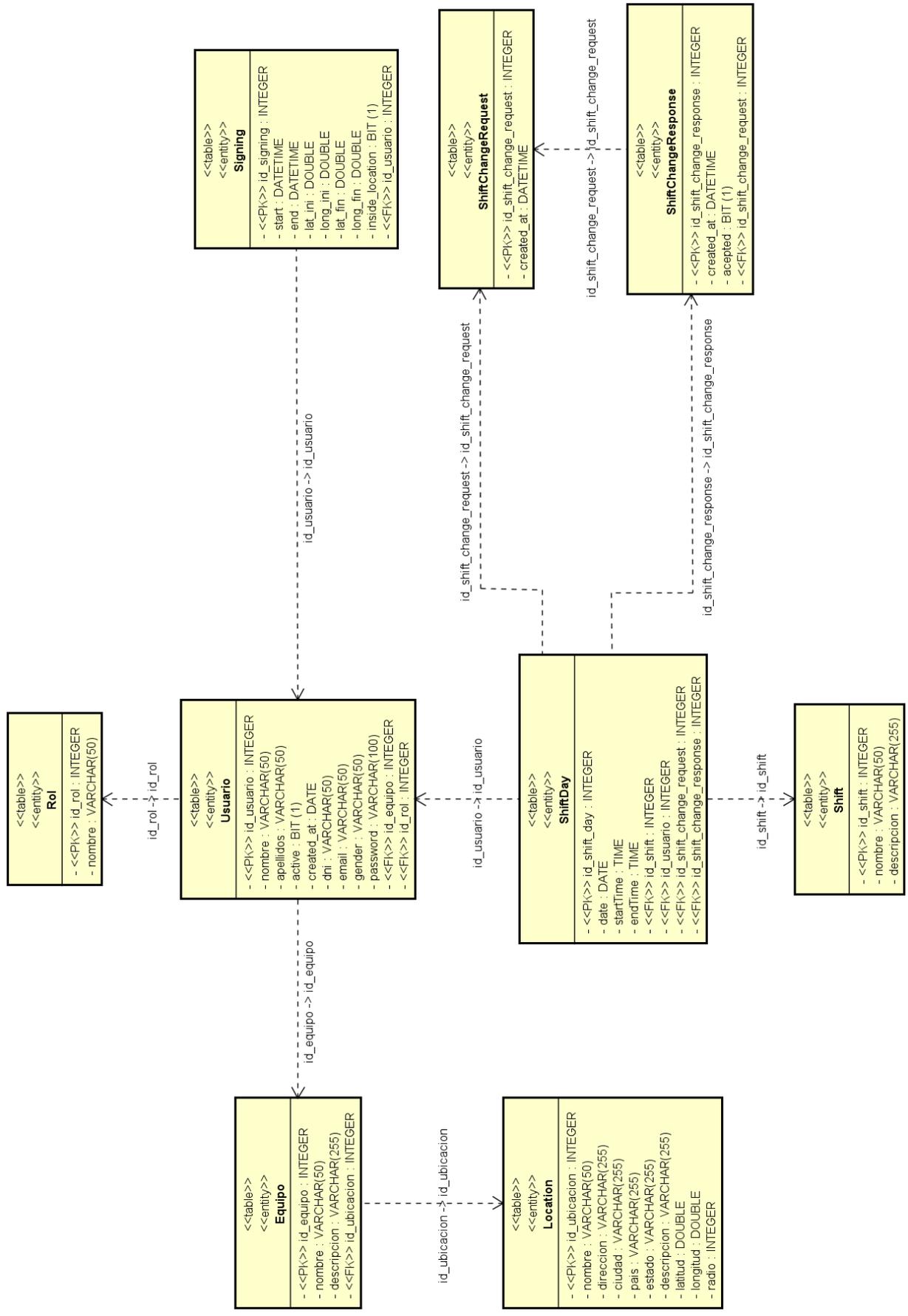


Figura 4.15: Modelo de datos de la aplicación

4.6. Despliegue de plataforma

Para el despliegue de la plataforma se ha optado por el servicio en la nube Heroku, que permite manejar servidores y su configuración, bases de datos y la administración y mantenimiento que estos conllevan.

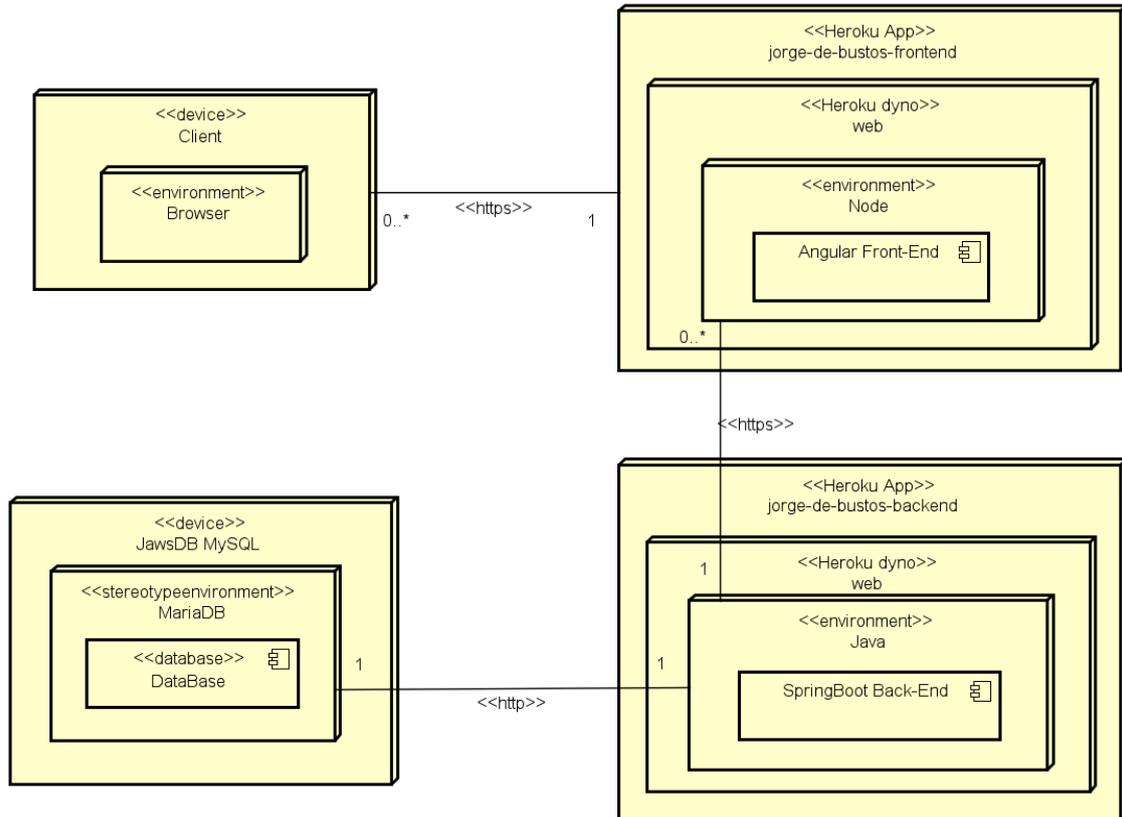


Figura 4.16: Diagrama de despliegue

4.7. Privacy by Design

En el desarrollo de aplicaciones web, es crucial garantizar la privacidad y la protección de los datos personales de los usuarios desde el inicio del proyecto. El enfoque de "Privacy by Design" (Privacidad desde el Diseño) consiste en garantizar la intimidad y la protección de los datos personales desde el momento en que se inicia un proyecto [\[21\]](#).

Como parte de nuestro enfoque en la privacidad y la transparencia, se da por supuesto que los usuarios han aceptado una "Disclosure Policy" (Política de Divulgación), que establece las condiciones y términos relacionados con la recopilación, el uso y la divulgación de sus datos personales dentro de la aplicación, puesto que la empresa arrendataria pide a sus trabajadores que lo acepten para así poder tratar sus datos.

Además, como parte de nuestras prácticas de seguridad, todas las contraseñas de los usuarios son generadas y almacenadas de forma encriptada gracias al uso de la clase *BCryptPasswordEncoder* propia de Spring Boot (la cual utiliza una función de hashing de contraseñas basada en el cifrado

blowfish). Esto asegura que las contraseñas estén protegidas y no sean accesibles para terceros, brindando una capa adicional de seguridad para los usuarios y sus datos personales.

4.8. Interfaz de usuario

4.8.1. Atributos de usabilidad

Los atributos de usabilidad de un sistema software son aquellas características y cualidades que definen la facilidad de uso y satisfacción de un usuario al interactuar con dicho sistema.

En este proyecto cabe destacar que se han aplicado los siguientes atributos:

- **Facilidad de aprendizaje:** tanto los casos de uso, como los flujos alternativos que plantea y permite el sistema, tienen una rápida y sencilla curva de aprendizaje debido a que no requieren gran intuición, si no que las opciones que se presentan son claras y concisas.

En la plataforma se puede apreciar en ejemplos tales como, la aparición de migas de pan que indican al usuario en todo momento su ubicación dentro de la estructura de diferentes páginas.

- **“Feedback” y respuesta:** el sistema informa del estado final de cada caso de uso, tanto como si se ha realizado con éxito como si no, indicando a que se debe dicho fallo en caso de que pueda informar.
- **Comprensión intuitiva:** los elementos de la interfaz se representan siempre con elementos de texto descriptivos o en su defecto, con iconos fielmente asociados a su función, es decir, la acción de editar se refleja mediante el icono de un lapicero, al igual que eliminar se ilustra con una papelera, así como botones de aceptar y eliminar mantienen colores uniformes a lo largo de la plataforma como es el verde y rojo, respectivamente.
- **Tolerancia a errores:** el sistema previene gran cantidad de errores que podrían generarse en los casos de uso, principalmente gestionando que la información que puede utilizar el usuario es válida, como en el caso del Registro de Equipos, los usuarios que se permiten asociar a un Equipo son aquellos que previamente no tienen ningún otro asignado, evitando fallos o sobreescritura de información.
- **Estética visual:** la plataforma se ha realizado utilizando una visual paleta de colores junto con los formatos “Card” que presenta el framework utilizado, obteniendo así vistas sencillas, atractivas y limpias

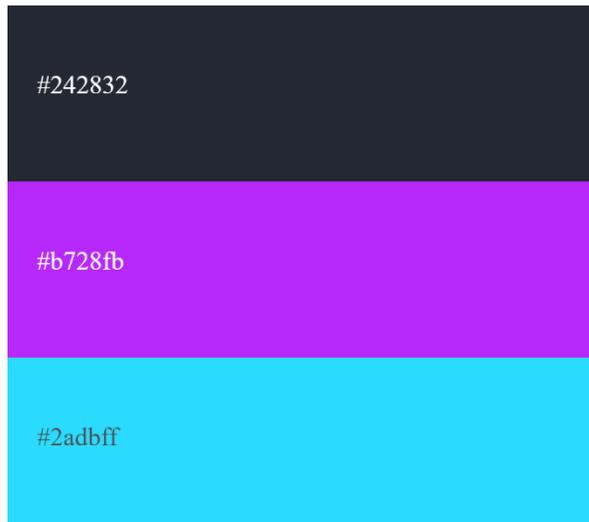


Figura 4.17: Paleta de colores utilizada en la plataforma

4.8.2. Bocetos de interfaz de usuario

Los bocetos que se presentarán a continuación han sido creados para dos tipos de dispositivos con diferentes resoluciones.

Cabe destacar que se han diseñado solamente 4 bocetos originales puesto que la idea de diseño, disposición y estilo estaba clara desde el principio del desarrollo.

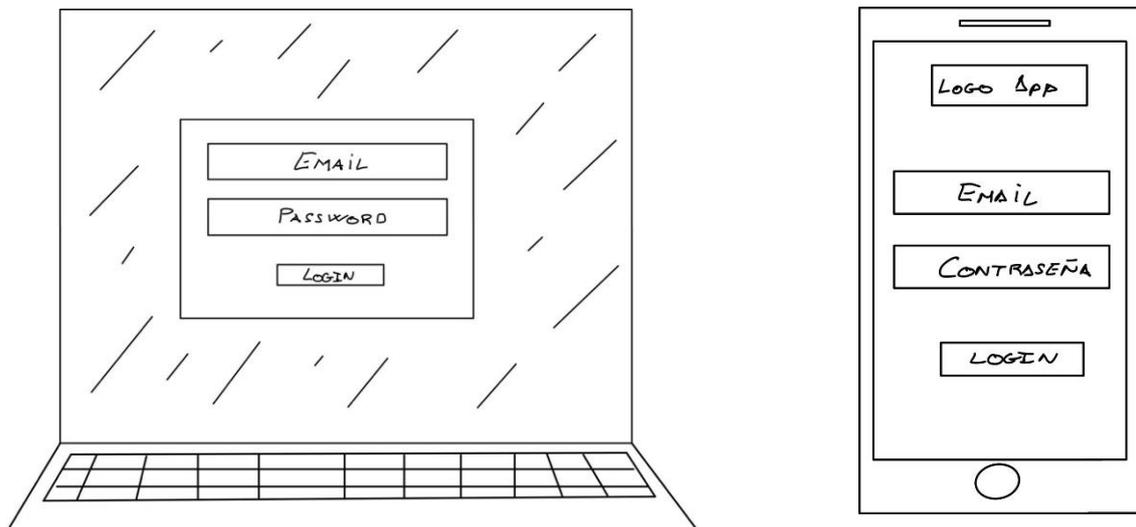


Figura 4.18: Boceto de interfaz de la pantalla de Login

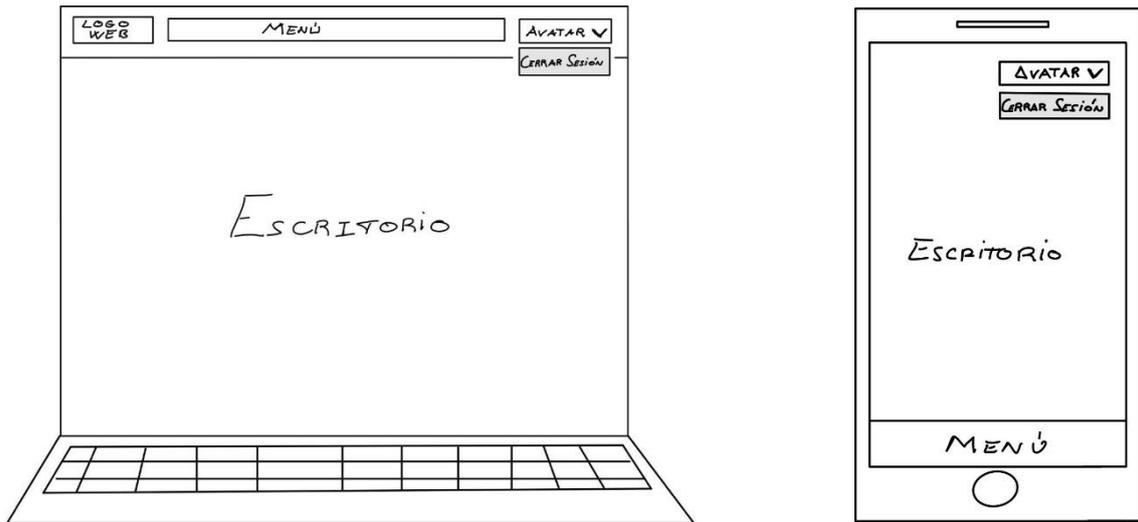


Figura 4.19: Boceto de interfaz de la pantalla de Escritorio

Capítulo 5

5. Tecnología utilizada

En este capítulo se presentarán las herramientas y tecnologías utilizadas durante el desarrollo de este proyecto.

5.1. Microsoft 365

Microsoft 365 es una solución de software como servicio (SaaS) desarrollada por Microsoft y que incluye Microsoft Office y otros servicios, como correo electrónico y colaboración, desde el servidor en la nube de Microsoft [\[22\]](#).

En este proyecto se han utilizado dos de las herramientas que Microsoft provee, por un lado, Teams, una plataforma de comunicación y colaboración en línea que permite comunicarse, colaborar y compartir información, en este caso, con la tutora del TFG, y por otro, Word, una aplicación de procesamiento de texto utilizada para realizar este documento.

5.2. Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

También cuenta con una gran cantidad de extensiones, que posibilitan crear y ejecutar código en cualquier lenguaje de programación [\[23\]](#).

5.3. Astah Profesional

Astah Profesional es una herramienta de modelado visual y diseño de software utilizada para realizar diagramas y modelos basados en UML (Unified Modeling Language), el lenguaje de modelado de sistemas software referente y estándar actualmente.

Astah permite hacer una gran variedad de diagramas y modelos como diagramas de relación de entidades, de flujo, de casos de uso, de clases, de secuencia, de componentes, de despliegue etc [\[24\]](#).

5.4. Git

Git es un sistema de control de versiones distribuido utilizado en el desarrollo software que permite sincronizar la una copia del código de un repositorio local con la copia en el servidor, basado en un sistema de repositorios y ramas [25].

Un repositorio es un almacén virtual donde se guardan todos los cambios y archivos de un proyecto vinculado a él, mientras que una rama es una línea de desarrollo independiente dentro del repositorio, es decir, una “copia” del código sobre la que realizar cambios, correcciones etc. sin afectar al código ubicado en la línea de desarrollo padre de la que esta ha surgido.

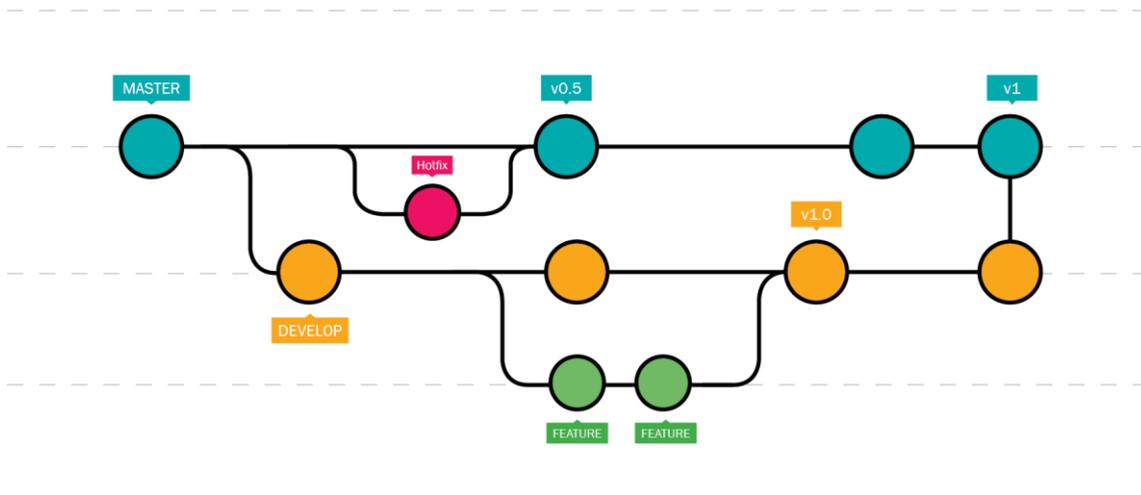


Figura 5.1: Esquema funcionamiento ramas Git [26]

5.5. PostMan

PostMan es una herramienta de creación de peticiones HTTP utilizada para probar y realizar solicitudes a servicios web, como la API (Application Programming Interface) que se ha desarrollado en este proyecto. PostMan proporciona una interfaz simple e intuitiva que permite interactuar con una API de forma eficiente bajo un entorno aislado y controlado.

5.6. Angular

Angular es un framework de ingeniería de software de código abierto mantenido por Google, que sirve para desarrollar aplicaciones web de estilo Single Page Application (SPA) y Progressive Web App (PWA) [27]. Angular se basa en el lenguaje de programación TypeScript y utiliza el patrón de arquitectura Modelo-Vista-Modelo de Vista (MVVM) como se ha explicado en el apartado 4.1.

En Angular, las aplicaciones se construyen utilizando componentes reutilizables que encapsulan la lógica y la interfaz de usuario de partes específicas de la aplicación. Estos componentes están formados por un template (archivo HTML), una hoja de estilos (SCSS o CSS) y el archivo de la lógica (archivo TypeScript). Además, estos componentes poseen un ciclo de vida que abarca desde la creación de este hasta su destrucción.

El frontend de la plataforma ha sido desarrollada en Angular.

5.7. Spring Boot

Spring Boot es un framework de desarrollo de aplicaciones Java que nos permite crear aplicaciones autocontenidas, obviando así la arquitectura y enfocándonos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias mediante un gestor de dependencias *Maven* o *Gradle*, desplegar nuestro servicio o aplicación a un servidor de aplicaciones. Para conseguir esto, Spring Boot hace un uso interno de un servidor de aplicaciones embebido, por defecto, Tomcat [\[28\]](#).

La parte *Backend* de la plataforma ha sido desarrollada en Spring Boot.

5.8. JWT

JWT (JSON Web Token) es un estándar que está dentro del documento RFC 7519, basado en JSON utilizado para la creación de tokens de seguridad y autenticación, que contiene la identidad de un determinado usuario, además de una serie de claims o privilegios [\[29\]](#).

El contenido de un JWT se distingue en tres partes:

- **Header:** encabezado dónde como mínimo se indica el algoritmo de encriptación y el tipo de token.
- **Payload:** cuerpo del token, donde aparecen los datos del usuario junto con sus privilegios, así como la información que se desee añadir.
- **Signature:** firma digital que permite verificar la validez del token.

5.9. HeidiSQL

HeidiSQL es una herramienta ligera de gestión de bases de datos de código abierto. Permite administrar y manipular bases de datos MySQL, MariaDB, PostgreSQL y Microsoft SQL Server de manera eficiente.

Hace uso de los estilos VCL nativos de Delphi de manera muy efectiva para proporcionar una interfaz atractiva que el desarrollador puede personalizar según sus preferencias [\[30\]](#).

5.10. MariaDB

MariaDB es un sistema de gestión de bases de datos relacionada con MySQL y que incorpora sus características junto con diversas mejoras, como la ejecución de consultas SQL complejas, un sistema de almacenamiento caché de estas, acceso a clusters de datos y soporte de jerarquías de grafos [\[31\]](#).

Capítulo 6

6. Implementación y Pruebas

En este capítulo se expondrá la implementación de la plataforma exponiendo la estructura del código tanto en el front-end como en el back-end y las pruebas llevadas a cabo para validar el correcto funcionamiento de esta.

6.1. Estructura del código frontend

A continuación, se muestra unas imágenes que describe la estructura final del código que presenta el frontend.

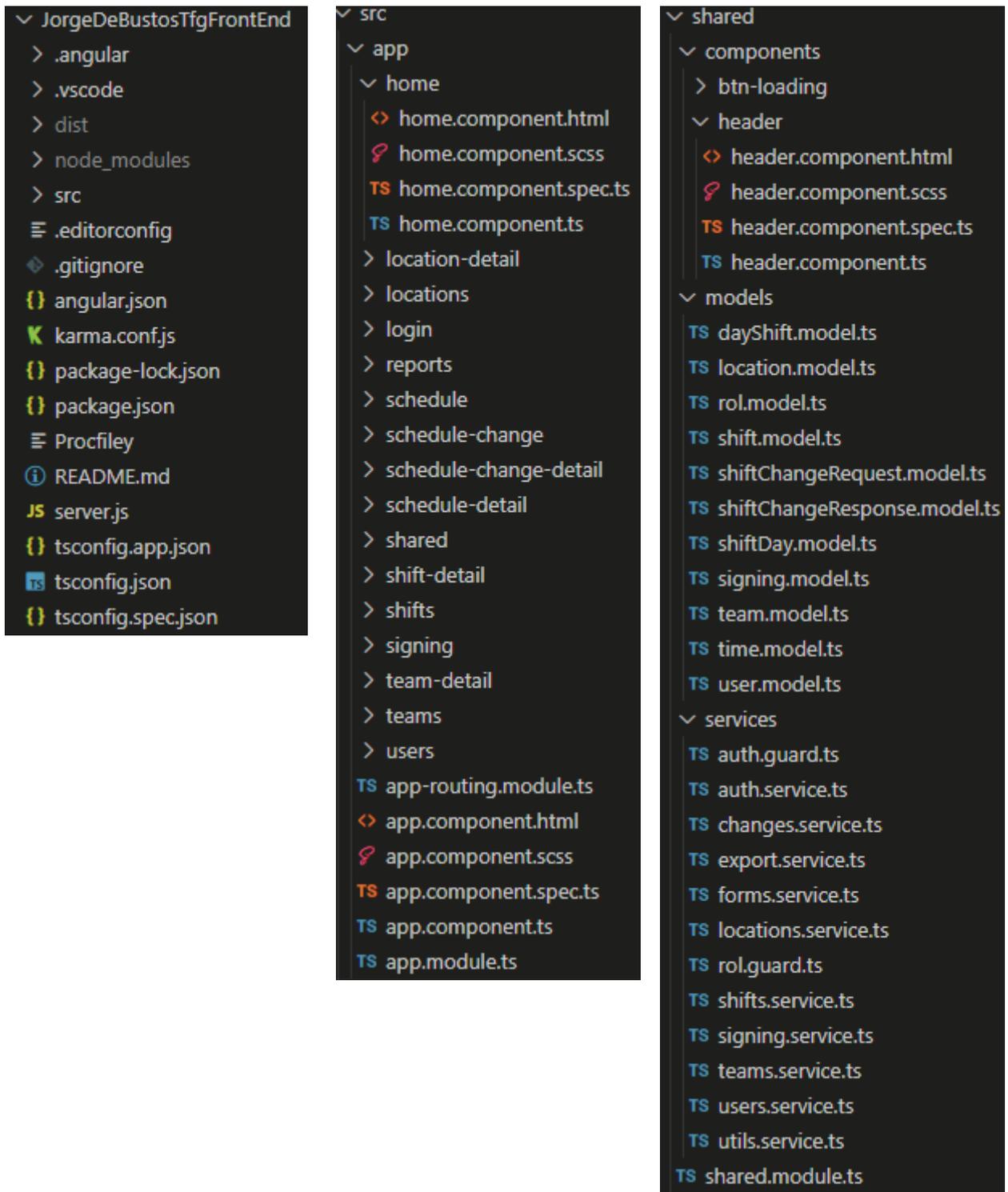


Figura 6.1: Estructura código frontend

Como se presenta en las imágenes la estructura inicial del directorio es la proporcionada por Node al crear un proyecto Angular, generando archivos de configuración y estructuras predefinidas, como los archivos angular.json, package.json, tsconfig.json etc.

Por otro lado, el código fuente está ubicado en la carpeta src/app, donde tenemos los diferentes componentes que conforman la aplicación con la estructura indicada. Cabe mencionar que los

componentes no desplegados, como location, location-detail, users, teams, team-detail, shift, shift-detail etc. no se han desplegado pues su estructura es la misma que la del componente home (template HTML, hoja de estilos SCSS y el archivo TypeScript).

El directorio shared sin embargo, tiene su propia estructura, ya que aquí se aloja cualquier recurso común en la aplicación; componentes, modelos y servicios.

6.2. Estructura del código backend

A continuación, se muestra unas imágenes que describe la estructura final del código que presenta el backend.

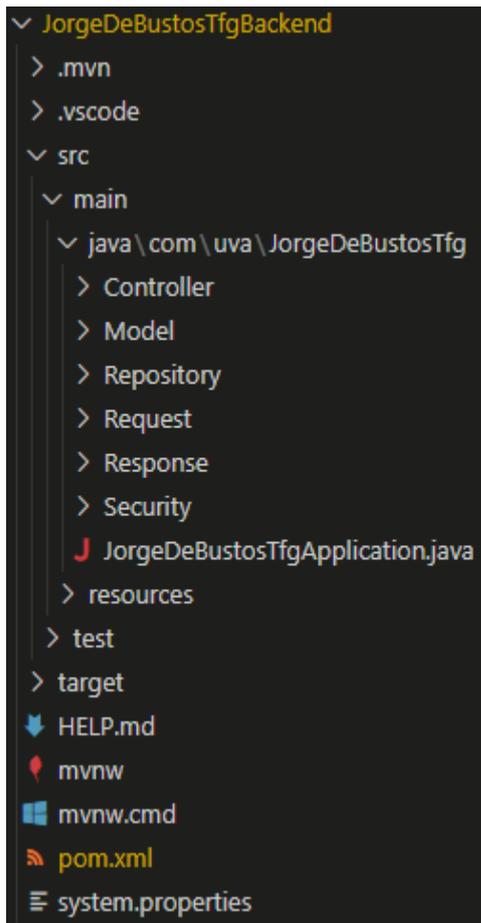


Figura 6.2: Estructura código backend (General, Controllers, Entities, Repository)

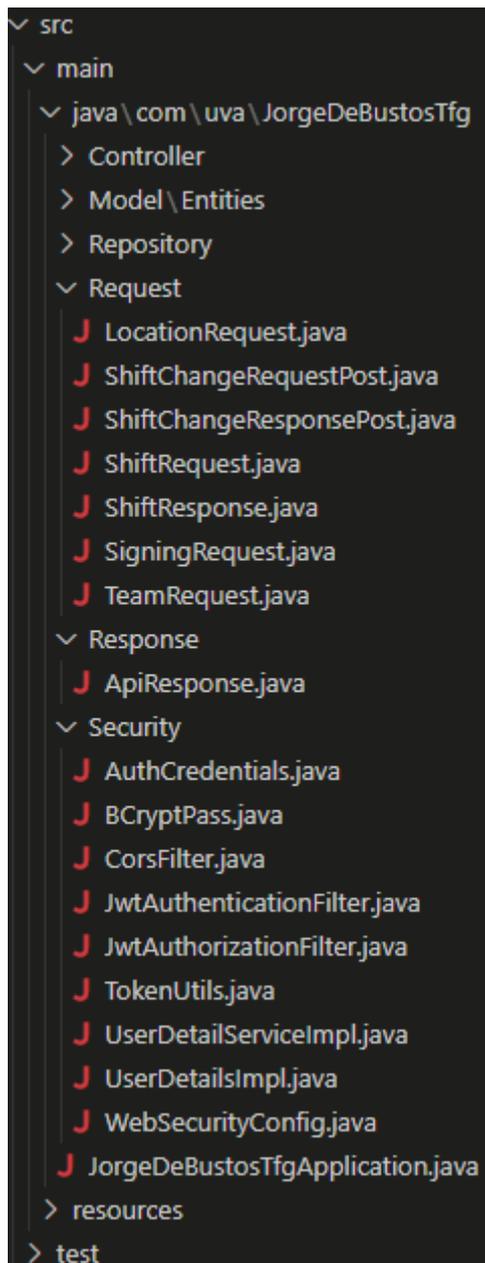


Figura 6.3: Estructura código backend (Requests, Response, Security)

El directorio general del proyecto se aprecia reside en la carpeta `src/main/java/com/uva/JorgeDeBustosTfg`, aquí se ubican las clases correspondientes a los Controller, Repository, Entities, Requests, Response y Security que especificaron en el apartado 4.4.

6.3. Pruebas de caja negra

A la hora de validar el correcto funcionamiento de un proyecto software es necesario implementar una serie de pruebas con el fin de evitar errores no contemplados en la fase de desarrollo.

Es por esto por lo que en un proyecto software se implementan pruebas de dos tipos:

- **Pruebas de caja blanca:** Estas pruebas se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Se escogen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados [32].
- **Pruebas de caja negra:** Las pruebas de caja negra pueden definirse como una técnica donde se busca la verificación de las funcionalidades del software, es decir, no se toma como referente la estructura del código interno [33].

En este proyecto nos centraremos en las pruebas de caja negra guiadas por casos de uso, es decir, para cada caso de uso definido estableceremos diferentes flujos y situaciones junto con un comportamiento esperado y una confirmación posterior de si este flujo se ha realizado con éxito.

Las pruebas PCN-006, PCN-007 y PCN-008 recogen los mismos escenarios que las relacionadas con Ubicaciones y Turnos y puesto que sus resultados finales de estas dos han sido correctos se consideran que sus pruebas de caja negra quedan ya reflejadas.

PCN-001		
Iniciar Sesión		
Entrada	Resultado Esperado	Resultado Final
El usuario ya registrado en la plataforma introduce sus datos de acceso (email y contraseña) y pulsa el botón de inicio de sesión	El usuario es redirigido a la página de Inicio con una sesión válida de 4 horas	Correcto
El usuario sin estar previamente registrado introduce los datos de acceso y pulsa el botón de inicio de sesión	El sistema informa de que las credenciales aportadas son erróneas	Correcto
El usuario ya registrado en la plataforma introduce datos de acceso incorrectos y pulsa el botón de inicio de sesión	El sistema informa de que las credenciales aportadas son erróneas	Correcto
El usuario ya registrado en la plataforma, pero inhabilitado, introduce sus datos de acceso y pulsa el botón de inicio de sesión	El sistema comprueba que el usuario está inhabilitado y no permite el acceso	Correcto
El usuario pulsa el botón de inicio de sesión sin aportar datos	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto

Tabla 6.1: PCN-001 Iniciar Sesión

PCN-002	Cerrar Sesión	
Entrada	Resultado Esperado	Resultado Final
El usuario ya identificado en la plataforma selecciona el botón de cierre de sesión	El usuario es redirigido a la página de Login sin sesión válida	Correcto

Tabla 6.2: PCN-002 Cerrar Sesión

PCN-003	Editar Perfil	
Entrada	Resultado Esperado	Resultado Final
El usuario selecciona su información personal de la cabecera de la aplicación, cambia los datos deseados y pulsa en editar usuario	El sistema actualiza la información e informa al usuario	Correcto
El usuario selecciona su información personal de la cabecera de la aplicación, vacía campos obligatorios y pulsa en editar usuario	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto

Tabla 6.3: PCN-003 Editar Perfil

PCN-004	Importar usuarios	
Entrada	Resultado Esperado	Resultado Final
El responsable importa en la sección Importar Usuarios una plantilla válida, con datos correctos y pulsa el botón importar usuarios	El sistema registra a los nuevos usuarios e informa al responsable	Correcto
El responsable importa en la sección Importar Usuarios una plantilla incorrecta	El sistema indica los errores e inhabilita la importación	Correcto
El responsable importa en la sección Importar Usuarios una plantilla correcta pero con datos errneos	El sistema indica los errores e inhabilita la exportación	Correcto

Tabla 6.4: PCN-004 Importar usuarios

PCN-005		Inhabilitar perfil	
Entrada	Resultado Esperado	Resultado Final	
El responsable selecciona en la tarjeta del usuario habilitado que desee inhabilitar	El sistema inhabilita al usuario e informa al responsable	Correcto	
El responsable selecciona en la tarjeta del usuario inhabilitado que desee habilitar	El sistema habilita al usuario e informa al responsable	Correcto	

Tabla 6.5: PCN-005 Inhabilitar perfil

PCN-006		Registrar Equipo	
Entrada	Resultado Esperado	Resultado Final	
El responsable selecciona la opción de Nuevo Equipo y completa el formulario con los campos y asignaciones requeridas y pulsa en Crear Equipo	El sistema registra el nuevo equipo y redirige a la pestaña de Equipos	Correcto	
El responsable selecciona la opción de Nuevo Equipo y no completa el formulario con los campos y asignaciones requeridas y pulsa en Crear Equipo	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto	

Tabla 6.6: PCN-006 Registrar Equipo

PCN-007	Editar Equipo	
Entrada	Resultado Esperado	Resultado Final
El responsable selecciona la opción de Editar Equipo sobre un equipo ya existente, modifica la información del equipo y pulsa en Editar equipo	El sistema actualiza los datos del equipo existente y redirige a la pestaña de Equipos	Correcto
El responsable selecciona la opción de Editar Equipo sobre un equipo ya existente, modifica la información del equipo dejando algún campo vacío o incorrecto y pulsa en Editar equipo	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto

Tabla 6.7: PCN-007 Editar Equipo

PCN-008	Eliminar Equipo	
Entrada	Resultado Esperado	Resultado Final
El responsable selecciona la opción de Eliminar Equipo sobre un equipo ya existente y pulsa sobre el botón Eliminar	El sistema elimina los datos del equipo y sus asociaciones y redirige a la pestaña de Equipos	Correcto

Tabla 6.8: PCN-008 Eliminar Equipo

PCN-009		Realizar Fichaje	
Entrada	Resultado Esperado	Resultado Final	
El usuario selecciona la opción de Fichaje, con equipo y ubicación preasignado y registra una entrada	El sistema registra la entrada de ese fichaje e informa al usuario	Correcto	
El usuario selecciona la opción de Fichaje, con equipo y ubicación preasignado y registra una salida de un fichaje ya existente	El sistema registra la salida del fichaje sin cerrar e informa al usuario	Correcto	
El usuario selecciona la opción de Fichaje, sin equipo y ubicación preasignado y registra una entrada	El sistema informa que el fichaje se realiza sin equipo ni ubicación asignada, registra la entrada del fichaje e informa al usuario	Correcto	

Tabla 6.9: PCN-009 Registrar Fichaje

PCN-010		Solicitar Cambio de turno	
Entrada	Resultado Esperado	Resultado Final	
El usuario con un turno preasignado selecciona la opción de Nuevo cambio en la sección de cambio de horario, selecciona los días deseados y pulsa sobre el botón Registrar cambio	El sistema registra la solicitud de cambio de jornada y redirige a la pestaña de cambios de horario	Correcto	
El usuario con un turno preasignado selecciona la opción de Nuevo cambio en la sección de cambio de horario, no selecciona ningún día y pulsa sobre el botón Registrar cambio	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto	
El usuario sin un turno preasignado selecciona la opción de Nuevo cambio en la sección de cambio de horario y pulsa sobre el botón Registrar cambio	El sistema indica que no tiene ningún turno asignado y que campos del formulario solicitado son obligatorios o incorrectos	Correcto	

Tabla 6.10: PCN-010 Solicitar Cambio de turno

PCN-011		Aceptar solicitud de cambio de turno
Entrada	Resultado Esperado	Resultado Final
El usuario con un turno preasignado selecciona la opción Ver de una solicitud disponible, selecciona la misma cantidad de días que se solicitan en la petición y pulsa sobre Aceptar Cambio	El sistema registra la respuesta de la solicitud de cambio de jornada y redirige a la pestaña de cambios de horario	Correcto
El usuario con un turno preasignado selecciona la opción Ver de una solicitud disponible, no selecciona la misma cantidad de días que se solicitan en la petición y pulsa sobre Aceptar Cambio	El sistema indica que campos del formulario solicitado son obligatorios o incorrectos	Correcto
El usuario sin un turno preasignado selecciona la opción Ver de una solicitud disponible, selecciona la misma cantidad de días que se solicitan en la petición y pulsa sobre Aceptar Cambio	El sistema indica que no tiene ningún turno asignado y que campos del formulario solicitado son obligatorios o incorrectos	Correcto

Tabla 6.11: PCN-011 Aceptar solicitud de cambio de turno

PCN-012		Aceptar respuesta de cambio de turno
Entrada	Resultado Esperado	Resultado Final
El usuario con un turno preasignado selecciona la opción Ver de una de sus solicitudes de cambio y acepta el cambio	El sistema efectúa el cambio de los días involucrados entre los usuarios y redirige a la página del horario del usuario	Correcto
El usuario con un turno preasignado selecciona la opción Ver de una de sus solicitudes de cambio y rechaza el cambio	El sistema registra el rechazo a la respuesta de cambio de turno, establece la solicitud de nuevo como pendiente y redirige a la página de Cambios	Correcto

Tabla 6.12: PCN-012 Aceptar respuesta de cambio de turno

PCN-013	Obtención de informes	
Entrada	Resultado Esperado	Resultado Final
El responsable selecciona la opción de Informes, visualiza y descarga los registros de los fichajes realizados	El sistema muestra y descarga los registros de los fichajes	Correcto

Tabla 6.13: PCN-013 Obtención de informes

6.4. Evaluación de usabilidad

A lo largo del proyecto, y como se ha explicado en el apartado 1.4, LaborFlix ha perseguido la usabilidad en la interfaz de usuario para así conseguir una curva de aprendizaje mínima, por lo que, para finalizar con las pruebas de la plataforma, se han elaborado unas pruebas de usabilidad a 2 usuarios que asumirán tanto el rol de responsable como el de trabajador.

Antes de presentar dichas pruebas a los usuarios, se ha hecho brevemente una introducción a la plataforma explicando la funcionalidad que esta ofrece y se ha poblado la base de datos con un solo usuario para el acceso.

Las pruebas a realizar para los distintos roles han sido las siguientes.

6.4.1. Responsable

Para el rol responsable se han definido las siguientes pruebas de usabilidad.

- **Prueba 1:** *Asignar un equipo llamado “Obreros” a los dos nuevos usuarios que ha de introducir en la plataforma.*
- **Prueba 2:** *Asignar una ubicación en la Calle Hijas de la Caridad, Palencia, con un radio de 50 metros al equipo “Obreros”.*
- **Prueba 3:** *Asignar un turno “Horario de Julio” con los días; 24 de julio (08:00 – 15:00) y 25 de julio (08:00 – 15:00) a uno de los usuarios previamente creados.*
- **Prueba 4:** *Asignar un turno “Horario de Agosto” con los días; 14 de agosto (08:00 – 15:00) y 15 de agosto (08:00 – 15:00) al usuario que no tiene asignado “Horario de Julio”.*

Prueba	Usuario 1	Usuario 2
-	Rango de edad: 22 – 33 Experiencia TICs: 7/10	Rango de edad: 40 – 51 Experiencia TICs: 5/10
Prueba 1	Completada en 1 min y 5 s	Completada en 2 min y 21 s
Prueba 2	Completada en 0 min y 32 s	Completada en 0 min y 57 s
Prueba 3	Completada en 0 min y 44 s	Completada en 1 min y 11 s
Prueba 4	Completada en 0 min y 35 s	Completada en 0 min y 56 s

Tabla 6.14: Pruebas de usabilidad para el rol responsable

De los tiempos de ejecución de las pruebas y del *feedback* obtenido por parte de los usuarios 1 y 2, se han concluido los siguientes puntos:

- La importación de usuarios es un proceso que generaba dudas en cuanto al formato que debían seguir los campos introducidos en el Excel, lo cual provocaba que algunos datos fuesen incorrectos, con el consiguiente error informado “No se ha podido introducir los usuarios “X”, “Y” etc.”.

Analizado este comportamiento se optó por; introducir dos ejemplos de usuarios en la plantilla Excel para escenificar el formato correcto de los datos y validar los campos previamente a su inserción, indicando así la validez fila por fila del contenido del archivo.

- Tanto la creación y asignación de equipos, ubicaciones y turnos son procesos autodescriptivos que permiten su ejecución en tiempos razonables (< 1 min 30 s).

6.4.2. Trabajador

Para el rol trabajador se han definido las siguientes pruebas de usabilidad:

- **Prueba 1:** Realizar la entrada de un fichaje, a continuación, navegar entre 3 páginas diferentes antes de cerrar el fichaje.
- **Prueba 2:** Registrar una solicitud de cambio de jornada laboral, solicitando solamente un día de tu turno laboral.
- **Prueba 3:** Aceptar una solicitud de cambio de otro usuario.
- **Prueba 4:** Aceptar la respuesta a tu solicitud registrada anteriormente.

Prueba	Usuario 1	Usuario 2
-	Rango de edad: 22 – 33 Experiencia TICs: 7/10	Rango de edad: 40 – 51 Experiencia TICs: 5/10
Prueba 1	Completada en 0 min y 18 s	Completada en 0 min y 26 s
Prueba 2	Completada en 0 min y 25 s	Completada en 0 min y 49 s
Prueba 3	Completada en 0 min y 18 s	Completada en 0 min y 41 s
Prueba 4	Completada en 0 min y 16 s	Completada en 0 min y 18 s

Tabla 6.15: Pruebas de usabilidad para el rol trabajador

De los tiempos de ejecución de las pruebas y del *feedback* obtenido por parte de los usuarios 1 y 2, se han concluido los siguientes puntos:

- El sistema de fichaje es altamente descriptivo indicando visualmente la localización del usuario y de la ubicación asignada
- La incorporación de migas de pan facilitó en gran medida la navegación rápida de los usuarios entre páginas, puesto que la utilizaban con mayor frecuencia que el botón de retroceso integrado de los navegadores.
- A pesar de los bajos tiempos de ejecución en la parte de cambios de jornada, se apreció como los usuarios navegaban innecesariamente entre las páginas de sus solicitudes de cambio y las de otros usuarios, por esto, se decidió unificar ambas en una misma página, consiguiendo así menor navegación entre dos componentes muy relacionados.

Para concluir, se puede afirmar que el diseño de la plataforma cumple con el objetivo de usabilidad y de facilitar una curva de aprendizaje mínima, puesto que los usuarios afirman comprender rápidamente el funcionamiento de esta gracias a la uniformidad de colores, distribución parecida entre componentes de distintas páginas y similitud en la realización de los procesos.

Capítulo 7

7. Seguimiento del proyecto

Como se indicó en la planificación ideal (Sección 2.2) el proyecto comenzó el 17/02/2023, definiendo las historias de usuario, la planificación inicial y el análisis de riesgos.

Una vez completadas estas secciones, se continuó elaborando el análisis de requisitos, casos de uso y modelo de dominio en análisis, lo cual conformaba una base suficientemente sólida para comenzar con el desarrollo de la aplicación.

En el desarrollo se comenzó instanciando una base de datos MariaDB de forma local, conformando un proyecto Spring Boot desde cero y definiendo en el archivo pom.xml todas aquellas dependencias y configuraciones necesarias para que el proyecto contase con las librerías que requeriría más adelante. Una vez estos pasos se finalizaron, se creó el sistema de login, autenticación y autorización de la parte de seguridad de la Api junto con el modelo de Usuario, concluyendo así en un sistema de autenticación que impedía realizar peticiones a cualquier endpoint de la plataforma sin aportar un token JWT válido resultante de un proceso de “login” correcto.

A partir de este punto, el desarrollo se basó en realizar uno por uno el análisis, diseño, implementación y pruebas de cada caso de uso expuesto a largo de este documento. Una vez este proceso acabó se completó la memoria de trabajo finalizando los apartados de introducción, planificación, diseño, las tecnologías utilizadas y conclusiones.

Cabe destacar dos puntos importantes ocurridos a lo largo del desarrollo del software.

En primer lugar, la aparición de algunos riesgos que se habían previsto en el análisis de riesgos, como errores en las estimaciones e interpretación errónea de los requisitos, afectaron a la duración del desarrollo, prolongándolo más de lo estimado y provocando que no se cumplieren los plazos de tiempo estipulados para la finalización del proyecto.

Estos sucesos produjeron un punto relevante a mencionar, la supresión de un objetivo y por lo tanto de los requisitos relacionados con este objetivo. En nuestro caso, el objetivo incumplido fue el relacionado con la funcionalidad de los permisos laborales, y por extensión, los requisitos funcionales relacionados con este objetivo.

Para finalizar, la cantidad de horas de trabajo final ha excedido la planificación inicial de 300 horas, obteniendo un total aproximado de **354 horas**.

Capítulo 8

8. Conclusiones

En este apartado se expondrán las conclusiones obtenidas del desarrollo del proyecto junto con posibles líneas de trabajo futuras.

8.1. Conclusiones

Tras finalizar el proyecto la retrospectiva que se obtiene de este es que ha sido un proceso largo que ha supuesto muchas horas de trabajo, las cuales se veían algo reducidas al tener que compaginarlas con el trabajo, sin embargo, el sentimiento que más se ha experimentado es el de satisfacción y de alegría al haber podido realizar íntegramente el desarrollo de un proyecto software desde su fase inicial hasta la final.

A estos sentimientos se le suman el aprendizaje adquirido, si bien sobre el framework de la parte frontend, Angular, ya se tenía conocimientos previos gracias a asignaturas, prácticas y trabajo, había un poco más de incertidumbre en la tecnología de backend, ya que Spring Boot es un framework con el que no se había interactuado demasiado.

En cuanto a los objetivos de desarrollo del trabajo fin de grado alcanzados se han conseguido:

- Elaboración, gestión y personalización de equipos de trabajos asociados a una ubicación física.
- Elaboración y personalización de turnos laborales para diferentes trabajadores.
- Realización de fichajes laborales geolocalizados en base a la ubicación previamente asignada.
- Posibilidad de visualización del horario laboral individual previamente asignado.
- Posibilidad de intercambio de días laborales incluidos en el horario con otro usuario de la plataforma.
- Obtención de informes de los fichajes realizados.

8.2. Líneas de trabajo futuras

Hay algunas mejoras propias del aplicativo y diferentes apartados de la gestión de recursos humanos que la aplicación podría albergar para así afianzar más la utilización de un solo software RRHH centralizado en las empresas, como, por ejemplo:

- Sistema de permisos laborales.

- Sistema de notificaciones a usuarios implantado en el sistema de cambio de jornadas y permisos laborales.
- Sistema de gestión de nóminas e informes propios del personal.
- Sistema de registro de laborales realizadas durante la jornada laboral.
- Mejora de la resolución de la plataforma en dispositivos móviles.
- Implementación de un sistema de recuperación y/o restauración de contraseña.

Bibliografía

- [1] Your company uses too many HR systems (and that costs you big). (2019, febrero 18). Paycom.com. <https://www.paycom.com/resources/blog/your-company-uses-too-many-hr-systems-and-that-costs-you-big/>
- [2] Zoho People. (2016, febrero 23). Conpas; Conpas.net. <https://www.conpas.net/zoho-people.html>
- [3] ¿Qué es ADP Workforce Now?. 411answers.com. Recuperado el 9 de julio de 2023, de <https://es.411answers.com/a/que-es-adp-workforce-now.html>
- [4] Qué es SCRUM. Arimetrics. Recuperado el 9 de julio de 2023, de <https://www.arimetrics.com/glosario-digital/scrum>
- [5] Desarrollo iterativo e incremental. (2008, septiembre 27). Proyectos Ágiles. <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- [6] Artefactos Scrum: las 3 herramientas clave de gestión. (2017, diciembre 5). Deloitte Spain. <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>
- [7] Qué son los Eventos en Scrum. (2021, mayo 26). DoneTonic. <https://donetonic.com/es/que-son-los-sprints-en-scrum/>
- [8] Scrum: roles y responsabilidades. (2017, noviembre 29). Deloitte Spain. <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>
- [9] Clasificación, D. y. Gestión y supervisión de riesgos. Lsi.us.es. Recuperado el 9 de julio de 2023, de <http://www.lsi.us.es/docencia/get.php?id=7133>
- [10] Riesgo del software. Webcindario.com. Recuperado el 9 de julio de 2023, de <https://ingenieriasoft.webcindario.com/gestion-y-planificacion-de-proyectos/planificacion-de-proyectos-de-software/riesgo-del-software.html>
- [11] Leon, I. F., Liliana, M., & Jimenez, A. GESTION DE RIESGOS EN PROYECTO DE SOFTWARE A DESARROLLAR EN EMPRESA PRIVADA DIRECTOR DE TRABAJO DE GRADO. Core.ac.uk. Recuperado el 9 de julio de 2023, de <https://core.ac.uk/download/pdf/143451627.pdf>
- [12] Salario para Programador Web en España - Salario Medio. Talent.com. Recuperado el 9 de julio de 2023, de <https://es.talent.com/salary?job=programador+web>
- [13] Asana. (2022, enero 21). Historias de usuario: 3 ejemplos para generar valor para el usuario. Asana. <https://asana.com/es/resources/user-stories>

- [14] ¿Qué es el patrón de arquitectura MVVM? (2022, septiembre 29).KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/>
- [15] Zuev, A. (2020, junio 5). Patrón MVVM en SwiftUI. Adictos al trabajo. <https://www.adictosaltrabajo.com/2020/06/05/patron-mvvm-en-swiftui/>
- [16] DAO [cursohibernate]. Cursohibernate.es. Recuperado el 9 de julio de 2023, de <http://www.cursohibernate.es/doku.php?id=patrones:dao>
- [17] Data Transfer Object (DTO) – Patrón de diseño. (2018, noviembre 30). Oscar Blancarte - Software Architecture. <https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/>
- [18] Janssen, T. (2019, enero 24) Implementing the Repository pattern with JPA and Hibernate. Thorben Janssen. <https://thorben-janssen.com/implementing-the-repository-pattern-with-jpa-and-hibernate/>
- [19] Terreros, J. C. Desarrollo de Software Basado en Componentes. Elguille.info. Recuperado el 9 de julio de 2023, de https://www.elguille.info/colabora/NET2005/julio_casal_Componentes.htm
- [20] Angular. Angular.Io. Recuperado el 9 de julio de 2023, de <https://angular.io/guide/architecture-components>
- [21] Privacidad por diseño (Privacy by Design) y por defecto. (2019, mayo 22). Asesorías. <https://asesorias.com/empresas/normativas/proteccion-datos/privacy-design/>
- [22] Rouse, M. (2012, enero 16). Microsoft Office 365. Techopedia. <https://www.techopedia.com/definition/26678/microsoft-office-365>
- [23] Flores, F. (2022, julio 22). Qué es Visual Studio Code y qué ventajas ofrece.Openwebinars.net. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- [24] Astah* professional. Software - O software que você precisa está aqui! Entre já! Recuperado el 9 de julio de 2023, de <https://astah.net/support/astah-pro/>
- [25] ¿Qué es Git?. Microsoft.com. Recuperado el 9 de julio de 2023, de <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>
- [26] Blueelephant, /. (2018, agosto 8).¿Cómo trabajar en equipo con Git?El blog del elefante azul. <https://blueelephant.wordpress.com/2018/08/08/como-trabajar-en-equipo-con-git/>
- [27] Coppola, M. (2022, junio 20).¿Qué es Angular? Características y ventajas. Hubspot.es. <https://blog.hubspot.es/website/que-es-angular>
- [28] Gonzalez, M. V. (2021, marzo 17) ¿Qué es Spring Boot? Codmind. <https://blog.codmind.com/que-es-spring-boot/>

- [29] Magaña, L. M. L. (2020, enero 17). Qué es Json Web Token y cómo funciona. Openwebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- [30] Cantu, M. (2020, octubre 29). HeidiSQL es una herramienta ligera de gestión de bases de datos de código abierto integrada en Delphi. Embarcadero RAD Studio, Delphi, & C++Builder Blogs. <https://blogs.embarcadero.com/es/heidisql-is-a-lightweight-open-source-database-management-tool-built-in-delphi/>
- [31] Qué es MariaDB y cuáles son sus características. Com.es. <https://www.hostingplus.com.es/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>
- [32] Myers, Glenford J., The art of software testing, John Wiley & Sons 2004 ISBN: 0471469122
- [33] Binder, Robert V. Testing object-oriented systems: models, patterns, and tolos, Addison-Wesley, 2000
- [34] Documentation. Heroku.com. Recuperado el 9 de julio de 2023, de <https://devcenter.heroku.com/categories/reference/>

Anexos

I. Manual de instalación

En esta sección del apartado de Anexos se procederá a explicar los pasos seguidos para el despliegue de la plataforma en su conjunto, tanto de la parte backend, frontend y base de datos.

El despliegue se ha realizado sobre el servicio en la nube Heroku [34], el cual está basado en la creación de aplicaciones donde gracias a su integración con Git se pueden desplegar proyectos de distintos tipos de lenguaje.

Para el alojamiento de LaborFlix en Heroku se han creado dos aplicaciones, “jorge-de-bustos-backend” para alojar el proyecto Spring Boot de la parte backend y “jorge-de-bustos-frontend”, para alojar la parte frontend del aplicativo desarrollado en Angular.

I.I. Manual de despliegue de backend

Para el despliegue del backend se ha creado una aplicación Heroku llamada “jorge-de-bustos-backend” que proporcionaba un enlace de integración Git, y dado que mi repositorio local ya estaba conectado con un repositorio alojado en el GitLab de la propia escuela de Ingeniería Informática, la solución pasaba por añadir un repositorio en la nube a este proyecto mediante el comando “`git remote add heroku git@heroku.com:project.git`”.

Una vez añadido el código a un repositorio git faltaba conectarlo con la propia aplicación creada previamente en mi entorno Heroku mediante el comando “`heroku git:remote -a jorge-de-bustos-backend`”, consiguiendo así el poder desplegar el proyecto en la aplicación.

Sin embargo, para que mi proyecto Spring Boot funcionase correctamente en este entorno eran necesario dos pasos previos.

- La creación de una base de datos MariaDB en dicha aplicación remota mediante la tienda de add-ons que proporciona Heroku en su plataforma, para así alojar la parte de persistencia del proyecto.

Posteriormente había que modificar el archivo *application.properties* de Spring Boot para referenciar a esta base de datos.

- La creación de un archivo *system.properties* en el código backend para así definir la versión requerida de Java, en mi caso, Java17.

Una vez finalizados estos dos procesos el comando “`git push heroku main`” desplegaba el contenido de la rama main del repositorio en la aplicación heroku generando un enlace URL que posteriormente se utilizará como URL de destino de las peticiones que genera la parte frontend.

I.II. Manual de despliegue de frontend

La inicialización de la aplicación Heroku “*jorge-de-bustos-frontend*” y del repositorio que este necesita se ha realizado de la misma forma que para la parte backend.

Las peculiaridades para destacar en esta parte fueron, en primer lugar, la sustitución de la URL de destino de las peticiones HTTP por la generada en el despliegue de la parte backend, y en segundo lugar, la creación de un archivo *server.js* que indicase a Heroku como se ha de desplegar y ejecutar este proyecto Node.

```
let express = require('express');

let app = express();

app.use(express.static(__dirname+'./dist/jorge-de-bustos-frontend'));

app.get('/*', function(req, res) {
  res.sendFile(path.join(__dirname + './dist/MY_APP_NAME/index.html'));
});

app.listen(process.env.PORT || 5000);
```

Figura A.1: Código del archivo *server.js* del proyecto Angular

Una vez finalizados estos dos procesos se desplegó la aplicación con el comando “*git push heroku main*”, el cual generó una URL de acceso a la plataforma:

<https://jorge-de-bustos-frontend-4b02b6921b8f.herokuapp.com/>

II. Manual de usuario

A continuación, se definirá el uso de la plataforma utilizando un usuario con perfil Responsable, ya que comparte funcionalidad con el rol Trabajador, y, además, dispone de varias a mayores.

Para ilustrar mejor los procesos disponibles, dividiremos el manual en cuanto a los módulos que este presenta.

II.I. Inicio de sesión

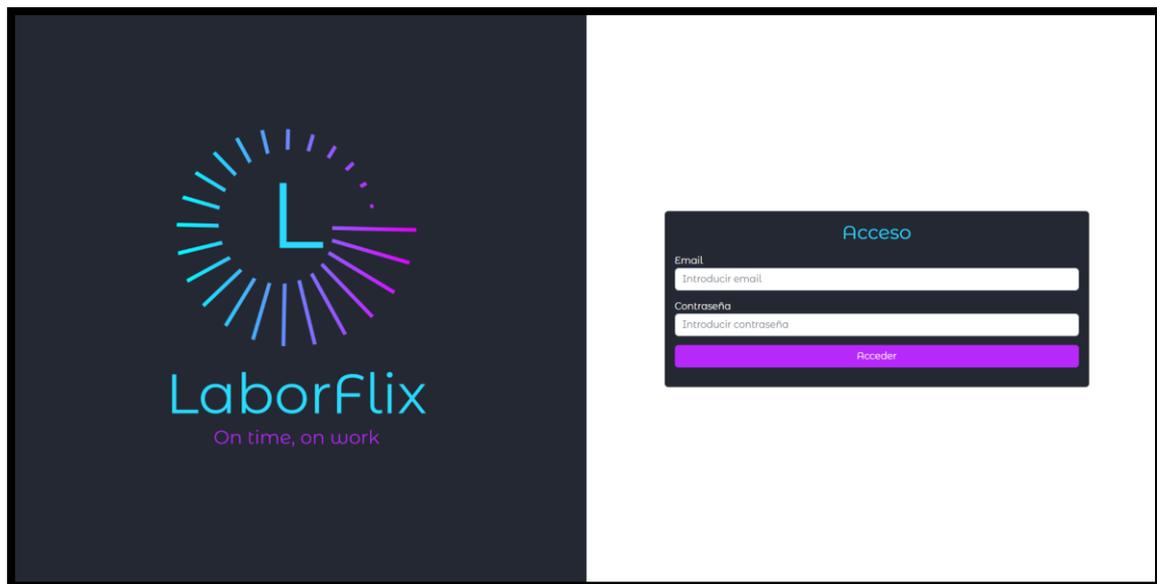


Figura A.2: Pantalla de inicio de sesión

La primera vista con la que nos encontramos es la página de inicio de sesión o acceso a la plataforma, donde deberemos introducir las credenciales de un usuario ya registrado en la plataforma. En este caso utilizaremos las credenciales del administrador que se establece por defecto

- Usuario: admin@laborflix.com
- Contraseña: adminlabor123

Una vez introducidas dichas credenciales seremos redirigidos a la pantalla de Inicio o Escritorio

II.II. Inicio

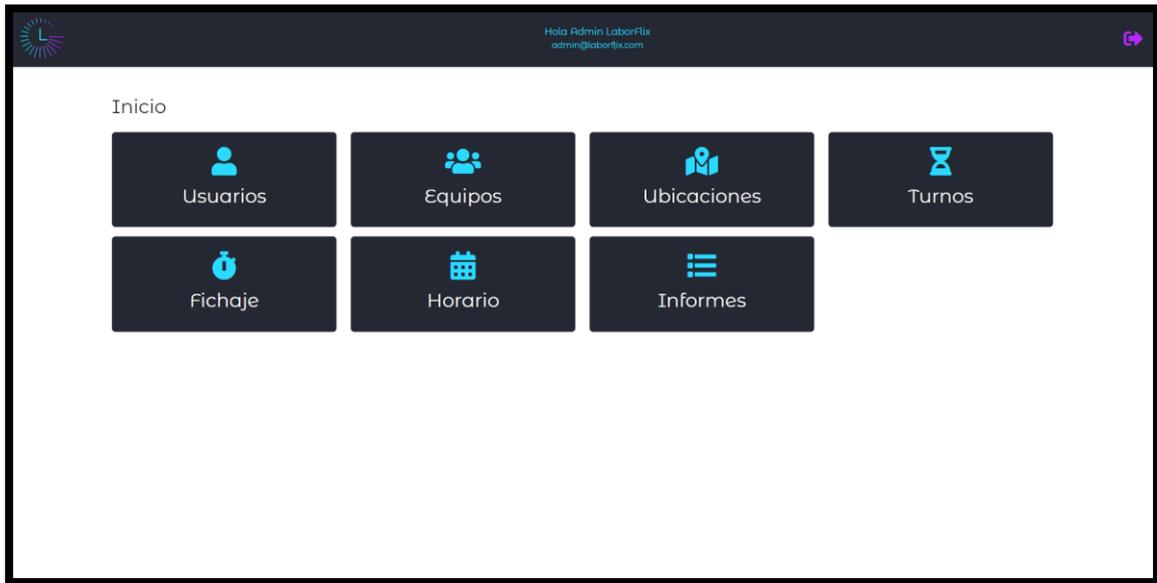


Figura A.3: Página de Inicio

En la pantalla de Inicio caben describir dos componentes, el propio Escritorio y la cabecera (elemento persistente a lo largo de la plataforma).

La cabecera cuenta con tres elementos:

- **Logo LaborFlix:** ubicado en la parte superior izquierda nos permite regresar al Escritorio desde cualquier página solo con pulsarlo.
- **Información del usuario:** ubicada en la parte superior central aparece un mensaje de saludo al usuario autenticado junto con su email. Si hacemos click en esta sección nos aparecerá un pop-up con un formulario de edición de toda la información de dicho usuario.

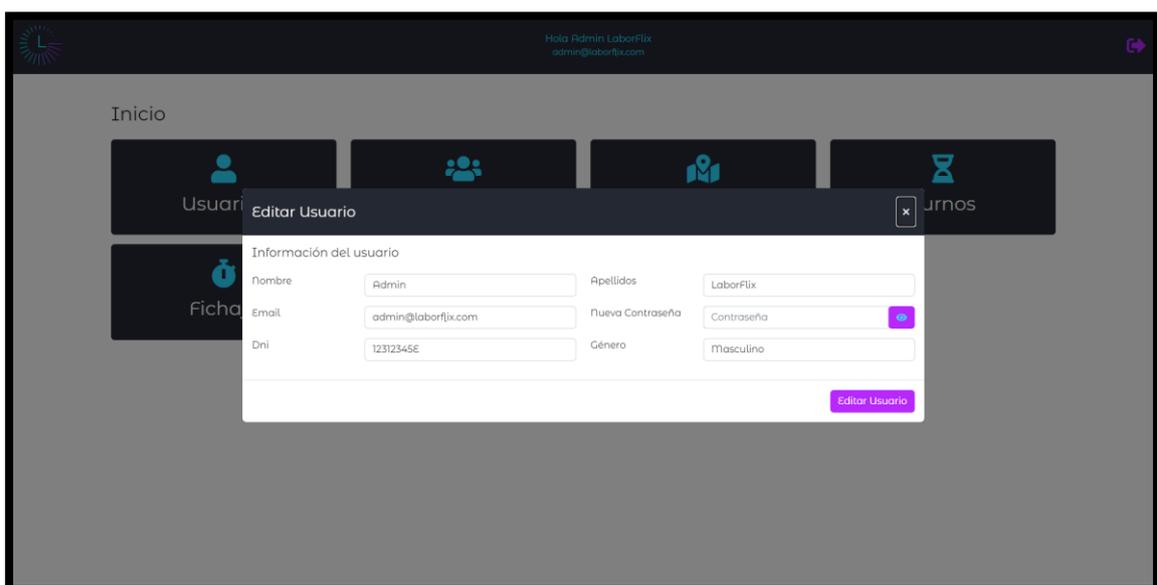


Figura A.4: Pop-up de edición del perfil

- **Cierre de sesión:** ubicado en la parte superior izquierda aparece un icono que nos permite cerrar la sesión actual siempre que queramos, redirigiéndonos a la pantalla de Inicio de Sesión.

Por último, en el propio contenido de la página de Inicio nos permite navegar a todas las páginas de las que se pueden hacer uso.

II.III. Usuarios

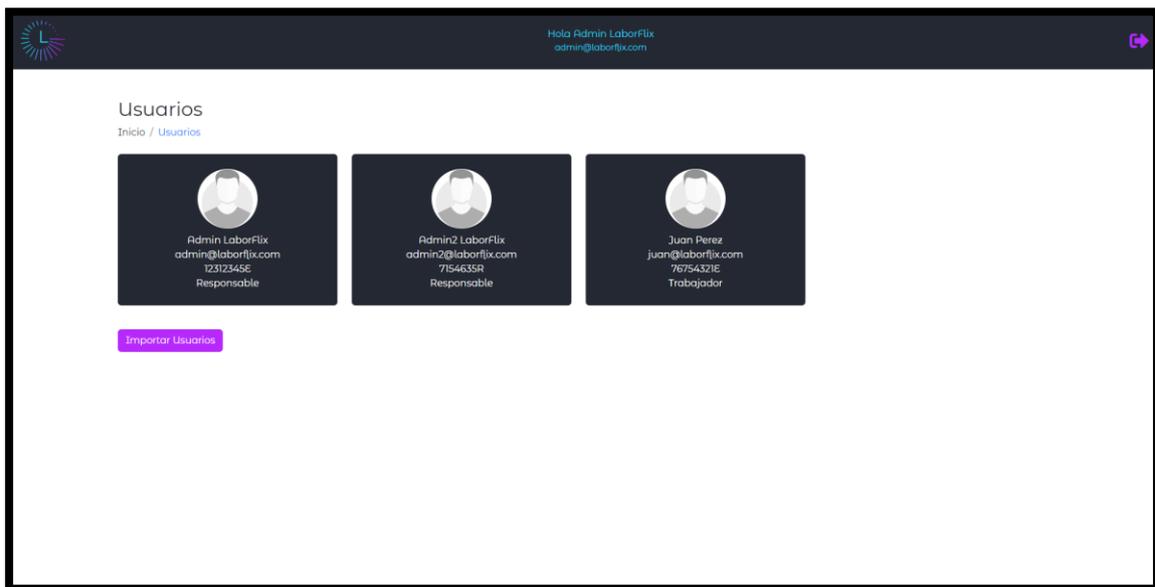


Figura A.5: Página de Usuarios

Al igual que todas las páginas, a excepción de la de Inicio, se puede observar que debajo del título de la página aparece unas migas de pan que no indican en todo momento nuestra ubicación actual dentro de la plataforma.

En la sección de Usuarios podemos obtener información de los usuarios registrados en la plataforma e inhabilitar usuarios para impedirles el acceso a esta haciendo click sobre su tarjeta.

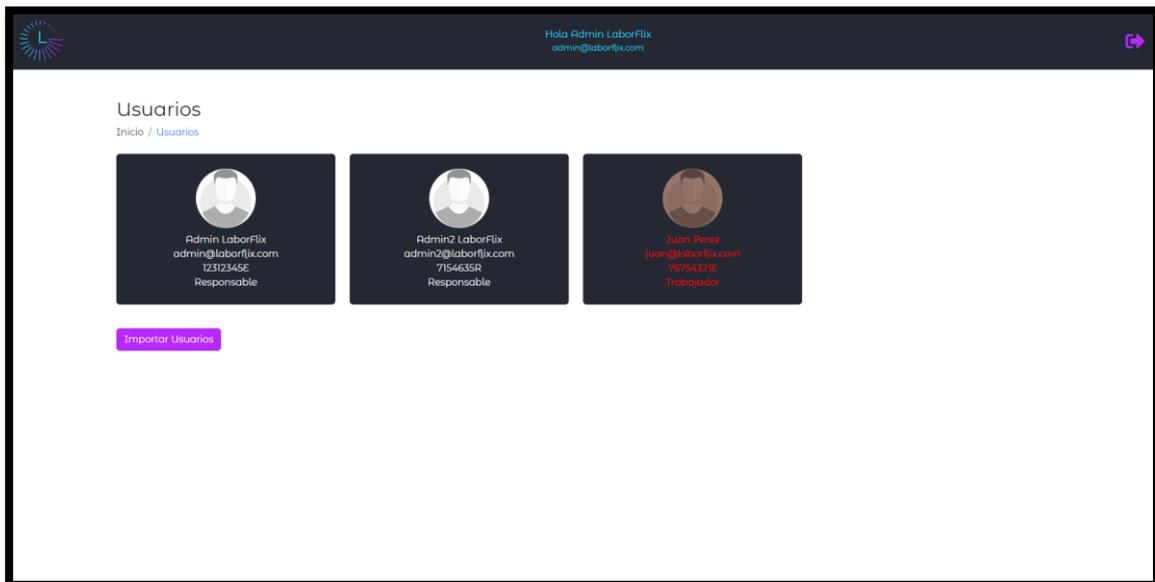


Figura A.6: Página de Usuarios con usuario inhabilitado

Por último, aquí podemos realizar la importación de usuarios a la plataforma seleccionando el botón Importar Usuarios, el cual nos mostrará un pop-up con dicha funcionalidad.

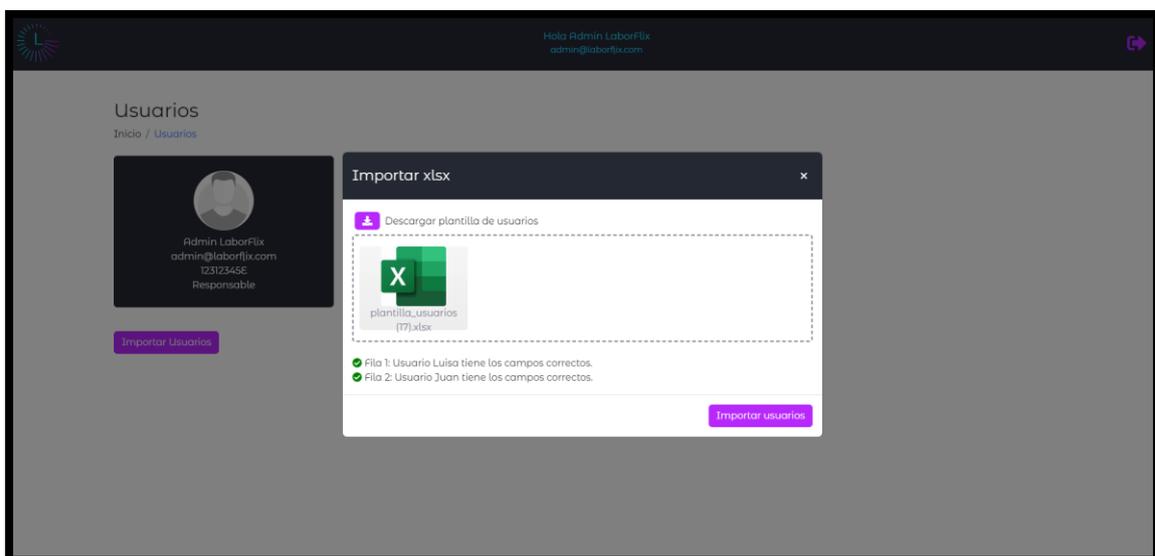


Figura A.7: Importación de usuarios

Como se puede apreciar tenemos dos acciones que realizar, una es descargar la plantilla Excel con la estructura necesaria para hacer de forma correcta la importación y una sección donde bien arrastrar o seleccionar dicho archivo.

Debajo de dicha sección aparecen mensajes que describen la validez línea por línea de los datos introducidos. Si los datos son correctos, se habilita el botón de importación para volcar dichos usuarios a la plataforma.

II.IV. Equipos

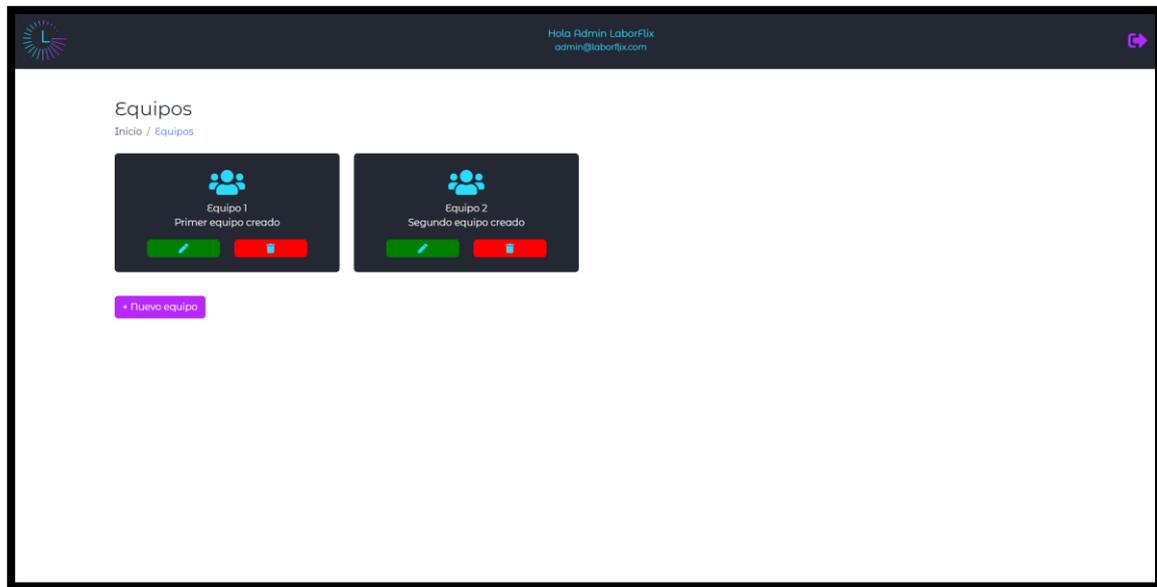


Figura A.8: Página de Equipos

En esta sección se encuentra la gestión y administración de Equipos de trabajo, pudiendo visualizar los equipos creados en la plataforma y permitiendo su edición, borrado y creación de nuevos.

Tanto la creación como la edición comparten la misma página, es decir, se visualiza un formulario donde se introducen los datos y usuarios del equipo.

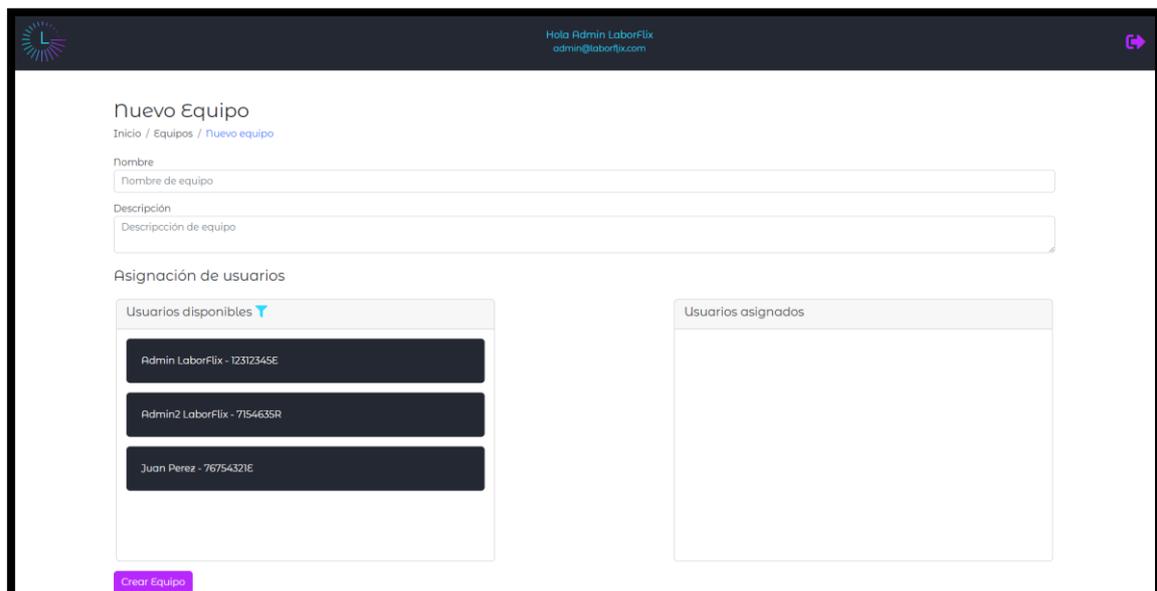


Figura A.9: Página de creación de un equipo

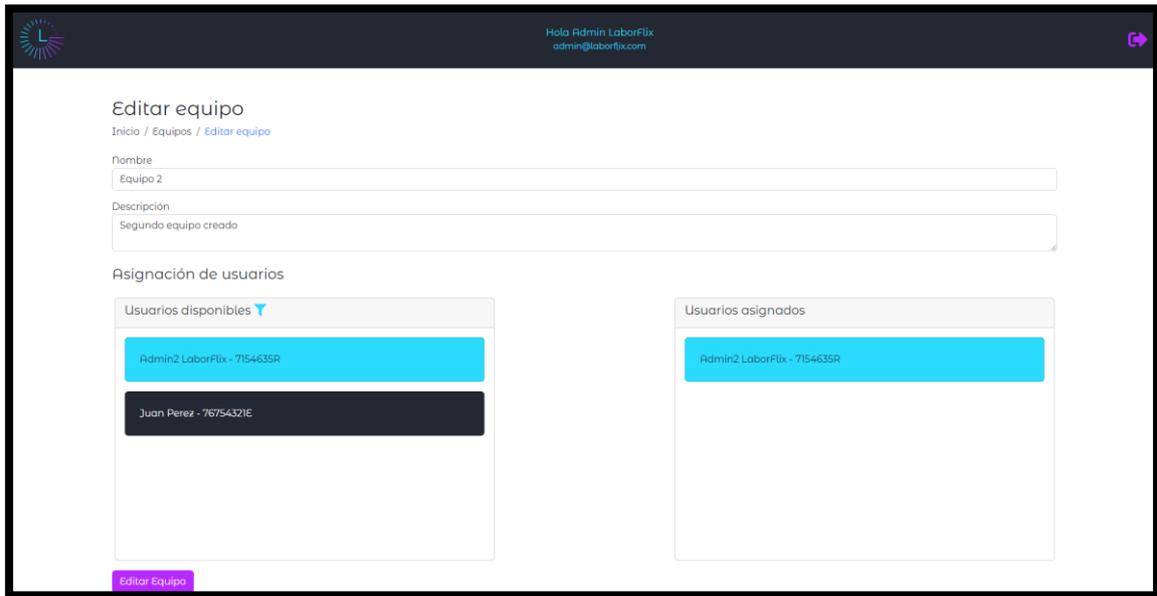


Figura A.10: Página de edición de un equipo

Como se puede apreciar en ambas páginas aparecen campos de la información del propio equipo como unas tarjetas de asignación de usuarios sin equipo asignado.

Al seleccionar la opción Crear Equipo o Editar Equipo la información se crea o actualiza, respectivamente, si los campos son correctos y se redirige a la página de Equipos.

Por último, la eliminación de un equipo nos mostrará un pop-up que nos pedirá confirmación para hacer efectivo el borrado.

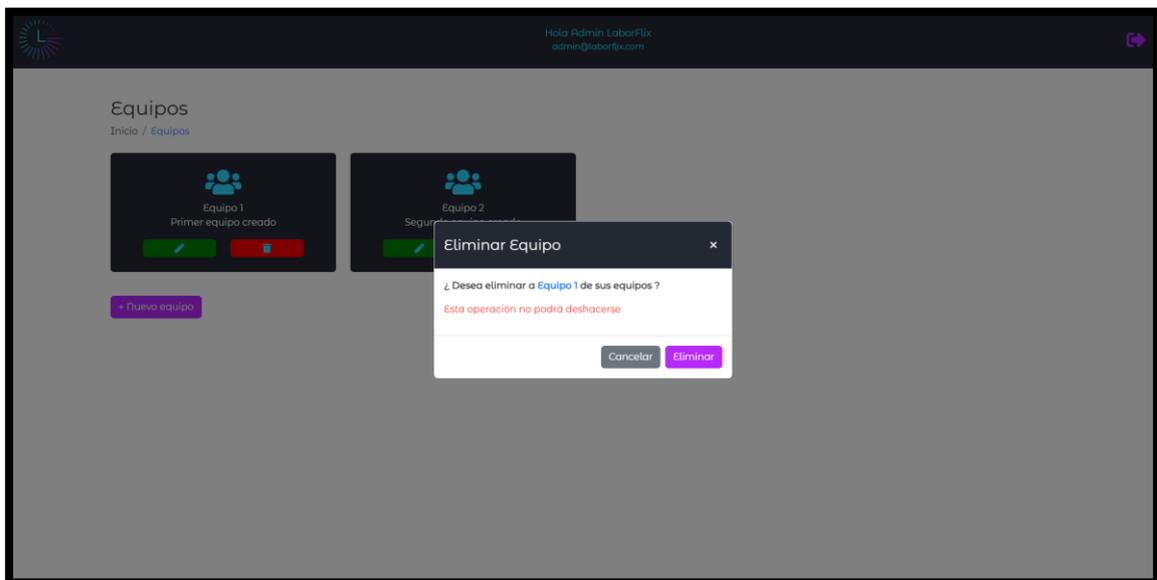


Figura A.11: Pop-up de eliminación de un equipo

II.V. Ubicaciones

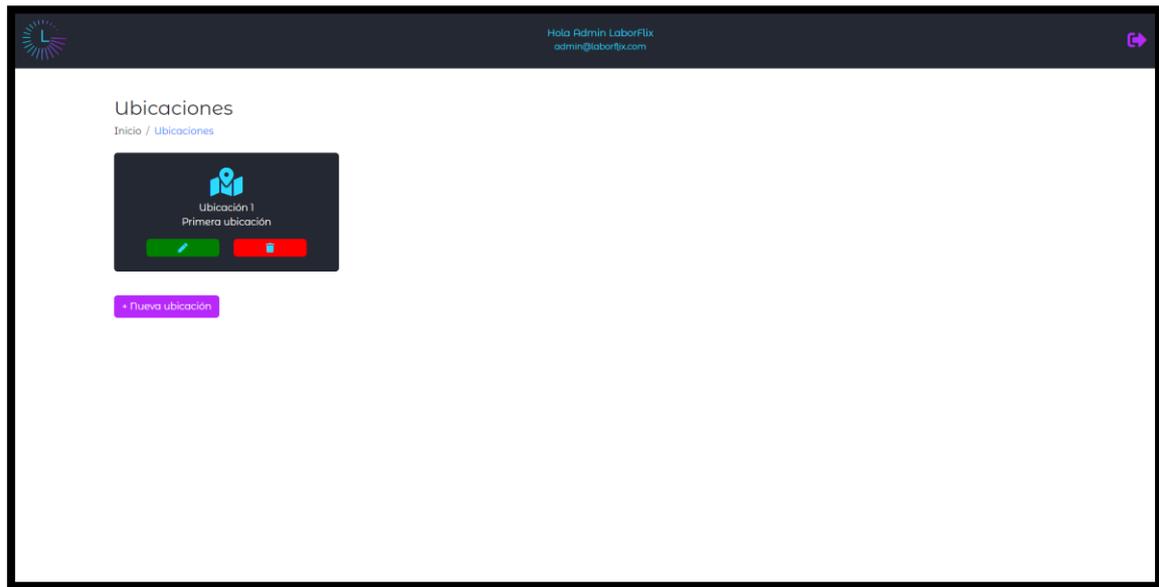


Figura A.12: Página de Ubicaciones

En esta sección se encuentra la gestión y administración de Ubicaciones, pudiendo visualizar las ubicaciones creadas en la plataforma y permitiendo su edición, borrado y creación de nuevas.

Tanto la creación como la edición comparten la misma página, es decir, se visualiza un formulario donde se introducen los datos y equipos asignados a la ubicación.

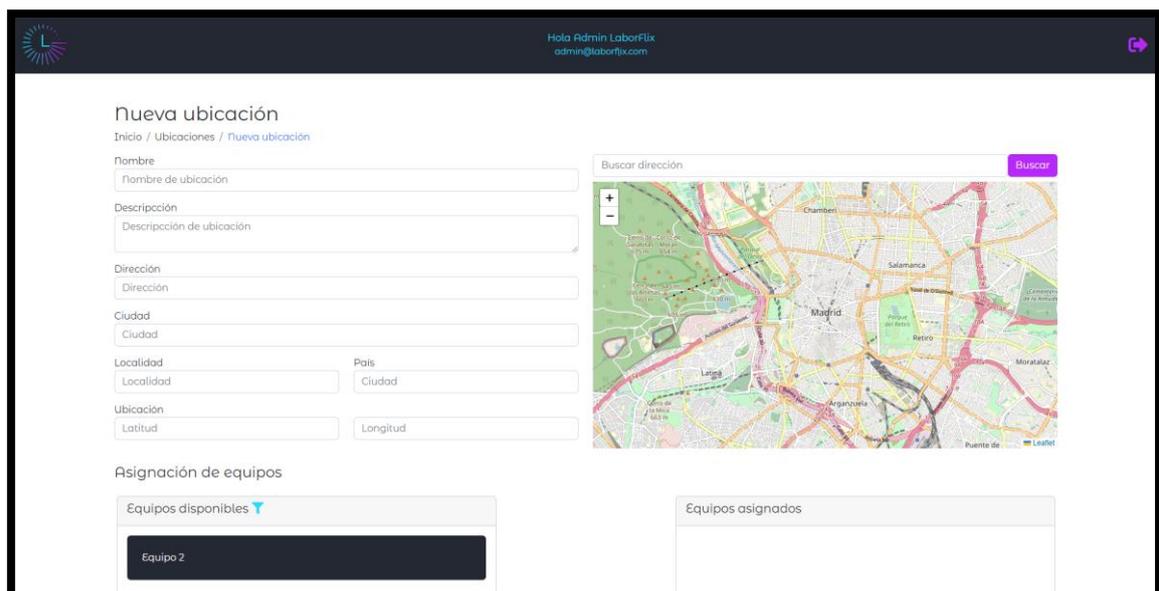


Figura A.13: Página de creación de una ubicación (1)

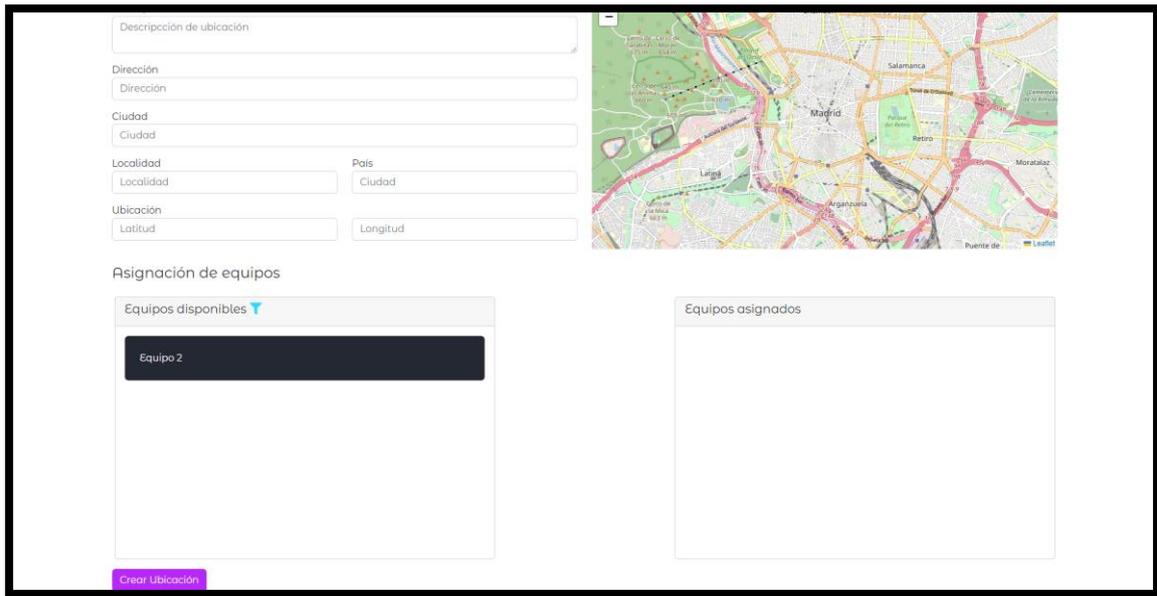


Figura A.14: Página de creación de una ubicación (2)

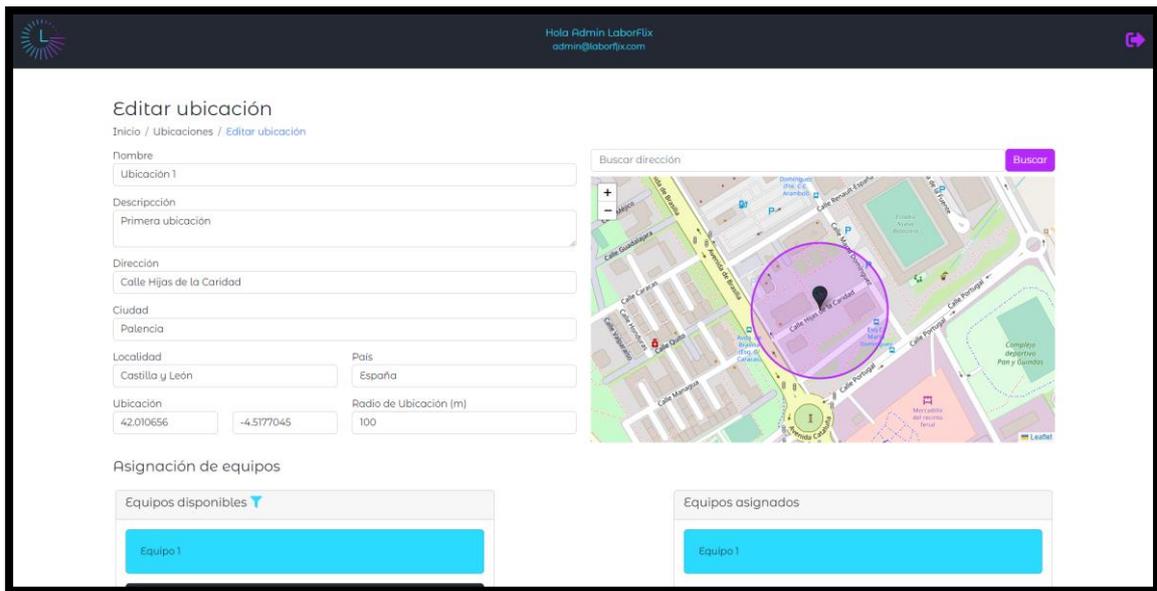


Figura A.15: Página de edición de una ubicación (1)

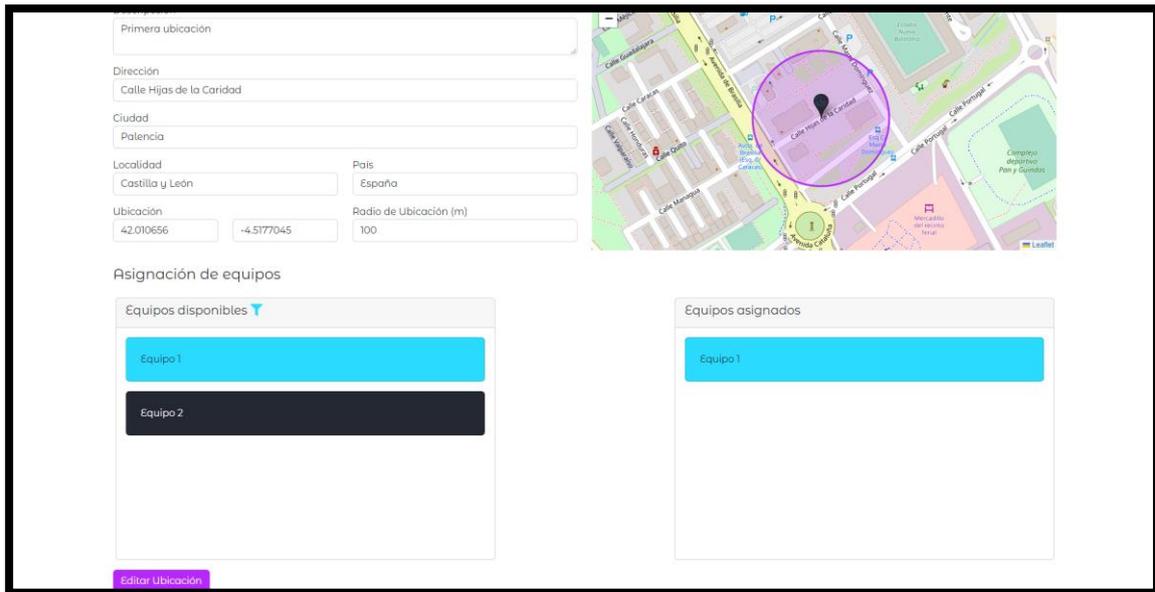


Figura A.16: Página de edición de una ubicación (2)

De estas páginas cabe mencionar que los campos nombre, descripción y radio de ubicación (radio que marcará si un fichaje asignado a esta ubicación está dentro o fuera del rango permitido) son editables para poder introducir la información deseada, sin embargo, todos aquellos campos referentes a la ubicación (dirección, localidad, país y coordenadas) se autocompletan al hacer click sobre el mapa y seleccionar una ubicación.

También podemos hacer uso del buscador de direcciones localizado en la parte superior del mapa, mostrándose todas aquellas ubicaciones coincidentes y permitiéndonos elegir una de estas.

Una vez completado el formulario y la asignación de equipos hacemos click en el botón Crear Ubicación o Editar Ubicación para así reflejar la información.

Por último, la eliminación de una ubicación nos mostrará un pop-up que nos pedirá confirmación para hacer efectivo el borrado.

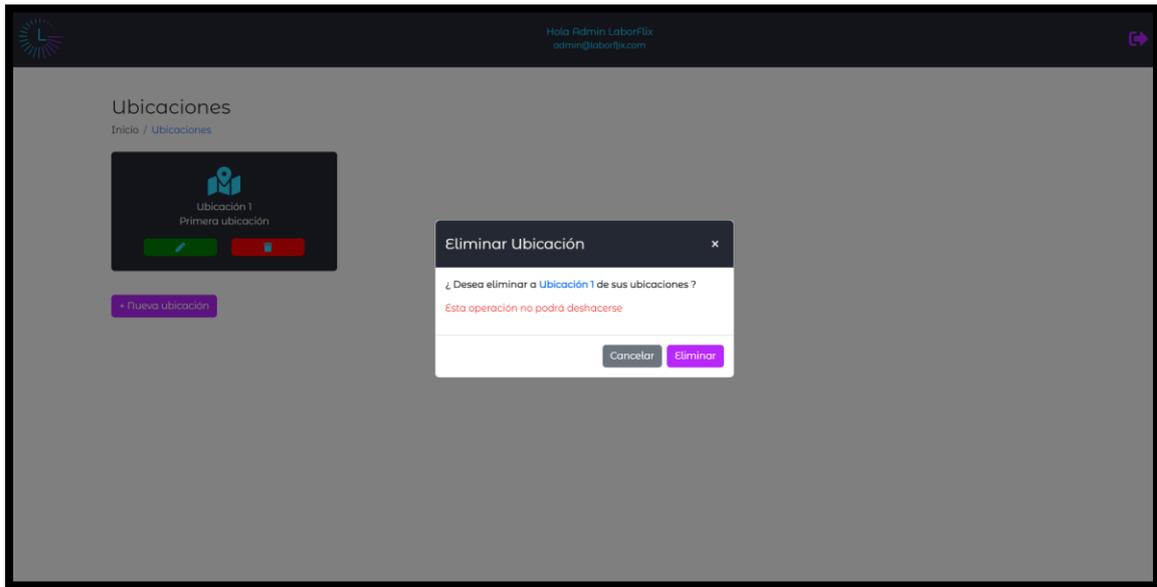


Figura A.17: Pop-up de eliminación de una ubicación

II.VI. Turnos

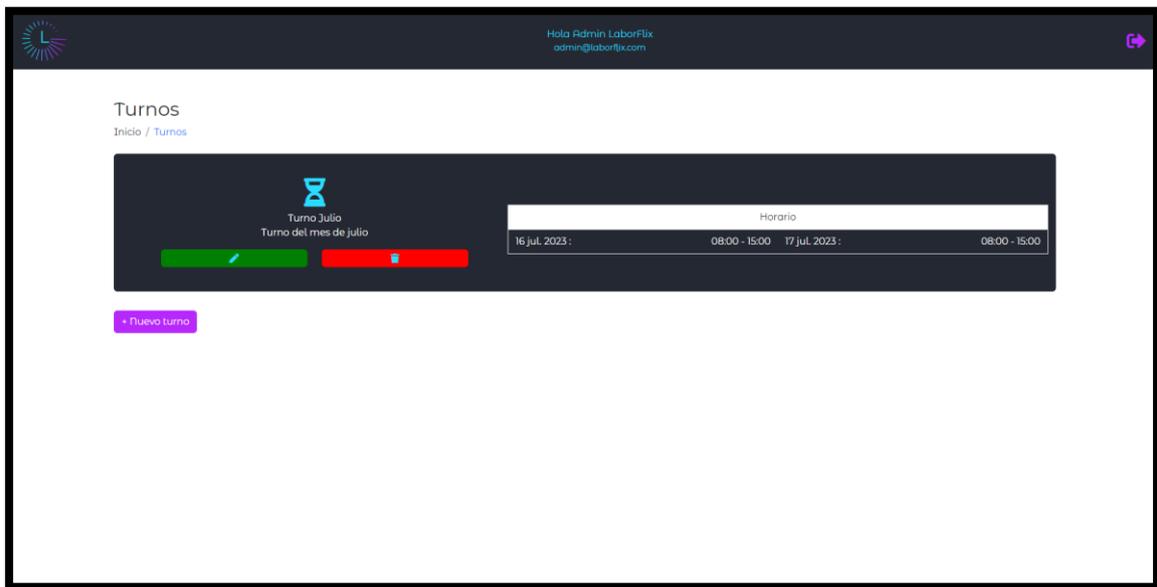


Figura A.18: Página de turnos

En esta sección se encuentra la gestión y administración de Turnos, pudiendo visualizar los turnos creados en la plataforma y permitiendo su edición, borrado y creación de nuevos.

Tanto la creación como la edición comparten la misma página, es decir, se visualiza un formulario donde se introducen los datos y usuarios asignados al turno.

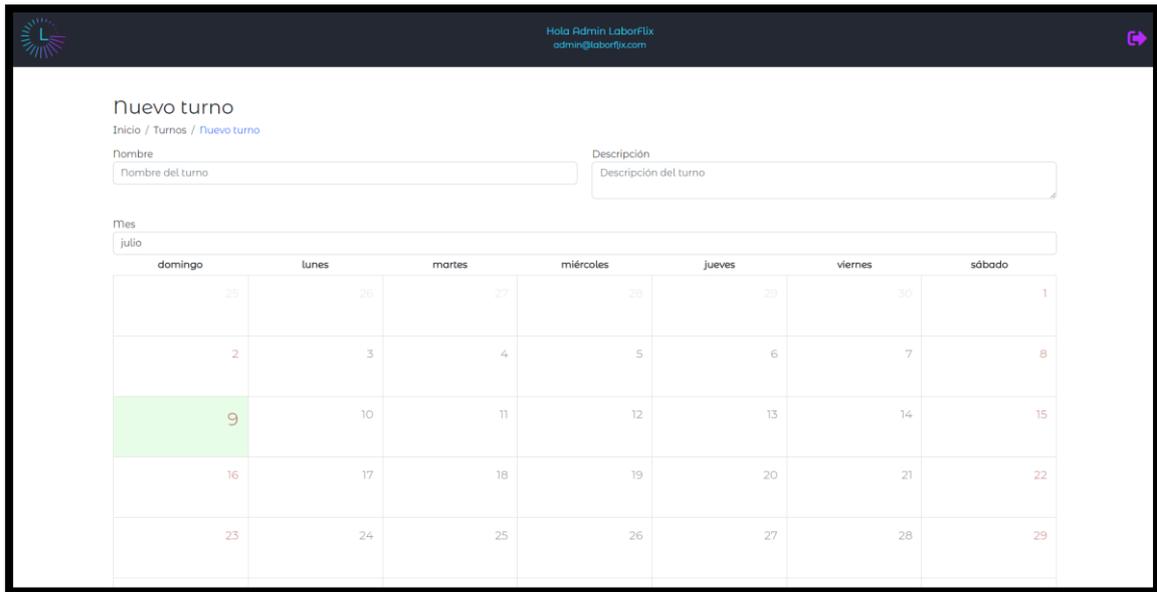


Figura A.19: Página de creación de turnos (1)

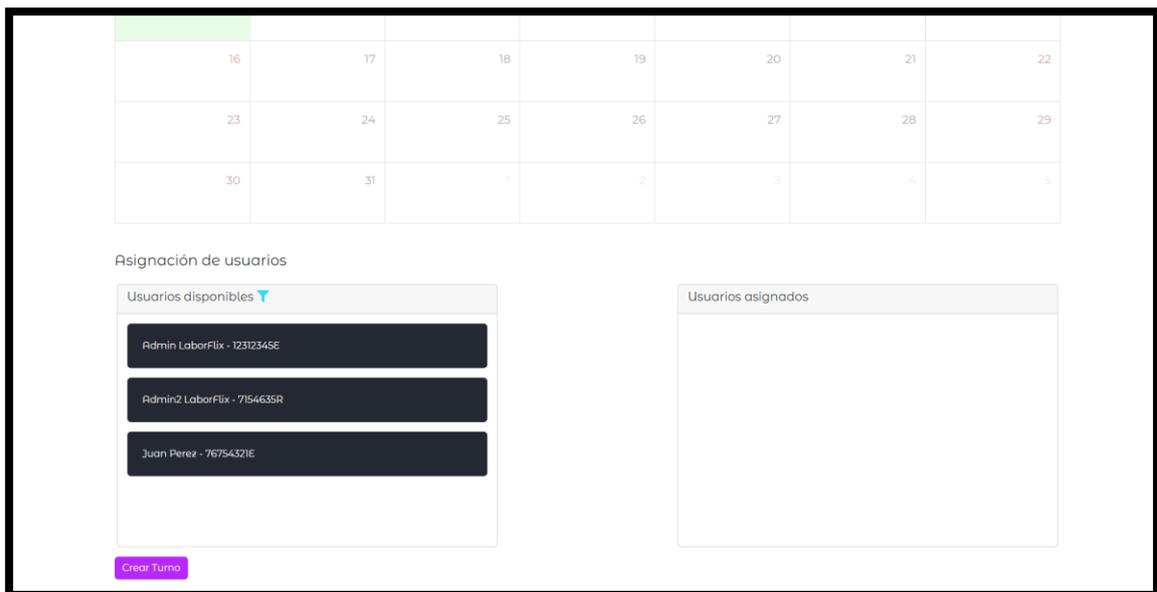


Figura A.20: Página de creación de turnos (2)

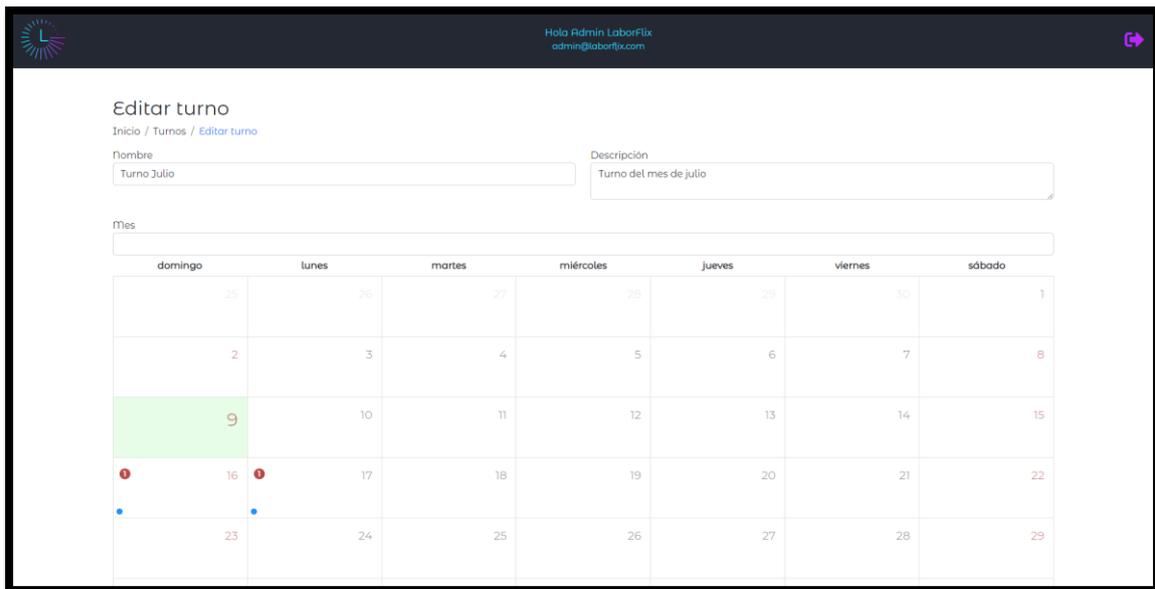


Figura A.21: Página de edición de turnos (1)

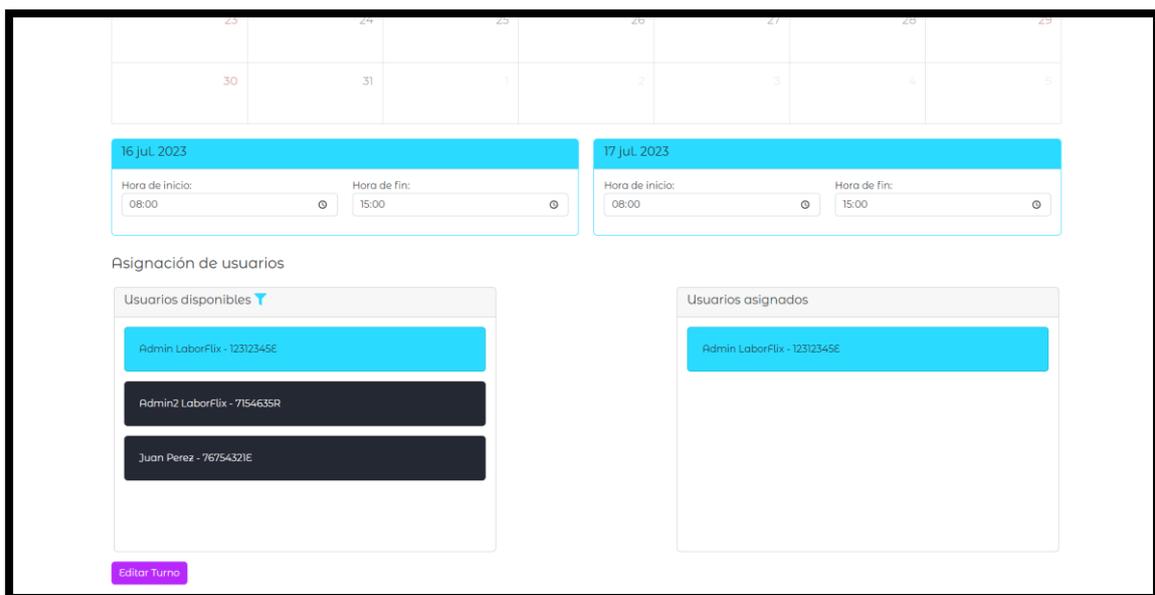


Figura A.22: Página de edición de turnos (2)

Tanto los campos de nombre, descripción y asignación a usuarios se han visto en otras páginas y en esta su comportamiento es el mismo, por lo que nos centraremos en el calendario de selección de días laborales del turno.

Para poder seleccionar los días laborales tenemos que seleccionar previamente el mes que deseemos en el selector “mes” ubicado en la parte superior del calendario, mostrándonos así los días disponibles.

Una vez seleccionemos un día aparecerá una tarjeta debajo del calendario, indicando el día escogido, pero solicitándonos la hora de inicio y la hora de fin de la jornada laboral.

Cuando hayamos seleccionado mínimo un día y hayamos completado el formulario, si hacemos click en el botón Crear Turno o Editar Turno la información se reflejará y seremos redirigidos a la página de Turnos.

Por último, la eliminación de un turno nos mostrará un pop-up que nos pedirá confirmación para hacer efectivo el borrado.

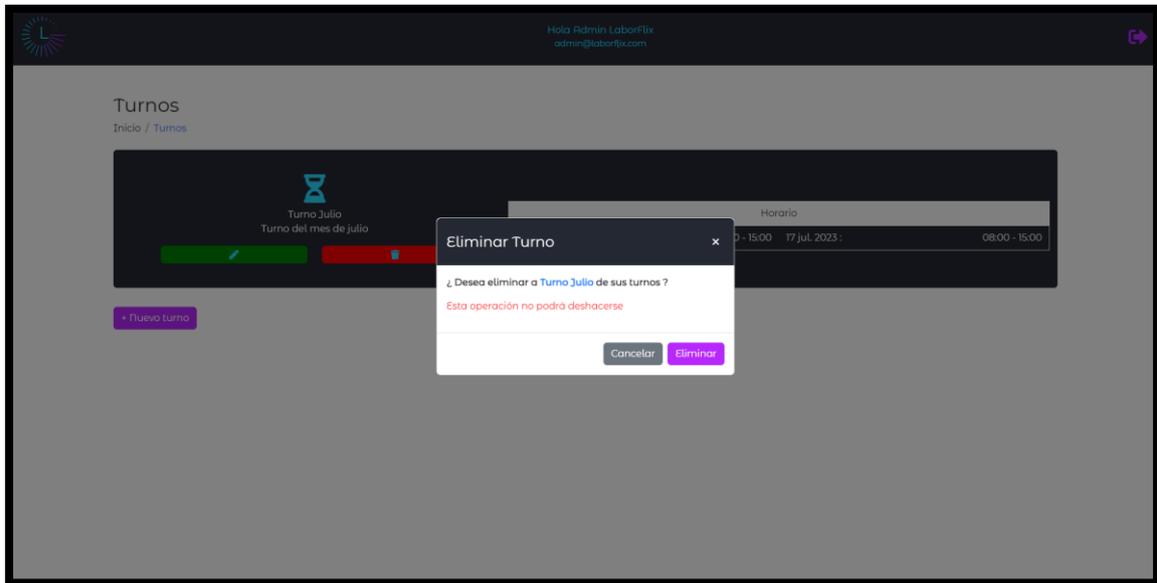


Figura A.23: Pop-up de eliminación de turno

II.VII. Fichaje

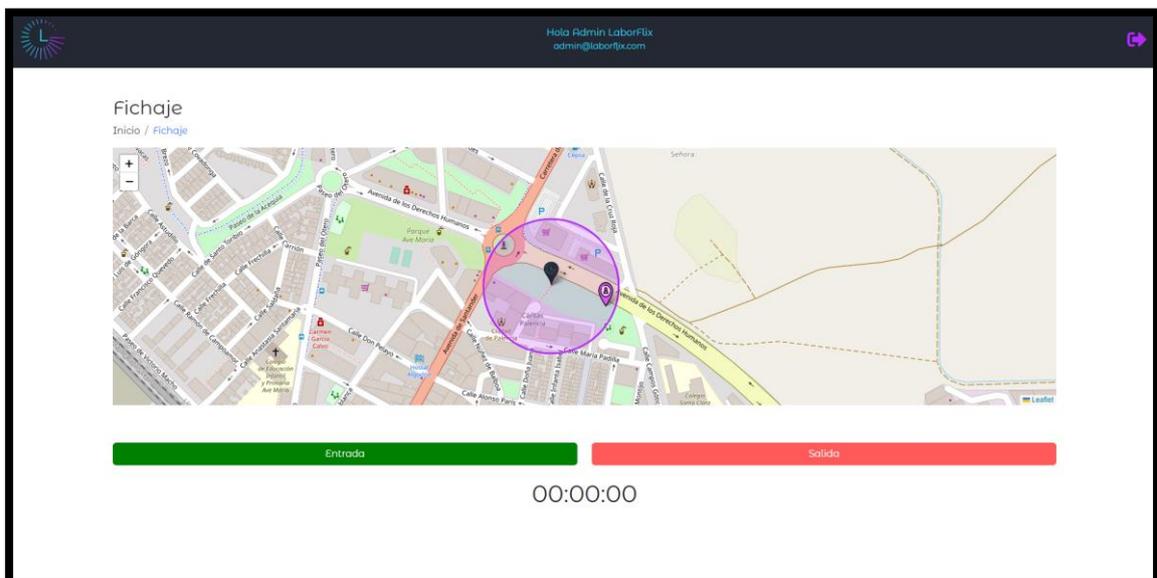


Figura A.24: Página de Fichaje

Está página es común para responsables y trabajadores.

Esta página está formada por un mapa que nos muestra nuestra ubicación actual y la ubicación que un responsable asignó a nuestro equipo. Si al realizar un fichaje no disponemos de equipo de trabajo y/o ubicación, un mensaje informativo en la parte superior derecha nos lo hará saber.

Por otro lado, contamos con dos botones, Entrada y Salida, que marcan el comienzo y final de un fichaje laboral, respectivamente, junto con un contador de tiempo que nos indicará la duración del fichaje actual.

II.VIII. Horario

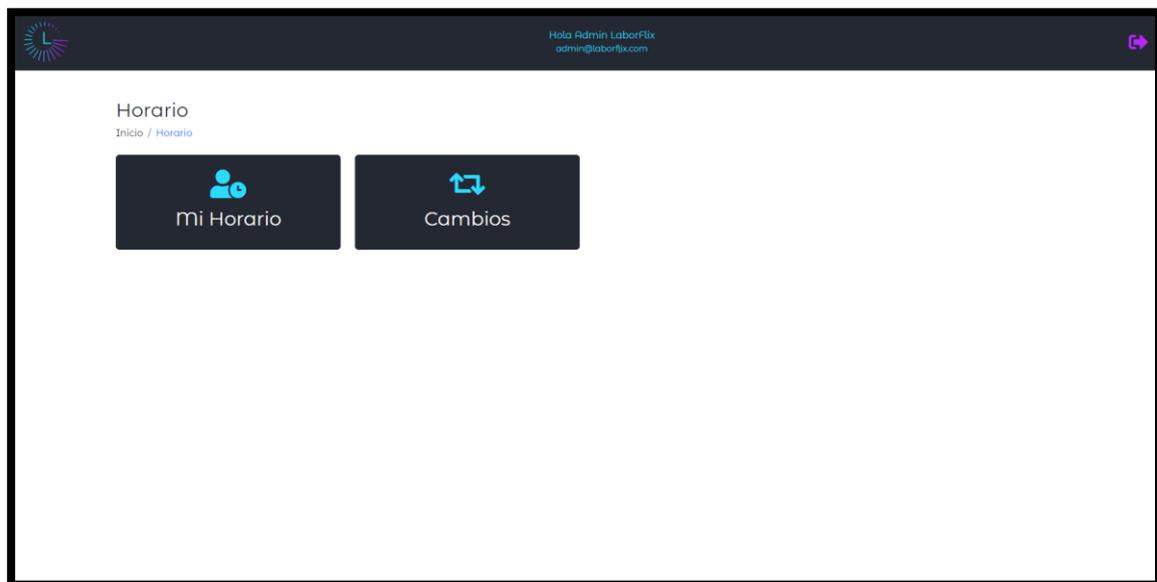


Figura A.25: Página de Horario

En esta página, también común para trabajadores y responsables, contamos con dos enlaces.

En primer lugar, Mi horario, el cual nos permite visualizar los turnos que actualmente tenemos asignados. Si seleccionamos alguno de los días marcados podremos visualizar una tarjeta de información con la hora de inicio y fin de las jornadas de ese día.

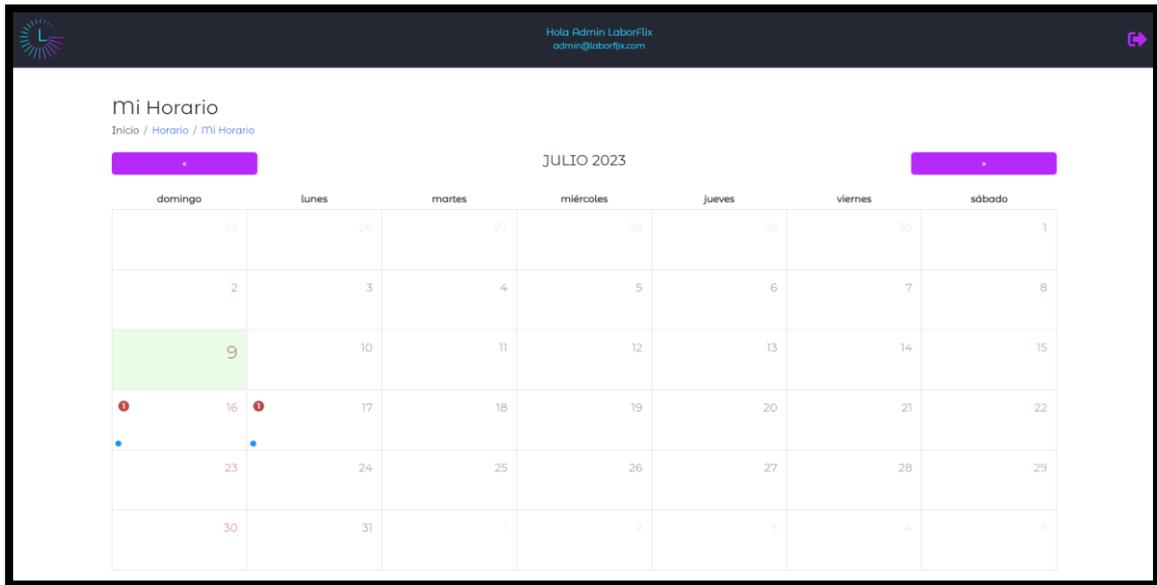


Figura A.26: Página de mi horario (1)

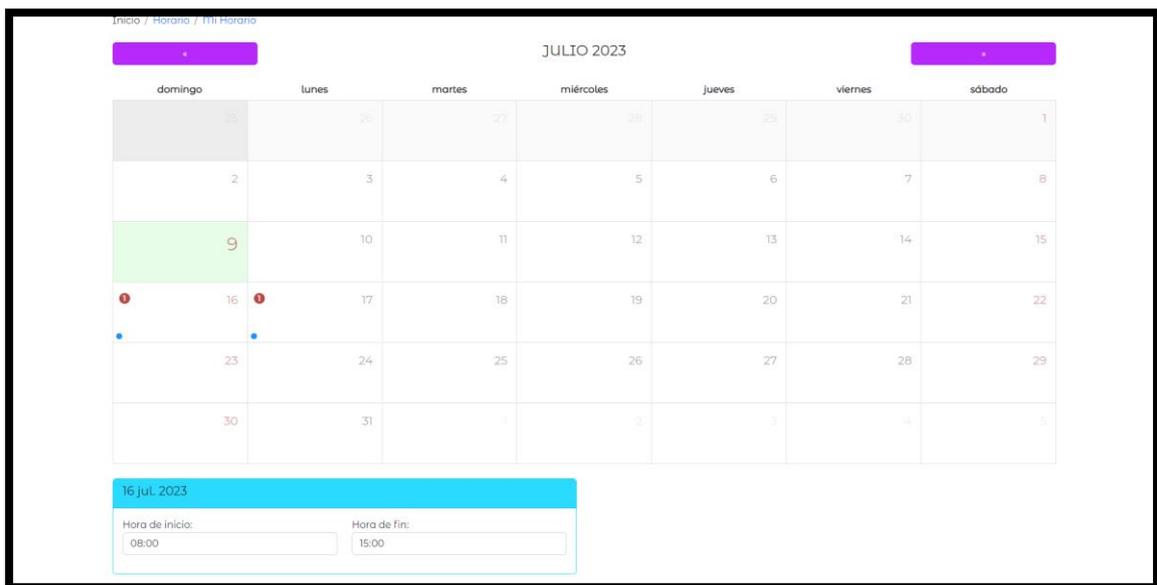


Figura A.27: Página de mi horario (2)

Y, por último, Cambios, donde podremos gestionar tanto nuestras solicitudes de cambio como las peticiones de otros usuarios.

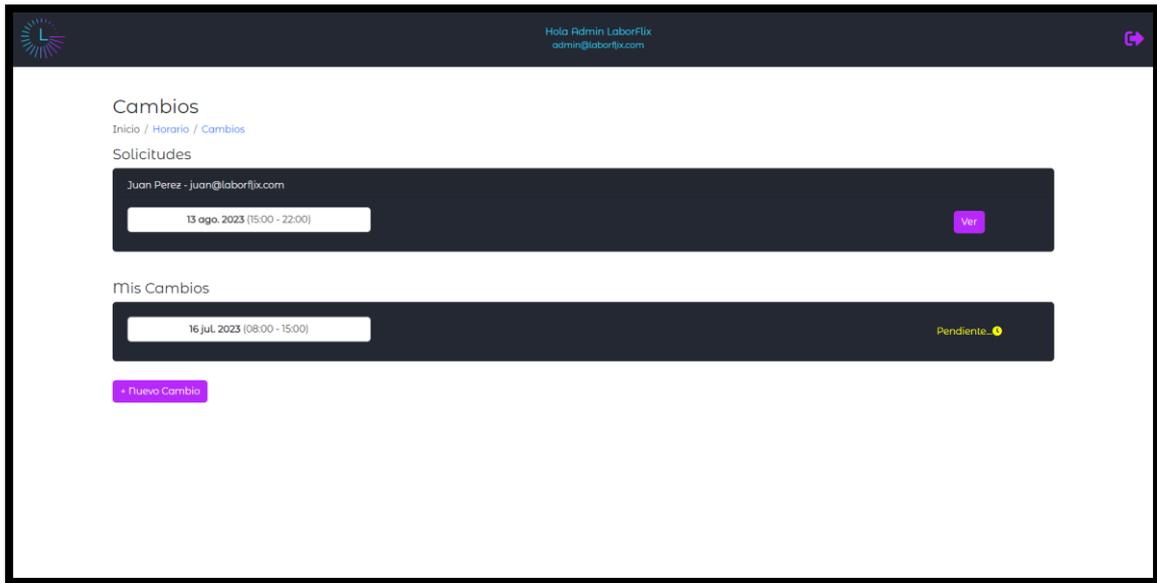


Figura A.28: Página de cambios

Aquí veremos dos secciones diferenciadas que muestran las solicitudes de otros usuarios junto con nuestras solicitudes.

El flujo a realizar sería, en primer lugar, seleccionar el botón Nuevo Cambio, donde se nos muestran nuestros días laborales disponibles, es decir, no están en trámite de ninguna solicitud.

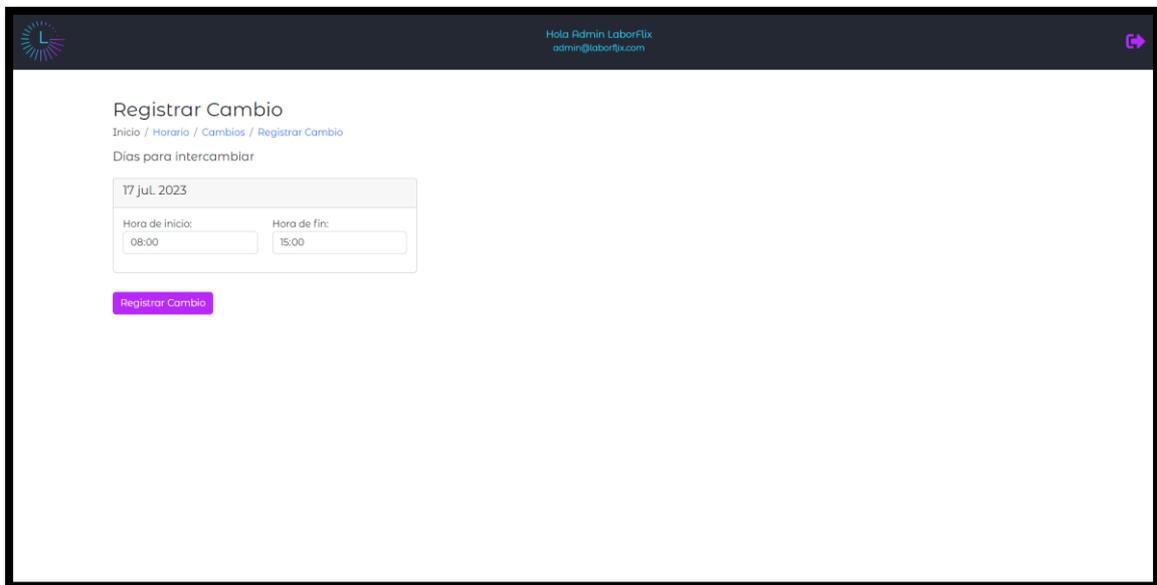


Figura A.29: Página de registrar cambio

Una vez hayamos seleccionado uno o más días pulsamos en el botón Registrar Cambio, para reflejar la solicitud (en este punto visible para el resto de los usuarios) y ser redirigidos a la página de cambios.

Hasta que otro usuario no conteste a nuestra petición esta se nos mostrará en estado Pendiente (como se aprecia en la Figura A.27).

Para continuar con la ejecución el usuario Juan Pérez ha registrado otra solicitud de cambio que nosotros aceptaremos al seleccionar el botón Ver de la solicitud en la página de Cambios.

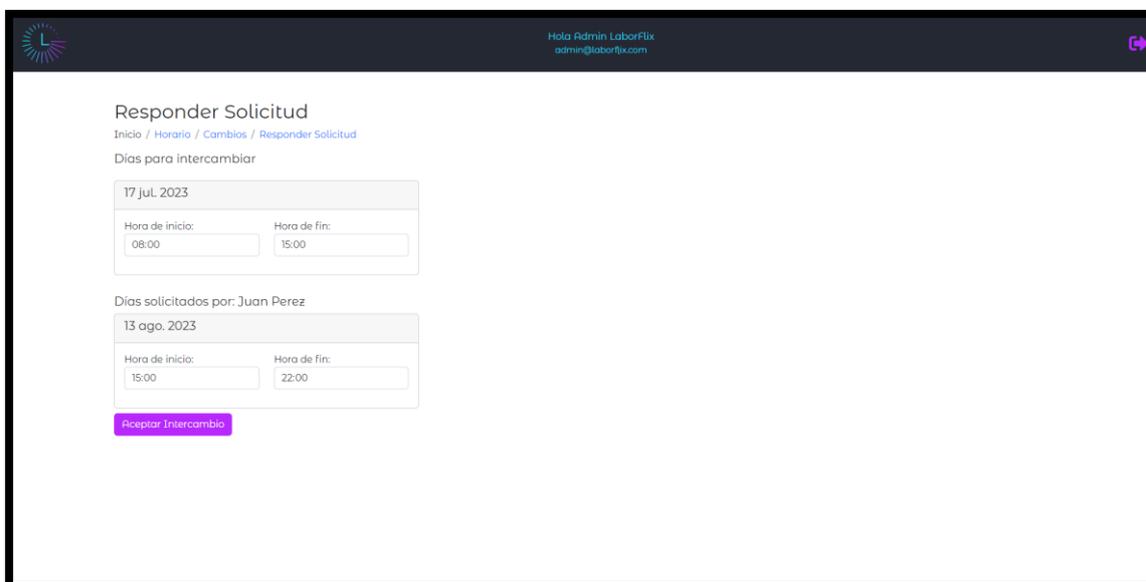


Figura A.30: Página de contestación de solicitud

Aquí visualizaremos en detalle los días que solicita Juan Pérez y los días de los que disponemos para responder a la solicitud. Cabe mencionar que la cantidad de días que propongamos a Juan será la misma que él ha solicitado.

En este caso, al seleccionar un día disponible para el intercambio y hacer click en el botón Aceptar Intercambio se registrará la respuesta y seremos redirigidos a la página de Cambios donde la vista se habrá actualizado, ya que los usuarios que antes podían contestar a esta solicitud ahora la visualizan con el estado En tramite...

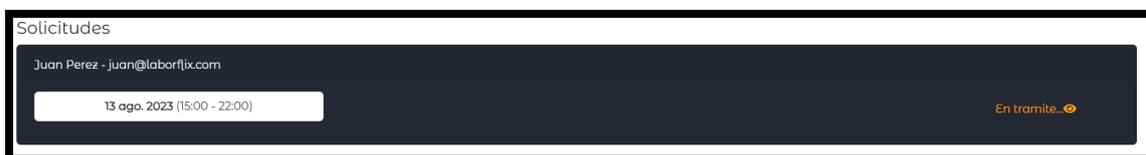


Figura A.31: Solicitud en tramite

Es en este punto cuando el usuario emisor de la solicitud, en este caso Juan Pérez, podrá rechazar o aceptar la respuesta que nosotros como admin@laborflix.com le hemos propuesto, ya que su solicitud no estará en estado Pendiente y se visualizará un botón Ver, el cual al seleccionarlo nos muestra un pop-up con el contenido de la respuesta.

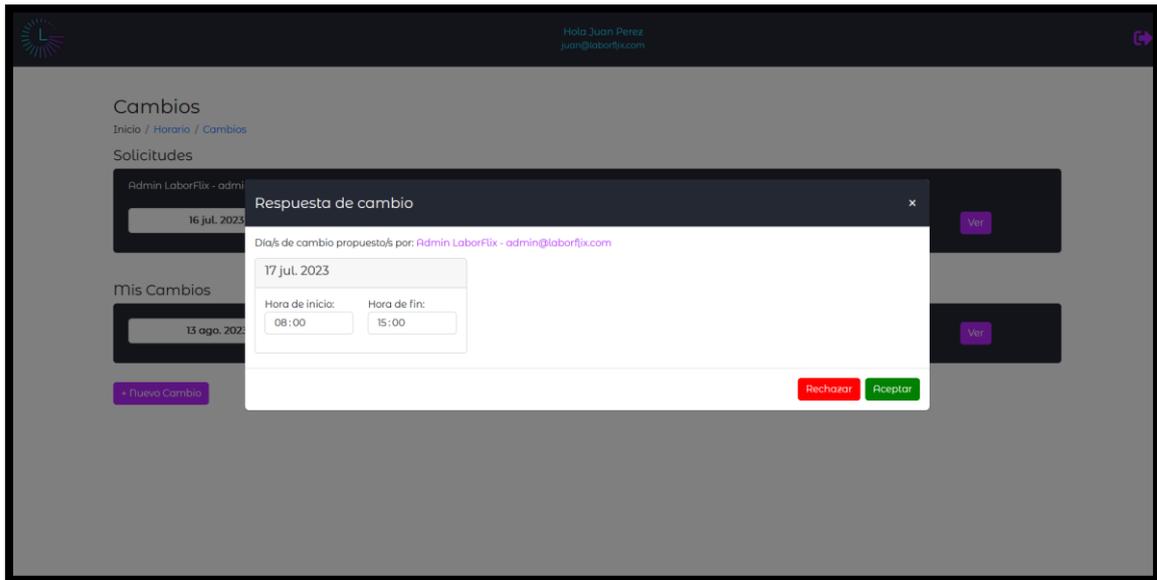


Figura A.32: Pop-up de respuesta a cambio

Si Juan Pérez decide rechazar esta propuesta la solicitud pasará de nuevo a estado Pendiente... y los usuarios podrán volver a contestarla (Figura A.29). Por el contrario, si la acepta, el cambio se hará efectivo y seremos redirigidos a la página de Mi Horario, donde podremos ver el estado resultante de nuestro horario laboral.

II.IX. Informes

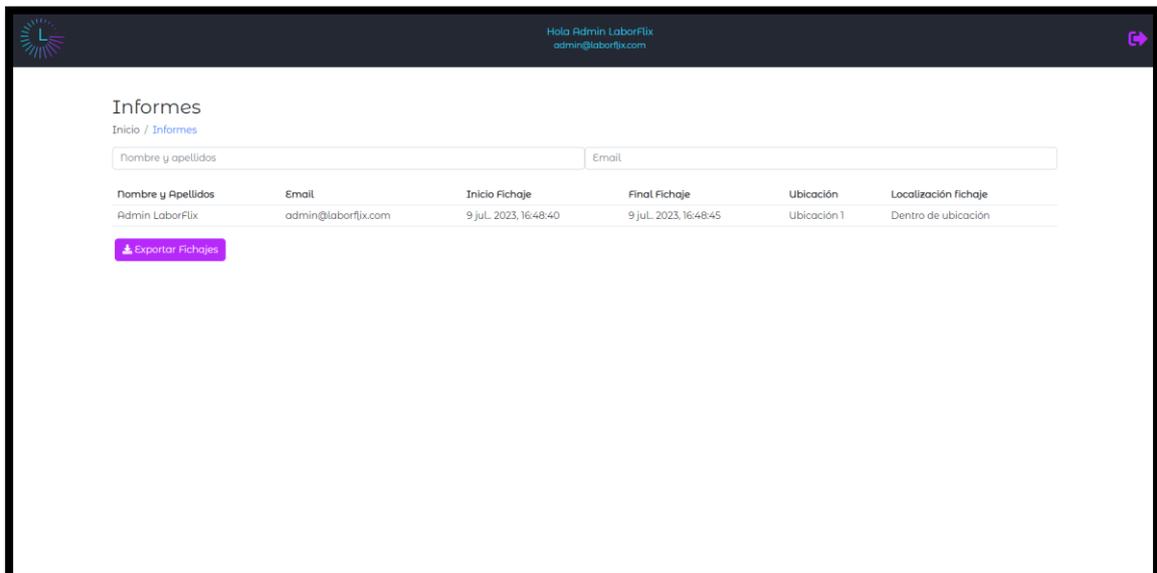


Figura A.33: Informes

En esta página podemos visualizar los fichajes realizados por los usuarios de la plataforma en formato tabla, pudiendo filtrar el contenido por nombre, apellido o email de los usuarios.

Si lo deseamos también podemos exportar toda esta información a un archivo Excel seleccionando el botón de la parte inferior, Exportar Fichajes.

III. Enlaces de código

Los enlaces de acceso al código de este proyecto son:

- Repositorio de la parte frontend:
<https://gitlab.inf.uva.es/jordebu/jorgedebustostfg-frontend>
- Repositorio de la parte backend:
<https://gitlab.inf.uva.es/jordebu/jorgedebustostfg-backend>
- URL de acceso a la plataforma en Heroku:
<https://jorge-de-bustos-frontend-4b02b6921b8f.herokuapp.com/>