



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención Ingeniería del Software)

Ficha2 – Sistema para el registro de la jornada laboral

Autor:

David González Pérez

Tutor:

Dr. Joaquín Adiego Rodríguez

RESUMEN

En este proyecto se expone el desarrollo de un sistema para el registro de la jornada laboral, compuesto por una aplicación Android como la parte Frontend, y una API Rest en NodeJs como la parte Backend junto con MongoDB como base de datos elegida. Todo el desarrollo se ha llevado a cabo siguiendo el marco de trabajo ágil Scrum.

La idea de este trabajo surgió tras la entrada en vigor del Real Decreto-ley 8/2019 que obliga a las empresas a realizar un registro de las horas de entrada y salida de sus empleados, con el propósito de proporcionar a pequeñas empresas una herramienta con la que poder cumplir el Real Decreto-ley.

ABSTRACT

In this project the development of a system for the registration of the working day is exposed, consisting of an Android application as the Frontend part, and an API Rest in NodeJs as the Backend part together with MongoDB as the chosen database. All development has been carried out following the agile Scrum framework.

The idea of this work arose after the entry into force of Real Decreto-ley 8/2019 that requires companies to make a record of the hours of entry and exit of their employees, in order to provide small companies with a tool with which can comply with the Real Decreto-ley.

Índice

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Descripción del proyecto.....	1
2	Estado del arte.....	2
2.1	Aplicaciones ya existentes.....	2
2.1.1	Control Laboral.....	2
2.1.2	Hora Laboral.....	3
2.2	Tecnologías utilizadas.....	4
2.2.1	Git.....	4
2.2.2	Android.....	5
2.2.3	Npm.....	5
2.2.4	NodeJS.....	6
2.2.5	MongoDB.....	6
2.2.6	Heroku.....	7
3	Análisis.....	8
3.1	Actores.....	8
3.2	Casos de uso.....	9
3.2.1	Descripción CU Registro entrada diferido.....	10
3.2.2	Descripción CU Registro completo diferido.....	10
3.2.3	Descripción CU Completar registro.....	11
3.2.4	Descripción CU Consultar registros.....	11
3.2.5	Descripción CU Consultar registros de empleados.....	11
3.2.6	Descripción CU Generar PDF.....	12
3.3	Diagramas de actividad.....	13
3.3.1	Registro completo diferido.....	13
3.3.2	Consultar registros de empleado generando un PDF.....	14
3.4	Historias de usuario.....	15
3.5	Tareas.....	17
4	Planificación.....	21
4.1	Scrum.....	21
4.1.1	Roles.....	21
4.1.2	Artefactos.....	22
4.1.3	Eventos.....	22
4.2	Product Backlog.....	23
4.3	Sprints.....	23
4.3.1	Sprint 1.....	23
4.3.2	Sprint 2.....	24
4.3.3	Sprint 3.....	25

4.3.4	Sprint 4.....	26
4.3.5	Sprint 5.....	27
5	Ejecución del proyecto	28
5.1	Arquitectura.....	28
5.1.1	Características de la arquitectura REST	28
5.1.2	Ventajas de la arquitectura REST.....	28
5.2	Patrones de diseño	29
5.2.1	DAO.....	29
5.2.2	DTO	29
5.2.3	Façade.....	30
5.3	Despliegue	30
5.4	Diseño de la base de datos	31
5.5	Diagramas de secuencia	32
5.5.1	Registro completo diferido.....	33
5.5.2	Obtener registros de un empleado	35
5.6	Interfaces gráficas.....	37
5.6.1	Splash screen	37
5.6.2	Login	38
5.6.3	Vista principal o de acceso rápido	39
5.6.4	Vista del registro diferido de entrada-salida	40
5.6.5	Vista para consultar registros.....	41
5.6.6	Vista cambio de contraseña	42
5.6.7	Vista de un administrador para consultar registros	43
5.6.8	Vista de un administrador para añadir empleado.....	44
5.6.9	Ejemplo de PDF con los registros realizados	45
5.7	Pruebas	46
5.7.1	Pruebas durante los Sprints.....	46
5.7.2	Pruebas de usabilidad.....	51
6	Conclusiones y trabajo futuro	54
6.1	Trabajo futuro.....	54
7	Bibliografía.....	55

1 Introducción

1.1 Motivación

El 12 de marzo de 2019 entró en vigor el Real Decreto-ley 8/2019 que obliga a las empresas a llevar un registro del inicio y finalización de la jornada de trabajo de sus empleados [\[1\]](#). En pequeñas empresas y comercios, esta nueva ley podría suponer un problema a la hora de realizar dicho registro de forma rápida y sencilla. Por ello, planteé este proyecto como una posible solución. Desarrollarla mediante una aplicación Android aporta la flexibilidad que muchas empresas necesitan para que sus empleados puedan realizar el registro de entrada y salida del trabajo incluso cuando sus labores se realizan fueran de una oficina.

Por otro lado, la elección de usar NodeJS y MongoDB para el desarrollo del backend fue tomada con la idea de poder aplicar los conocimientos adquiridos durante la carrera en dos de las tecnologías que más me llamaron la atención, ya que usan el lenguaje JavaScript en el lado del servidor, que tradicionalmente solo era usado en los navegadores web.

1.2 Descripción del proyecto

El sistema está formado por dos partes: una aplicación Android y un backend formado por una API desarrollada en NodeJS, y MongoDB como base de datos.

La aplicación Android permitirá a los empleados registrar su hora de entrada y salida. Además, como en el Real Decreto-ley se indica que los registros deben estar disponibles tanto para los sindicatos como para los propios empleados, será desde la misma aplicación Android donde podrán consultar los registros. A su vez, para garantizar la protección de datos, los empleados solo podrán acceder a sus propios registros. Si el empleado lo desea, podrá descargar un PDF con sus registros realizados.

La aplicación Android hará uso de la API para guardar y obtener los registros de los empleados. La API se ha desarrollado mediante el framework ExpressJS y el ODM Mongoose para hacer que la comunicación con la base de datos sea más sencilla.

Para que la API sepa qué usuario está realizando un registro, se usa el estándar JWT como método de autenticación basada en token. En cada petición que realice la aplicación, esta tendrá que enviar en la cabecera HTTP Authorization el token JWT. Para obtener este token, en la API existe un endpoint el cual al introducir el usuario y contraseña devuelve el token JWT que podrá ser usado durante un día antes de que expire y el usuario tenga que volver a iniciar sesión.

En el sistema existen dos tipos de usuarios, los usuarios normales y los administradores. Mediante una cuenta de administrador se podrá añadir nuevos usuarios al sistema, indicando la información mínima necesaria: nombre, apellidos, login y contraseña. Esta contraseña inicial la podrá cambiar el usuario una vez haya iniciado sesión.

2 Estado del arte

2.1 Aplicaciones ya existentes

En este apartado se describirán otras aplicaciones que existen actualmente para el registro de la jornada laboral de los empleados.

2.1.1 Control Laboral

Se trata de un sistema gestión de horas laborales con el que principalmente las empresas podrán realizar el fichaje de las horas trabajadas por los empleados y generar informes en diferentes formatos [2]. Además, también permite un control de los calendarios vacacionales y festivos de cada empleado. Todos los datos los almacenan en la nube cumpliendo con la Ley Orgánica de Protección de Datos (LOPD). El uso de este sistema tiene un coste que va desde los 19€ hasta los 69€ mensuales dependiendo del número de empleados. A continuación, se muestran unas capturas de pantalla tomadas desde su página de la Play Store de Android.

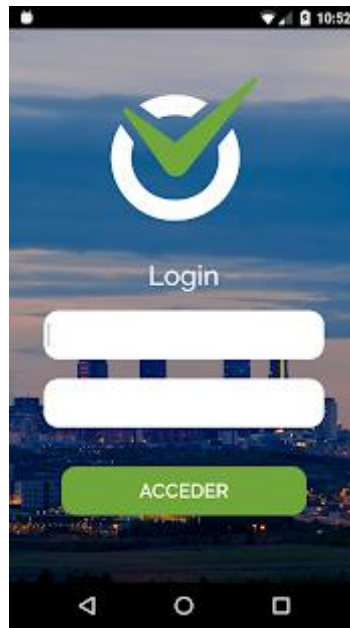


Figura 1 Captura de pantalla del inicio de sesión de la aplicación Control Laboral



Figura 2 Captura de pantalla de la visualización de un registro en la aplicación Control Laboral



Figura 3 Captura de pantalla de la validación de un registro en la aplicación Control Laboral

2.1.2 Hora Laboral

Hora Laboral [3] es una aplicación con un coste de 99€ anuales con la que los empleados pueden validar su horario y realizar el fichaje de sus horas trabajadas. A diferencia de la anterior, Hora Laboral no tiene la posibilidad de tener un calendario de vacaciones o festivos. Un punto en contra de esta aplicación es que no mencionan en ningún momento cómo almacenan los datos de los usuarios.

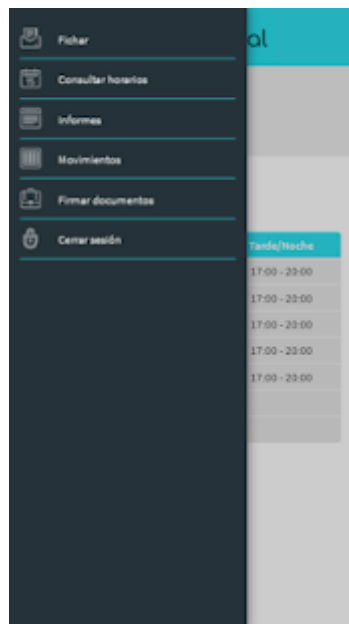


Figura 4 Captura de pantalla del menú principal de la aplicación Hora Laboral



Figura 5 Captura de pantalla del horario de trabajo de un empleado en la aplicación Hora Laboral

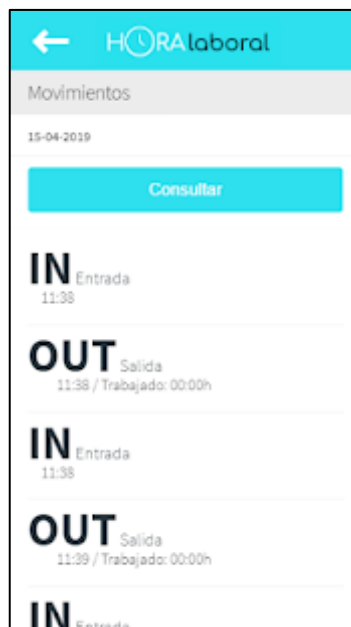


Figura 6 Captura de pantalla de los registros de entrada y salida de un empleado en la aplicación Hora Laboral

2.2 Tecnologías utilizadas

En esta sección se detalla de forma general las diferentes tecnologías y herramientas empleadas para el desarrollo del proyecto.

2.2.1 Git

Git [\[4\]](#) es un sistema de control de versiones distribuido, gratuito y de código abierto diseñado para manejar tanto proyectos grandes como pequeños, con rapidez y eficiencia.

2.2.1.1 GitHub

Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git [\[5\]](#). Para el desarrollo del proyecto se ha utilizado la cuenta personal del estudiante creando dos repositorios, uno para la aplicación Android y otro para la API.

2.2.2 Android

Android es un sistema operativo móvil desarrollado por Google basado en el kernel de Linux y otros softwares de código abierto [6]. Es el sistema operativo para móviles más usado en el mundo con cerca de un 75% de cuota del mercado [7].

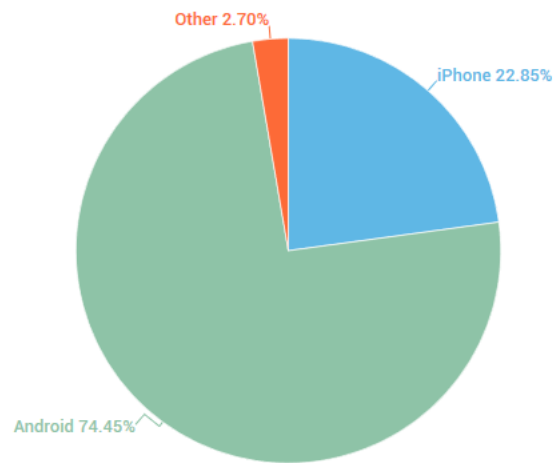


Figura 7 Comparativa de cuota de mercado entre Android y iPhone entre enero de 2018 y enero de 2019

Uno de los principales motivos por los que Android es tan popular, es debido a que Google liberó Android bajo licencia Apache, haciendo que los fabricantes de terminales puedan usarlo en sus dispositivos. A pesar de la alta cuota de mercado de dispositivos Android, los ingresos generados por las aplicaciones publicadas en la Google Play (tienda de aplicaciones de Android) es significativamente inferior al store de su principal competente iOS [8].

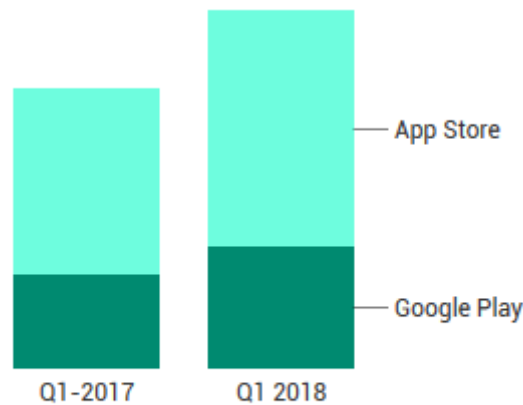


Chart: Xataka Móvil - Fuente: Sensor Tower - Get the data - Created with Datawrapper

Figura 8 Ingresos de Google Play y App Store en el primer trimestre de 2017 y 2018

Sin embargo, cada vez son más los desarrolladores optan por publicar sus aplicaciones en Google Play. Esta tendencia es incentivada por su comunidad de desarrolladores, una de las más grandes a nivel mundial. Esto, unido a que para publicar una aplicación en su store solo es necesario el pago de una única tarifa de \$25, mientras que en iOS, la App Store cobra una tarifa de \$99 anuales.

2.2.3 Npm

Desarrollado en JavaScript, Npm [9] es el registro de software más grande con más de 800000 paquetes de código además de ser el gestor de paquetes y módulos por defecto de NodeJs. Con él podemos manejar las

dependencias de nuestro proyecto de forma eficiente y centralizada mediante un único fichero *package.json*. Para añadir una nueva dependencia a nuestro proyecto, lo único que debemos hacer es añadirla al *package.json* y ejecutar un comando para que npm se encargue de descargar e instalar lo que esta nueva dependencia necesite [\[10\]](#).

2.2.4 NodeJS

NodeJs [\[11\]](#) es un entorno en tiempo de ejecución de JavaScript basado en el motor de JavaScript V8 de Google. Fue creado con la razón principal de poder desarrollar programas de red con una alta escalabilidad ya que los entornos tradicionales, como los basados en Java o PHP, sufrían problemas de servicio cuando el número de usuarios concurrentes era elevado. Esto es debido a que cada conexión, que crea una solicitud, genera un nuevo subproceso, ocupando la RAM del sistema y regularmente maximizando la cantidad de RAM disponible. Sin embargo, NodeJS trabaja en un solo subproceso, utilizando el modelo entrada y salida sin bloqueo controlado por eventos, lo que permite soportar un mayor número de conexiones concurrentes mantenidas en bucles de eventos [\[12\]](#).

Desde el punto de vista de desarrollo de servidor web, NodeJS tiene un gran número de ventajas:

- Al usar JavaScript como lenguaje de programación, se pierde menos tiempo a la hora de alternar entre el desarrollo del backend y del frontend.
- El gestor de paquetes de NodeJS (Npm) proporciona acceso a miles de paquetes reutilizables.
- Es portable, con versiones que funcionan en Microsoft Windows, OS X, Linux, Solaris, etc.
- Tiene una comunidad de desarrolladores muy activa.

2.2.4.1 ExpressJS

Se trata de uno de los frameworks de NodeJS más usados actualmente para el desarrollo de aplicaciones web y APIs [\[13\]](#), caracterizado por su velocidad, flexibilidad y minimalismo. Proporciona mecanismos para [\[14\]](#):

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes rutas URL.
- Integración con motores de renderizado de vistas para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones “middleware” adicional en cualquier punto dentro de la tubería de manejo de peticiones.

2.2.5 MongoDB

MongoDB es un sistema de base de datos NoSQL multiplataforma orientado a documentos y de esquema libre, es decir, cada registro puede tener un esquema de datos diferente, con atributos que no tienen por qué repetirse de un registro a otro. Se caracteriza por su velocidad y su rico, y sencillo, sistema de consultas.

En MongoDB, cada registro o conjunto de datos se denomina documento. Los documentos se pueden agrupar en colecciones, las cuales serían el equivalente a las tablas en las bases de datos relacionales. Los distintos documentos se almacenan en formato BSON, Binary JSON, que es una versión modificada de JSON que permite búsquedas rápidas de datos [\[15\]](#).

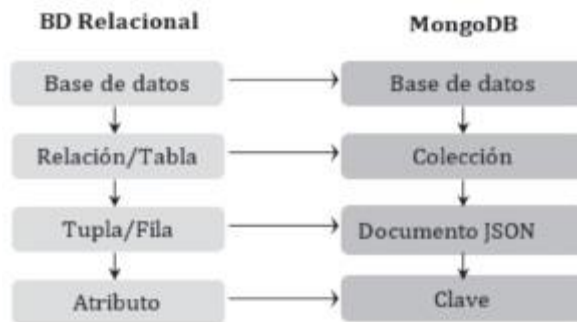


Figura 9 Comparación de términos entre una BD relacional y MongoDB

2.2.5.1 MongoDB Atlas

MongoDB Atlas [\[16\]](#) es la base de datos como servicio (DaS) que permite implementar, utilizar y escalar una base de datos de MongoDB fácilmente. Las principales ventajas de usar MongoDB Atlas son:

- Una rápida puesta en marcha de los clusters.
- Total escalabilidad, pudiendo realizar un escalado tanto horizontal como en vertical sin interrumpir la actividad.
- Copias de seguridad continuas con recuperación en un punto en el tiempo.
- Autenticación y cifrado.
- Modelo de planes de precio según la demanda.

2.2.6 Heroku

Heroku es una plataforma como servicio (PaaS) en la nube basada en contenedores Linux llamados “dynos”. Estos contenedores son utilizados para alojar las aplicaciones web, webservices o aplicaciones que se ejecuten del lado del servidor. El uso de Heroku nos permite ejecutar nuestra aplicación sin tener que preocuparnos por poner a punto el sistema sobre el que se lanzará. Las principales ventajas de Heroku son:

- Posee un plan gratuito.
- Los dynos se encuentran aislados entre sí.
- La infraestructura de Heroku utiliza AWS (Amazon Web Services), garantizando un alto rendimiento, disponibilidad y estabilidad.
- Los dynos cuentan con una buena capacidad de procesamiento, por lo que las aplicaciones se ejecutan rápidamente entregando tiempos de respuesta más que aceptables.
- Es muy sencillo redimensionar los dynos, pudiendo ser escalados tanto vertical (agregando más recursos a los dynos) como horizontalmente (agregando más dynos).

3 Análisis

3.1 Actores

Se distinguen tres actores principales, **Usuario**, **Empleado** y **Administrador**. Un **Usuario** es cualquiera que interactúe con el sistema sin llegar a identificarse. Sin embargo, un **Empleado** es un **Usuario** que se ha identificado ante el sistema. A su vez, el **Administrador** es un **Empleado** que puede realizar funciones especiales que un empleado no puede.

Se puede observar que existe una relación de herencia entre los distintos actores del sistema, tal y como se indica en la siguiente ilustración.

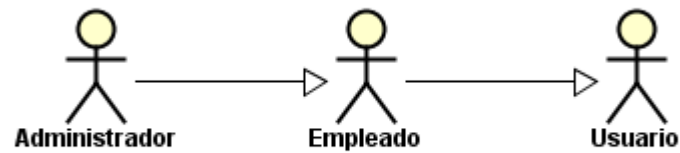


Figura 10 Diagrama de actores

3.2 Casos de uso

A pesar de haber seguido la metodología ágil Scrum, se ha elaborado un diagrama de casos de uso para poder tener una visión general rápida de las acciones que pueden realizar cada actor. Además, se ha realizado la descripción de los casos de uso, salvo aquellos que son triviales.

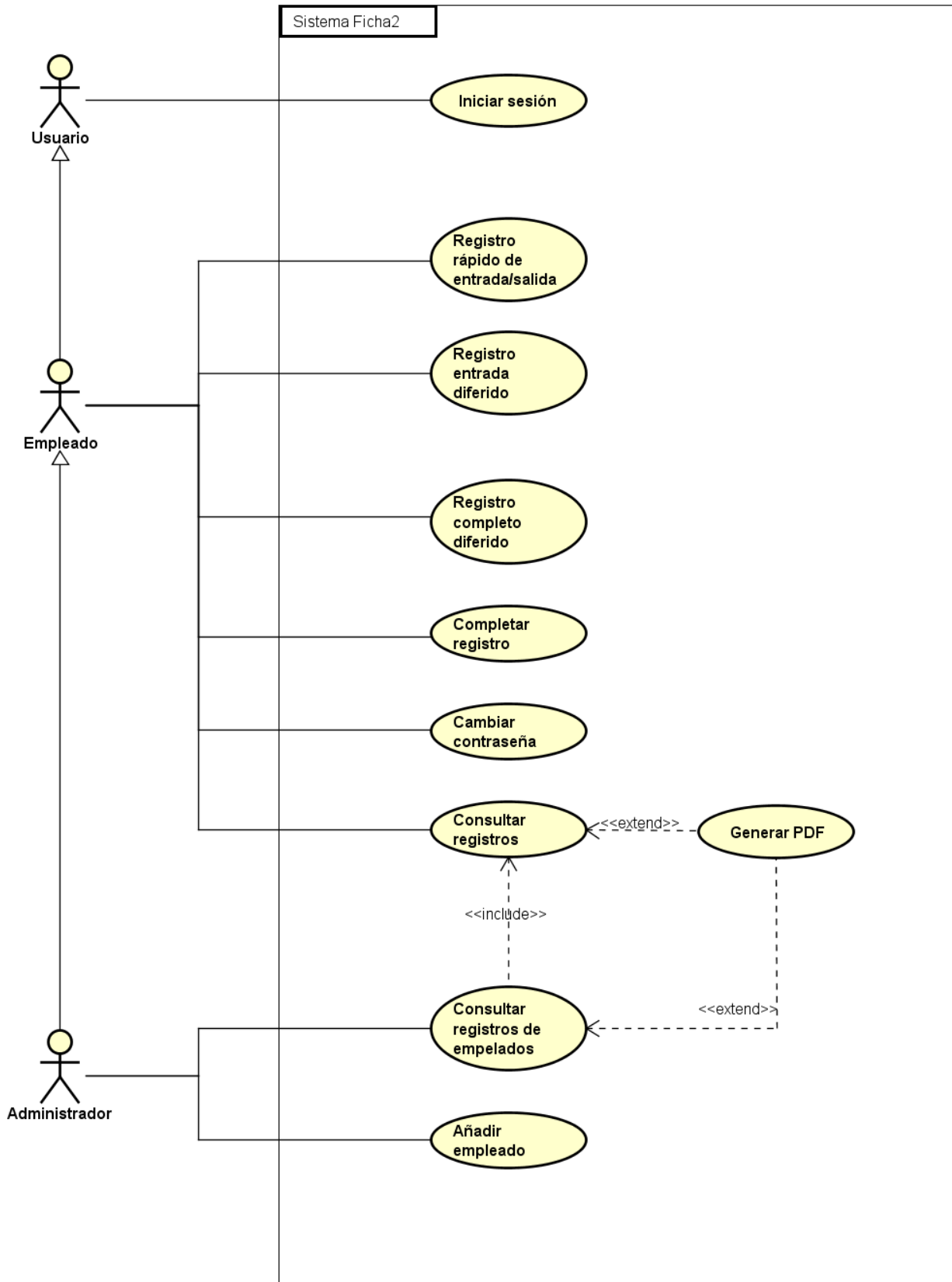


Figura 11 Diagrama de casos de uso

3.2.1 Descripción CU Registro entrada diferido

Nombre	Registro entrada diferido
Actores	Empleado
Precondición	El "Empleado" tiene que haber iniciado sesión previamente.
Postcondición	Se ha generado un registro con solo la hora de entrada indicada.
Flujo normal	<ol style="list-style-type: none"> 1. El actor "Empleado" accede a la sección Registro Manual. 2. El sistema muestra que no existe registro con solo la entrada. 3. El actor "Empleado" introduce la fecha y la hora de entrada. 4. El sistema verifica la fecha y la hora introducida. 5. El sistema crea un registro con la fecha y hora de entrada indicadas y muestra un mensaje de que el registro ha sido creado. 6. El sistema limpia los campos de fecha y hora.
Flujo alternativo	<ol style="list-style-type: none"> 2.1. Si existe un registro con solo la hora de entrada, entonces el sistema muestra dicho registro y el caso de uso continúa en el paso 3. 4.1. Si la hora de entrada se solapa con otro registro del empleado, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 3. 4.2. Si la fecha es la actual pero la hora de entrada es posterior a la actual, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 3.
Excepciones	<ol style="list-style-type: none"> 4.1. Si ya existe un registro con solo la hora de entrada, entonces el sistema muestra un mensaje de error y el caso de uso queda sin efecto. 4.2. Si la fecha es anterior al día actual, entonces el sistema muestra un mensaje de error indicando que se debe realizar el CU "Registro completo diferido", el caso de uso queda sin efecto.

3.2.2 Descripción CU Registro completo diferido

Nombre	Registro completo diferido
Actores	Empleado
Precondición	El "Empleado" tiene que haber iniciado sesión previamente.
Postcondición	Se ha generado un registro con la hora de entrada y salida indicadas.
Flujo normal	<ol style="list-style-type: none"> 1. El actor "Empleado" accede a la sección Registro Manual. 2. El sistema muestra que no existe registro con solo la entrada. 3. El actor "Empleado" introduce la fecha y las horas de entrada y salida. 4. El sistema verifica la fecha y las horas introducidas. 5. El sistema crea un registro con la fecha y horas de entrada y salida indicadas y muestra un mensaje de que el registro ha sido creado. 6. El sistema limpia los campos de fecha y horas.
Flujo alternativo	<ol style="list-style-type: none"> 2.1. Si existe un registro con solo la hora de entrada, entonces el sistema muestra dicho registro y el caso de uso continúa en el paso 3. 4.1. Si la hora de entrada es posterior a la hora de salida, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 3. 4.2. Si el registro se solapa con otro registro del empleado, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 3. 4.3. Si la fecha es la actual pero la hora de entrada o salida es posterior a la actual, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 3.
Excepciones	

3.2.3 Descripción CU Completar registro

Nombre	Completar registro
Actores	Empleado
Precondición	El "Empleado" tiene que haber iniciado sesión previamente. El "Empleado" debe tener un registro con solo la hora de entrada especificada.
Postcondición	Se ha modificado el registro que estaba sin completar indicando la hora de salida.
Flujo normal	<ol style="list-style-type: none"> 1. El actor "Empleado" accede a la sección Registro Manual. 2. El sistema muestra que existe registro con solo la entrada. 3. El actor "Empleado" selecciona "Completar registro pendiente" 4. El sistema rellena los campos de fecha y hora de entrada. 5. El actor "Empleado" introduce la hora de salida. 6. El sistema verifica la fecha y las horas introducidas. 7. El sistema modifica el registro añadiendo la hora de salida y muestra un mensaje de registro completado. 8. El sistema limpia los campos de fecha y horas.
Flujo alternativo	<ol style="list-style-type: none"> 6.1. Si la hora de salida es anterior a la hora de entrada, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 5. 6.2. Si la hora de salida se solapa con otro registro, entonces el sistema muestra un mensaje de error y el caso de uso continúa en el paso 5.
Excepciones	2.1. Si no existe un registro con solo la hora de entrada, entonces el caso de uso finaliza sin efecto.

3.2.4 Descripción CU Consultar registros

Nombre	Consultar registros
Actores	Empleado
Precondición	El "Empleado" tiene que haber iniciado sesión previamente.
Postcondición	El "Empleado" visualiza sus registros realizados en el intervalo indicado.
Flujo normal	<ol style="list-style-type: none"> 1. El actor "Empleado" accede a la sección Mis Registros. 2. El sistema muestra los intervalos de tiempo de los que el "Empleado" puede visualizar sus registros. 3. El actor "Empleado" selecciona el intervalo de tiempo del que quiere visualizar sus registros. 4. El sistema muestra los registros del intervalo de tiempo seleccionado.
Flujo alternativo	4.1. Si el actor "Empleado" quiere descargar los registros en formato PDF, entonces se realiza el caso de uso Generar PDF
Excepciones	

3.2.5 Descripción CU Consultar registros de empleados

Nombre	Consultar registros de empleados
Actores	Administrador
Precondición	El "Administrador" tiene que haber iniciado sesión previamente.
Postcondición	El "Administrador" visualiza los registros realizados en el intervalo indicado de un empleado.
Flujo normal	<ol style="list-style-type: none"> 1. Se realiza el caso de uso Consultar registros. 2. El sistema muestra una lista con los empleados registrados en el sistema. 3. El actor "Administrador" selecciona un empleado de la lista. 4. El sistema muestra los registros del empleado seleccionado en el intervalo indicado.
Flujo alternativo	4.1. Si el actor "Administrador" quiere descargar los registros en formato PDF, entonces se realiza el caso de uso Generar PDF
Excepciones	

3.2.6 Descripción CU Generar PDF

Nombre	Generar PDF
Actores	Empleado
Precondición	El "Empleado" tiene que haber iniciado sesión previamente. Se ha tenido que realizar previamente el CU Consultar Registros o Consultar registros de empleados .
Postcondición	Se ha generado un PDF con los registros mostrados en el CU realizado en la precondición.
Flujo normal	<ol style="list-style-type: none">1. El actor "Empleado" selecciona descargar PDF.2. El sistema genera el PDF con los registros realizados en el intervalo seleccionado.3. El sistema guarda el PDF en el dispositivo del actor "Empleado" y da la opción de visualizarlo.
Flujo alternativo	<ol style="list-style-type: none">3.1. Si el actor "Empleado" quiere visualizar el PDF, entonces selecciona la opción de visualizar y el caso de uso termina.
Excepciones	

3.3 Diagramas de actividad

En este apartado se muestran los diagramas de actividad de dos de los casos de uso más significativos y completos, **Registro completo diferido** y **Consultar registros de empleados** generando el PDF.

3.3.1 Registro completo diferido

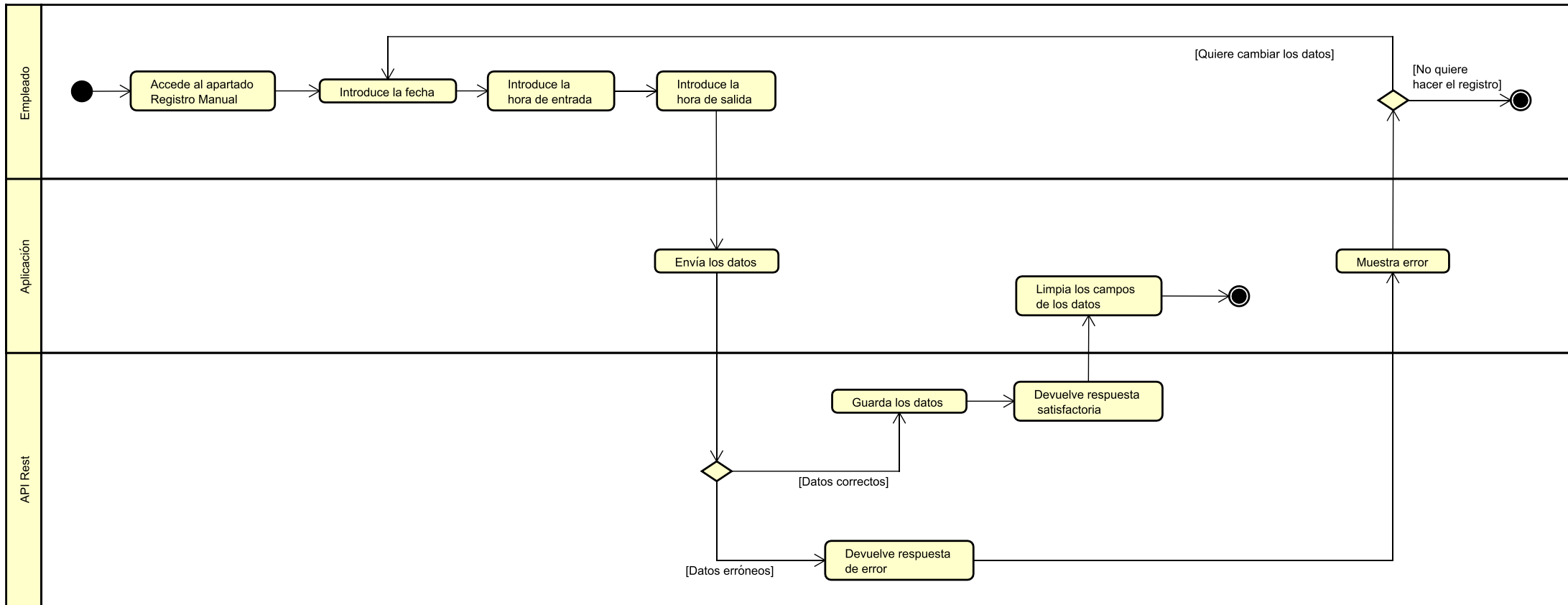


Figura 12 Diagrama de actividad - Registro completo diferido

3.3.2 Consultar registros de empleado generando un PDF

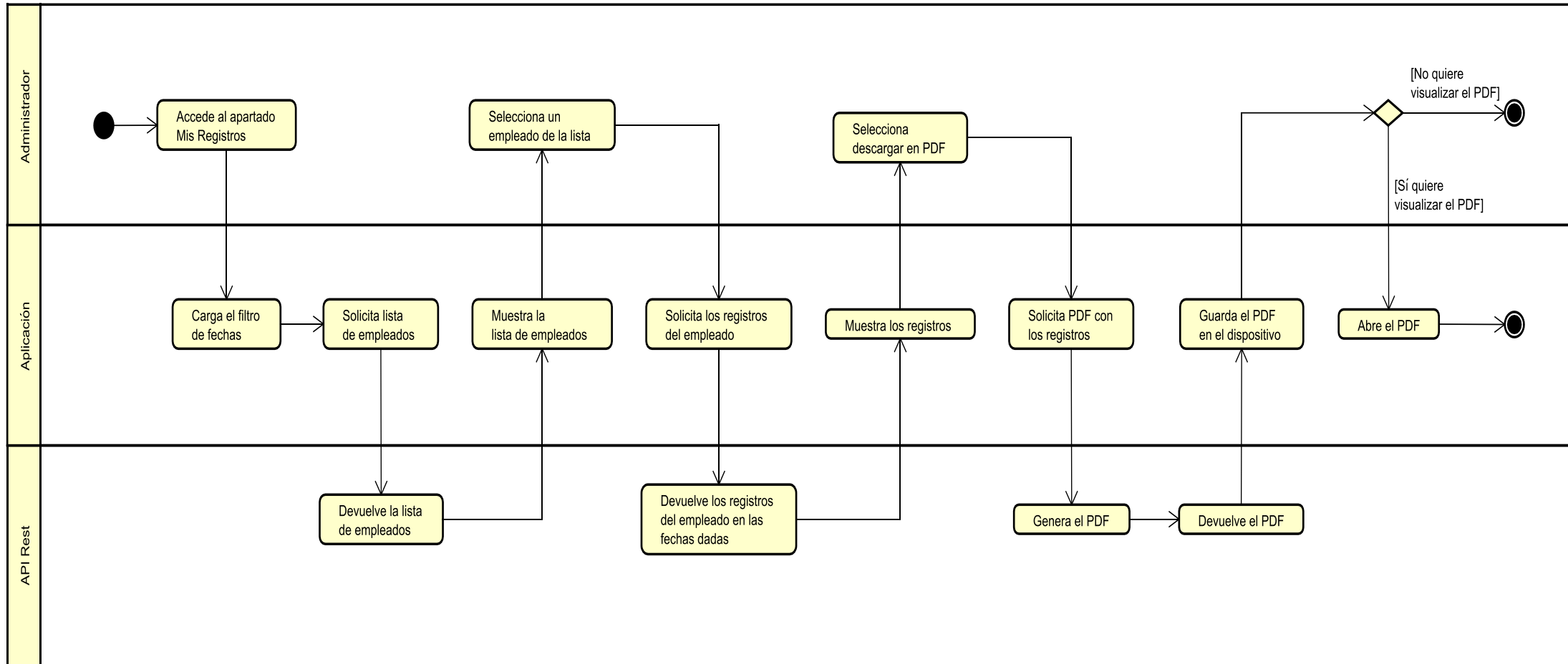


Figura 13 Diagrama de actividad - Consultar registros de empleado generando un PDF

3.4 Historias de usuario

Las historias de usuario describen una funcionalidad tal y como la percibe el usuario final [17].

Los campos son:

- **Id:** identificador único de la historia de usuario
- **Nombre:** nombre breve y descriptivo de la historia de usuario.
- **Prioridad:** nivel de prioridad usado para determinar el orden de desarrollo.
- **Descripción:** breve descripción de la historia de usuario. El lenguaje utilizado es el coloquial, aunque es común que tengan el siguiente formato

Como [rol en el sistema] quiero [objetivo]

- **Validación:** pruebas de aceptación mínima que el usuario

Historia de usuario			
Id	HU-1	Nombre	Registrar entrada-salida
Prioridad	Alta		
Descripción	Como empleado quiero poder fichar tanto a la entrada como a la salida de mi puesto de trabajo.		
Validación	- Quiero poder fichar mediante un acceso rápido, tanto para entrar como para salir.		

Historia de usuario			
Id	HU-2	Nombre	Registro diferido
Prioridad	Alta		
Descripción	Como empleado quiero poder realizar un registro diferido de mi entrada y/o salida.		
Validación	- Quiero poder realizar el registro diferido especificando el día, la hora de entrada y la de salida. - Quiero poder realizar el registro diferido de mi entrada solamente. - Quiero poder realizar el registro diferido de mi salida para completar el registro de la entrada que quedó sin completar.		

Historia de usuario			
Id	HU-3	Nombre	Consultar registros
Prioridad	Alta		
Descripción	Como empleado quiero poder consultar mis registros realizados.		
Validación	- Quiero poder visualizar mis registros del día actual. - Quiero poder consultar mis registros de días pasados.		

Historia de usuario			
Id	HU-4	Nombre	Consultar registros de empleados
Prioridad	Alta		
Descripción	Como administrador quiero poder consultar los registros realizados por los empleados.		
Validación	- Quiero que solo los administradores puedan consultar los registros de otros empleados.		

Historia de usuario			
Id	HU-5	Nombre	Descargar mi informe de registros
Prioridad	Media		
Descripción	Como empleado quiero poder descargar un informe con mis registros de entrada-salida.		
Validación	- Quiero que el informe sea un documento PDF.		

Historia de usuario			
Id	HU-6	Nombre	Descargar informe de empleados
Prioridad	Media		
Descripción	Como administrador quiero poder descargar un informe con los registros de entrada-salida de cualquier empleado.		
Validación	- Quiero que el informe sea un documento PDF.		

Historia de usuario			
Id	HU-7	Nombre	Iniciar sesión
Prioridad	Media		
Descripción	Como empleado quiero poder iniciar sesión.		
Validación	- Quiero iniciar sesión mediante usuario y contraseña.		

Historia de usuario			
Id	HU-8	Nombre	Cambiar contraseña
Prioridad	Baja		
Descripción	Como empleado quiero poder cambiar la contraseña de mi cuenta.		
Validación			

Historia de usuario			
Id	HU-9	Nombre	Añadir empleados
Prioridad	Baja		
Descripción	Como administrador quiero poder añadir nuevos empleados.		
Validación	- Quiero indicar si el nuevo empleado es administrador o no.		

Historia de usuario			
Id	HU-10	Nombre	Online
Prioridad	Media		
Descripción	Como empleado quiero conectarme a la aplicación desde cualquier lugar.		
Validación			

3.5 Tareas

Las historias de usuario se van a descomponer en diferentes tareas a realizar.

Tarea			
Id	T-01	HU-Id	Inicial
Descripción	Aprendizaje de la tecnología a usar y configuración inicial. <ul style="list-style-type: none"> - Backend - API <ul style="list-style-type: none"> o Instalación de NodeJs o Instalación de MongoDB o Módulos iniciales en <i>package.json</i> - Frontend – Aplicación Android <ul style="list-style-type: none"> o Instalación de Android Studio o Instalación de la librería Retrofit2 		

Tarea			
Id	T-02	HU-Id	HU-1
Descripción	Implementación de los modelos, controladores y rutas API para realizar el registro de entrada y salida. <ul style="list-style-type: none"> - Implementación de los modelos <i>Employee</i> y <i>Record</i>. - Implementación del controlador <i>Record</i>. - Implementación de las rutas para <i>Record</i>. 		

Tarea			
Id	T-03	HU-Id	HU-1
Descripción	Diseño e implementación de las vistas para la aplicación Android. <ul style="list-style-type: none"> - Diseño e implementación de la vista principal con el acceso rápido para fichar. 		

Tarea			
Id	T-04	HU-Id	HU-1
Descripción	Implementación del consumo de la API desde la aplicación Android para realizar el registro de entrada-salida desde el acceso rápido.		

Tarea			
Id	T-05	HU-Id	HU-2
Descripción	Implementación de las rutas API para realizar el registro diferido de entrada y/o salida.		

Tarea			
Id	T-06	HU-Id	HU-2
Descripción	Diseño e implementación de la vista de la aplicación Android para realizar el registro diferido de entrada y/o salida.		

Tarea			
Id	T-07	HU-Id	HU-2
Descripción	Conectar la vista de la aplicación Android con la API para realizar el registro diferido.		

Tarea			
Id	T-08	HU-Id	HU-3
Descripción	Implementación de las rutas API para que un empleado consulte sus registros.		

Tarea			
Id	T-09	HU-Id	HU-3
Descripción	Diseño e implementación de la vista de la aplicación Android donde un empleado consulta sus registros realizados.		

Tarea			
Id	T-10	HU-Id	HU-3
Descripción	Conectar la vista de la aplicación Android con la API para que un empleado obtenga sus registros realizados.		

Tarea			
Id	T-11	HU-Id	HU-4
Descripción	Modificar la implementación de las rutas API para que un administrador pueda consultar los registros de los empleados.		

Tarea			
Id	T-12	HU-Id	HU-4
Descripción	Añadir en la vista de la tarea T-09 la opción de que administrador pueda seleccionar el empleado del que quiere consultar los registros.		

Tarea			
Id	T-13	HU-Id	HU-5
Descripción	Implementación de la ruta API para que un empleado pueda descargar un PDF con sus registros.		

Tarea			
Id	T-14	HU-Id	HU-5
Descripción	Añadir un botón en la vista de la tarea T-09 para descargar el PDF con los registros que se están consultando.		

Tarea			
Id	T-15	HU-Id	HU-5
Descripción	Conectar el botón de la tarea T-14 con la ruta API.		

Tarea			
Id	T-16	HU-Id	HU-5
Descripción	Guardar en la memoria del teléfono móvil el PDF		

Tarea			
Id	T-17	HU-Id	HU-5
Descripción	Dar la opción de abrir el PDF una vez descargado el fichero.		

Tarea			
Id	T-18	HU-Id	HU-6
Descripción	Habilitar la ruta de la tarea T-13 para que un administrador pueda descargar un PDF con los registros de cualquier empleado.		

Tarea			
Id	T-19	HU-Id	HU-7
Descripción	Implementación del middleware para la autenticación mediante JWT.		

Tarea			
Id	T-20	HU-Id	HU-7
Descripción	Implementación de la ruta API para obtener el token JWT tras ingresar usuario y contraseña.		

Tarea			
Id	T-21	HU-Id	HU-7
Descripción	Diseño e implementación de la vista de la aplicación Android para que un empleado introduzca su usuario y contraseña.		

Tarea			
Id	T-22	HU-Id	HU-7
Descripción	Conectar la vista de la tarea T-19 con la ruta API implementada en la tarea T-18.		

Tarea			
Id	T-23	HU-Id	HU-8
Descripción	Implementación de la ruta API para que un empleado pueda cambiar su contraseña.		

Tarea			
Id	T-24	HU-Id	HU-8
Descripción	Diseño e implementación de la vista de la aplicación Android para que un empleado introduzca su actual contraseña y la nueva.		

Tarea			
Id	T-25	HU-Id	HU-8
Descripción	Conectar la vista de la tarea T-22 con la ruta API de la tarea T-21.		

Tarea			
Id	T-26	HU-Id	HU-9
Descripción	Implementación de la ruta API para que un administrador pueda añadir nuevos empleados.		

Tarea			
Id	T-27	HU-Id	HU-9
Descripción	Diseño e implementación de la vista de la aplicación Android para que un administrador introduzca los datos del nuevo empleado.		

Tarea			
Id	T-28	HU-Id	HU-9
Descripción	Conectar la vista de la tarea T-25 con la ruta API de la tarea T-24.		

Tarea			
Id	T-29	HU-Id	HU-10
Descripción	Crear base de datos en Atlas MongoDB		

Tarea			
Id	T-30	HU-Id	HU-10
Descripción	Configurar y desplegar API en Heroku		

Tarea			
Id	T-31	HU-Id	HU-10
Descripción	Actualizar ruta de la API en la aplicación Android.		

4 Planificación

4.1 Scrum

Definido por Hirotaka Takeuchi e Ikujiro Nonaka en 1986, Scrum es un macro de trabajo ágil formado por roles, artefactos y eventos, con el cual las personas pueden abordar proyectos complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor posible. El proyecto se divide en ciclos de duración fija denominados sprints. En cada ciclo se produce una parte del producto terminada y funcionalmente operativa.

4.1.1 Roles

- Scrum Máster: es la persona responsable de apoyar al equipo de desarrollo, eliminar las barreras organizativas y mantener la consistencia del proyecto ágil [18].
- Product Owner: es la persona implicada en establecer un puente entre el usuario final, los responsables del negocio y el equipo de desarrollo, maximizando el valor del negocio para el proyecto. Trabaja día a día para clarificar los requisitos.
- Equipo de desarrollo o equipo Scrum: el grupo de personas que trabajan en la creación del producto. Son los diseñadores, programadores, testers, etc. Tienen la autoridad para decidir cómo organizarse para cumplir con los objetivos de un Sprint. Típicamente es un equipo de entre 5 y 10 personas.
- Stakeholders: es un término colectivo que incluye a clientes, usuario y patrocinadores, los cuales interactúan principalmente con el Product Owner, Scrum Máster y el equipo de desarrollo para proporcionarles las entradas y facilitar la creación del producto.

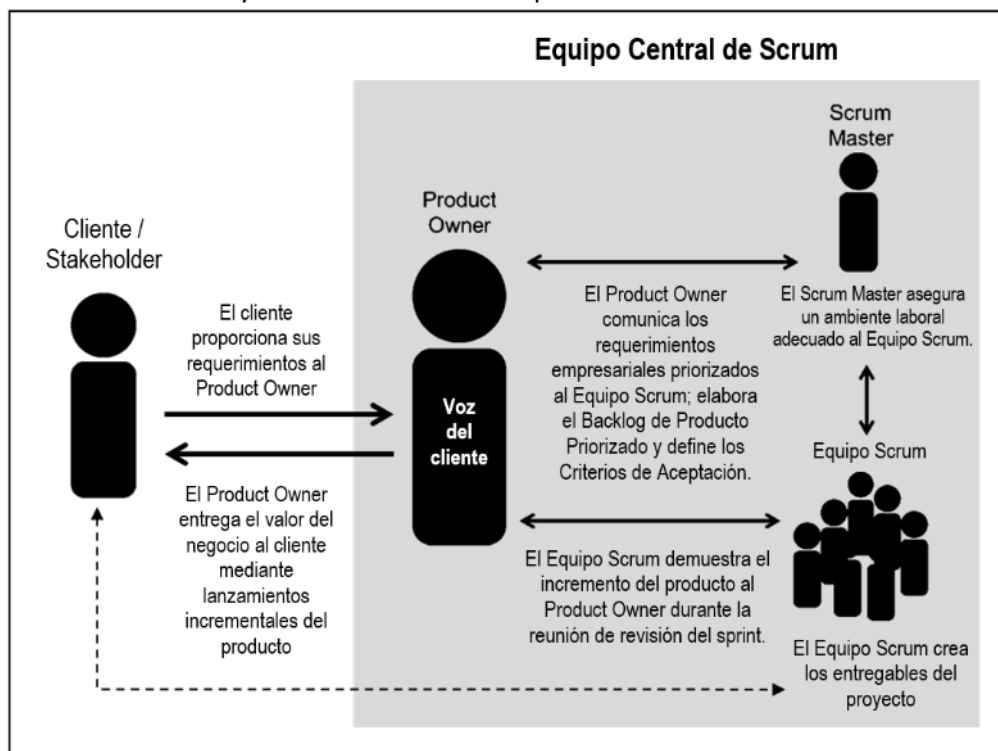


Figura 14 Roles en la metodología Scrum

Aunque en Scrum no se recomienda que una misma persona adopte todos los roles, al ser un proyecto de carácter formativo, he desempeñado la función de los diferentes roles.

4.1.2 Artefactos

Los artefactos representan el trabajo o el valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto. A continuación se detallan los principales.

- **Product Backlog:** o Pila del Producto, es la lista ordenada de las historias de usuario que se han de incorporar al producto mediante los Sprints. Representa todo aquello que el Cliente/Stakeholder espera del producto. La pila nunca está completa, evoluciona a medida que el producto y el entorno en el que se usará lo hacen. En las primeras etapas del proyecto, la pila refleja los requerimientos conocidos y mejor entendidos al principio.
- **Sprint Backlog:** o Pila del Sprint, es la lista de tareas que debe realizar el equipo de desarrollo durante el Sprint para construir un Incremento: una parte completa y operativa del producto. Esta pila la realiza el equipo durante la reunión de planificación del Sprint, autoasignando cada tarea a un miembro del equipo, e indicando en la misma lista cuánto tiempo o esfuerzo prevé que falta para terminarla. De esta forma, el Sprint Backlog descompone el trabajo en unidades de tamaño adecuado para monitorizar el avance a diario e identificar riesgos o problemas sin necesidad de procesos de gestión complejos.
- **Incremento:** es la funcionalidad del producto completa y en funcionamiento al final de cada sprint que es potencialmente entregable: terminada y probada. Si el proyecto requiere documentación, o procesos de validación y verificación documentados, éstos también tienen que estar realizados para considerar que el incremento está hecho.

4.1.3 Eventos

Los eventos en Scrum tienen la finalidad de crear una regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son periodos de tiempo limitados, de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse ni alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento.

- **Sprint:** la pieza clave de Scrum, es un periodo de tiempo durante el cual se crea un incremento del producto terminado y utilizable. Su duración es prefijada al comienzo del proyecto y no debe modificarse. Normalmente la duración es de una o dos semanas. Durante el Sprint, no se realizan cambios que puedan poner en peligro el objetivo del Sprint, los objetivos de calidad no disminuyen y el alcance pueden clarificarse y renegociarse entre el Product Owner y el equipo de desarrollo a medida que se va aprendiendo más.
- **Sprint Planning:** o Planificación del Sprint, es una reunión que se realiza al comienzo de cada Sprint donde se fijan los objetivos del mismo. Se identifican los requisitos y las tareas a realizar para completar cada requisito.
- **Daily Scrum:** es una reunión de no más de 15 minutos realizada cada día del Sprint donde los miembros del equipo de desarrollo indican lo realizado el día anterior y planifican lo que se va a realizar durante el día actual.
- **Sprint Review:** o Revisión del Sprint, es la reunión informal que se realiza al final de cada Sprint, iniciada por el Product Owner, donde se analiza la funcionalidad del producto conseguida durante el Sprint.
- **Sprint Retrospective:** o Retrospectiva del Sprint. Es otra reunión realizada al final de cada Sprint donde todo el equipo Scrum analiza qué fue bien, qué debe cambiar y cómo realizar los cambios necesarios. Su duración es normalmente entre 15 y 30 minutos.

4.2 Product Backlog

El Product Backlog es la lista ordenada por prioridades de las funcionalidades que se van a desarrollar a lo largo del proyecto.

Id	Como	Quiero poder...	Prioridad
HU-1	Empleado	Registrar mi entrada y salida de mi puesto de trabajo.	Alta
HU-2	Empleado	Realizar un registro diferido de mi entrada y/o salida.	Alta
HU-3	Empleado	Consultar mis registros de entrada y salida.	Alta
HU-4	Administrador	Consultar los registros de entrada y salida de los empleados.	Alta
HU-5	Empleado	Descargar un PDF con mis registros de entrada-salida.	Media
HU-6	Administrador	Descargar un PDF con los registros de entrada-salida de los empleados.	Media
HU-7	Empleado	Identificarme en la aplicación.	Media
HU-8	Empleado	Cambiar mi contraseña de inicio de sesión.	Baja
HU-9	Administrador	Añadir nuevos empleados y administradores.	Baja
HU-10	Empleado	Conectarme desde cualquier sitio.	Media

4.3 Sprints

Como se describió anteriormente, el Sprint Backlog es la lista de tareas a realizarse durante el Sprint, proveniente del desglose de las historias de usuario que forman el Product Backlog.

El desarrollo de este proyecto se ha planificado mediante Sprints de una semana de duración, con un total de 5 Sprints.

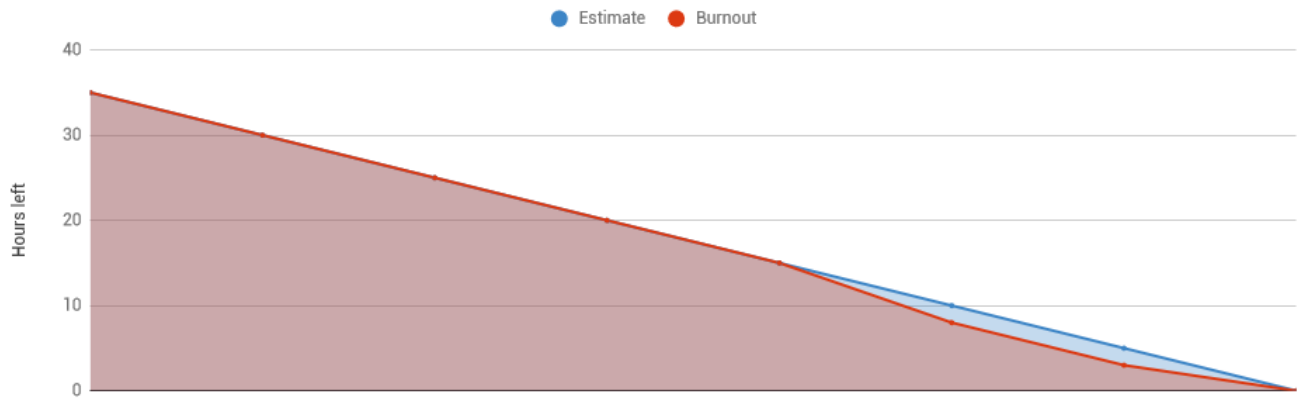
4.3.1 Sprint 1

Durante el primer Sprint se realizaron unas primeras tareas relacionadas con el aprendizaje de la tecnología, configuración del entorno y la implementación de las historias de usuario HU-7 y HU-9. En la Tabla 1 Tareas Sprint 1 se muestra la evolución del trabajo durante este primer Sprint. En la primera columna se indican las tareas que se van a realizar durante el Sprint, en la segunda columna las horas estimadas a emplear por cada tarea. En la fila *Tiempo diario estimado (horas)* se indica las horas totales estimadas a dedicar cada día al proyecto. Al final de cada día, se fue rellenando las celdas (en amarillo) con las horas reales dedicadas a cada tarea en ese día.

Podemos observar, que tanto la tarea T-01 como la T-17 fueron ligeramente subestimadas. Sin embargo, gracias a dedicar un poco más de tiempo al aprendizaje, se consiguió suplir esta diferencia con lo estimado en las tareas T-24 y T-18, terminando el Sprint con las todas las tareas terminadas y completadas.

Tarea	Tiempo estimado (horas)	Tiempo empleado (horas)	Tiempo restante (horas)	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado	Domingo
(T-01) Aprendizaje de la tecnología	9	12	-3	5	5	2				
(T-26) Ruta API añadir empleados	5	3	2			3				
(T-27) Vistas Android añadir empleados	5	5	0					5		
(T-28) Conectar APP Android con la API	2	2	0					2		
(T-19) Middleware JWT	2	3	-1				2			1
(T-20) Ruta API login	5	3	2				3			
(T-21) Vista Android login	5	5	0						5	
(T-22) Conectar vista login con API login	2	2	0							2
TOTAL	35	35	0	5	5	5	5	7	5	3
Tiempo diario estimado (horas)	0			5	5	5	5	5	5	5
Tiempo total restante estimado (horas)			35	30	25	20	15	10	5	0
Tiempo total empleado (horas)			35	30	25	20	15	8	3	0

Tabla 1 Tareas Sprint 1

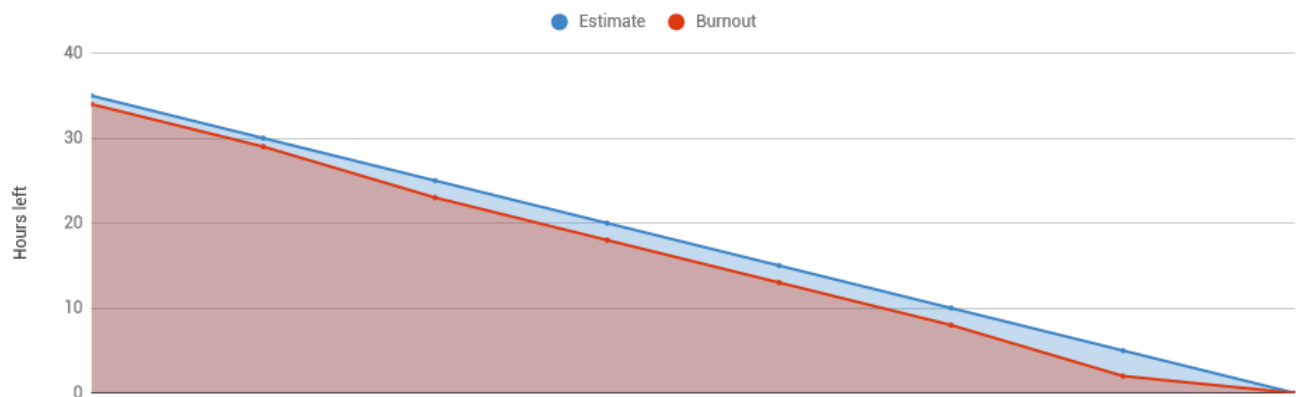


Gráfica 1 Burndown del Sprint 1

4.3.2 Sprint 2

Tarea	Tiempo estimado (horas)	Tiempo empleado (horas)	Tiempo restante (horas)	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado	Domingo
(T-02) Ruta API registro entrada-salida	7	6	1	3	3					
(T-03) Vistas Android registro entrada-salida rápido	4	3	1	2	1					
(T-04) Conectar APP Android con la API registro entrada-salida rápido	2	2	0		2					
(T-05) Ruta API registro entrada-salida diferido	13	13	0			5	5		3	
(T-06) Vista Android registro entrada-salida diferido	7	8	-1					5	3	
(T-07) Conectar APP Android con la API registro entrada-salida diferido	2	2	0							2
TOTAL	35	34	1	5	6	5	5	5	6	2
Tiempo diario estimado (horas)	0			5	5	5	5	5	5	5
Tiempo total restante estimado (horas)			35	30	25	20	15	10	5	0
Tiempo total empleado (horas)			34	29	23	18	13	8	2	0

Tabla 2 Tareas Sprint 2

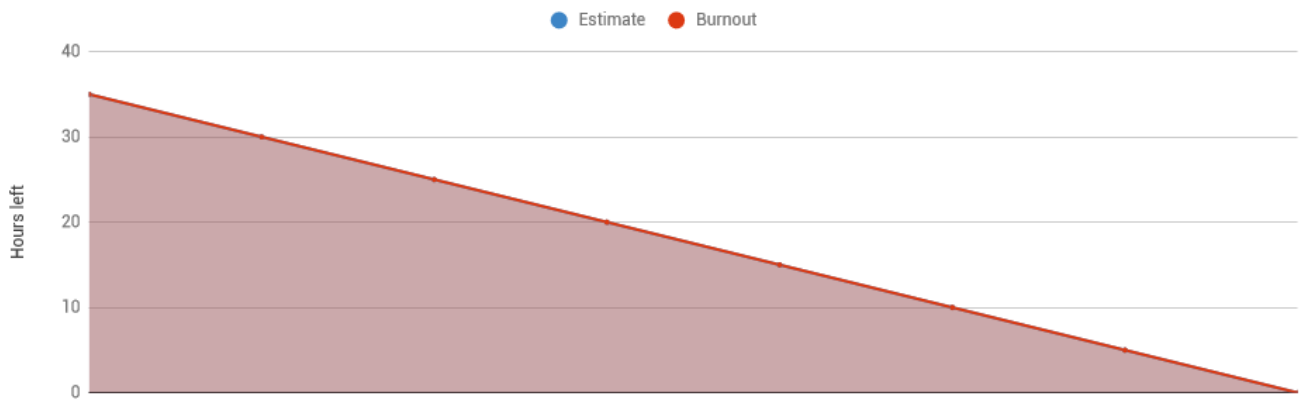


Gráfica 2 Burndown del Sprint 2

4.3.3 Sprint 3

Tarea	Tiempo estimado (horas)	Tiempo empleado (horas)	Tiempo restante (horas)	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado	Domingo
(T-08) Ruta API consultar mis registros	10	10	0	5	5					
(T-09) Vistas Android consultar mis registros	4	4	0			4				
(T-10) Conectar APP Android con la API consultar mis registros	1	1	0			1				
(T-11) Actualizar API para que admin consulte registros	15	15	0				5	5	4	1
(T-12) Actualizar vista Android para que admin consulte registros	5	5	0						1	4
TOTAL	35	35	0	5	5	5	5	5	5	5
Tiempo diario estimado (horas)	0			5	5	5	5	5	5	5
Tiempo total restante estimado (horas)			35	30	25	20	15	10	5	0
Tiempo total empleado (horas)			35	30	25	20	15	10	5	0

Tabla 3 Tareas Sprint 3

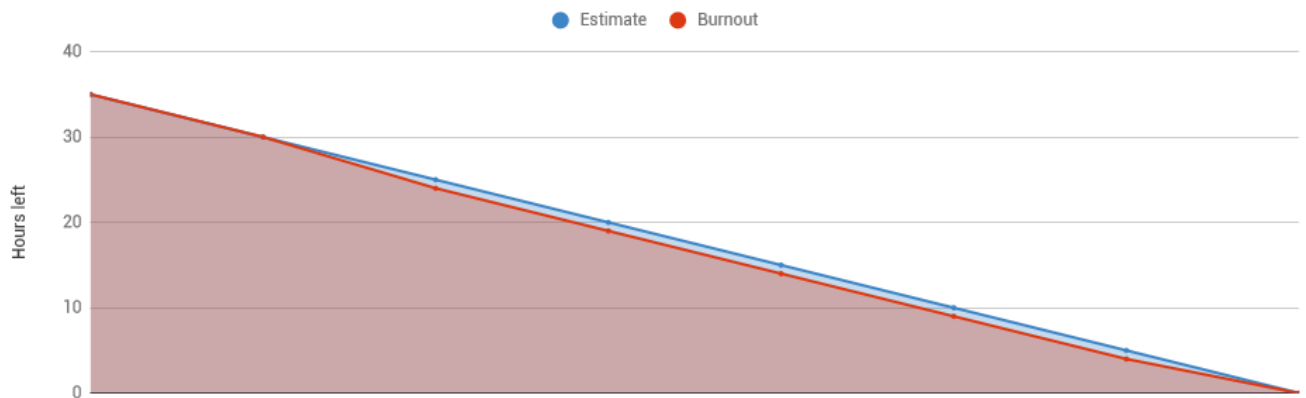


Gráfica 3 Burndown del Sprint 3

4.3.4 Sprint 4

Tarea	Tiempo estimado (horas)	Tiempo empleado (horas)	Tiempo restante (horas)	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
(T-13) Ruta API descargar mis registros en PDF	14	13	1	5	5	3				
(T-14) Actualizar vista Android para descargar mis registros en PDF	1	1	0		1					
(T-15) Conectar APP Android con la API descargar mis registros en PDF	3	3	0			2	1			
(T-16) Guardar PDF en Android	10	11	-1				4	5	2	
(T-17) Mostrar PDF en Android	5	5	0						3	2
(T-18) Actualizar API para admin descargar registros en PDF	2	2	0							2
TOTAL	35	35	0	5	6	5	5	5	5	4
Tiempo diario estimado (horas)	0			5	5	5	5	5	5	5
Tiempo total restante estimado (horas)			35	30	25	20	15	10	5	0
Tiempo total empleado (horas)			35	30	24	19	14	9	4	0

Tabla 4 Tareas Sprint 4

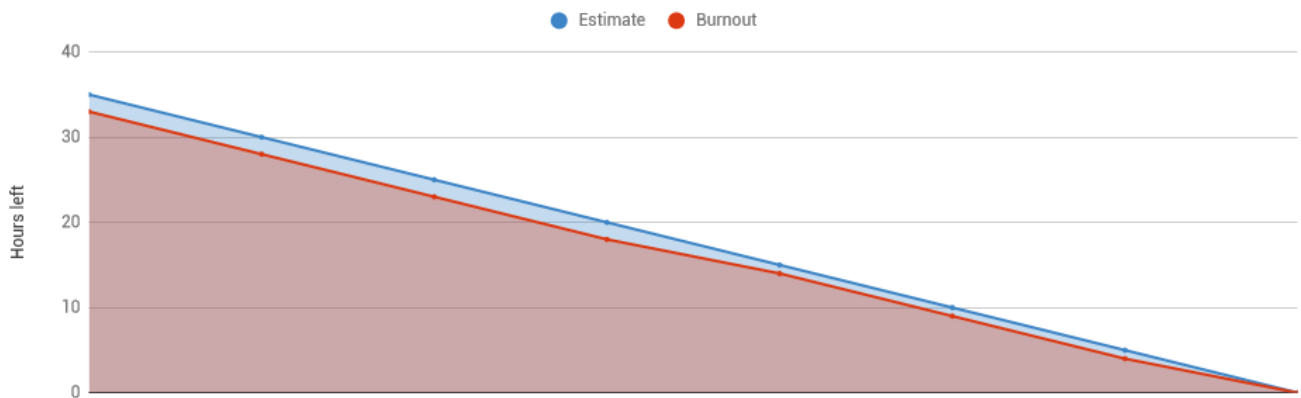


Gráfica 4 Burndown del Sprint 4

4.3.5 Sprint 5

Tarea	Tiempo estimado (horas)	Tiempo empleado (horas)	Tiempo restante (horas)	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado	Domingo
(T-23) Ruta API cambiar mi contraseña	8	8	0	5	3					
(T-24) Vista Android cambiar mi contraseña	5	5	0			3	2			
(T-25) Conectar APP Android con la API cambiar contraseña	2	2	0				2			
(T-29) Crear base datos en Atlas MongoDB	7	6	1		2	2		2		
(T-30) Configurar y desplegar API Heroku	10	10	0					3	5	2
(T-31) Actualizar ruta de la API en Android	3	2	1							2
TOTAL	35	33	2	5	5	5	4	5	5	4
Tiempo diario estimado (horas)	0			5	5	5	5	5	5	5
Tiempo total restante estimado (horas)			35	30	25	20	15	10	5	0
Tiempo total empleado (horas)			33	28	23	18	14	9	4	0

Tabla 5 Tareas Sprint 5



Gráfica 5 Burndown del Sprint 5

5 Ejecución del proyecto

5.1 Arquitectura

Para el desarrollo del sistema se ha optado por emplear la arquitectura de API REST, la cual se apoya totalmente en las peticiones HTTP para el manejo de la información. De esta manera, el sistema consta de dos partes bien diferenciadas:

- Backend: servidor encargado de procesar las peticiones HTTP que le lleguen a través de la API REST.
- Frontend: la aplicación Android la cual realizará dichas peticiones HTTP sobre el backend.

5.1.1 Características de la arquitectura REST

REST es cualquier interfaz entre sistemas que use HTTP como protocolo para obtener datos o generar operaciones sobre esos datos. Las características que definen la arquitectura REST son:

- Protocolo cliente/servidor sin estado, es decir, cada petición HTTP contiene toda la información necesaria para ejecutarla, permitiendo así que tanto el cliente como el servidor no tengan que recordar ningún estado previo para satisfacer dicha petición.
- Para realizar las peticiones HTTP se usan los verbos GET, POST, PUT y DELETE para el acceso, creación, actualización y borrado de datos respectivamente.
- La URI es el identificador único de cada recurso de un sistema REST. Esto facilita el acceso a la información para poder modificarla o eliminarla.

5.1.2 Ventajas de la arquitectura REST

El uso de una arquitectura REST para el desarrollo de un sistema software proporciona las siguientes ventajas principalmente:

- Separación entre el cliente y el servidor. Gracias a la arquitectura REST se tiene totalmente separada la interfaz de usuario del servidor y el almacenamiento de datos. Esto proporciona una gran libertad a la hora de cambiar la interfaz de usuario a otro tipo de plataforma, además de que los distintos componentes puedan evolucionar de forma independiente.
- La API REST siempre es independiente de la tecnología y lenguaje, pudiendo usarse la misma API en aplicaciones muy diferentes. Lo único indispensable es que las respuestas a las peticiones se hagan siempre con el mismo lenguaje de intercambio de información, usualmente XML o JSON.

5.2 Patrones de diseño

Los patrones de diseño nos proporcionan una solución general ante un problema común. Es una plantilla sobre cómo resolver un problema que se puede utilizar en muchas situaciones diferentes.

En este proyecto se han utilizado principalmente tres patrones: DAO, DTO y Façade.

5.2.1 DAO

El patrón DAO propone separar por completo la lógica de negocio de la lógica para acceder a los datos. De esta forma, el DAO proporciona los métodos necesarios para insertar, actualizar, borrar y consultar la información, mientras que la capa de negocio utiliza el DAO para interactuar con la fuente de datos [\[19\]](#).

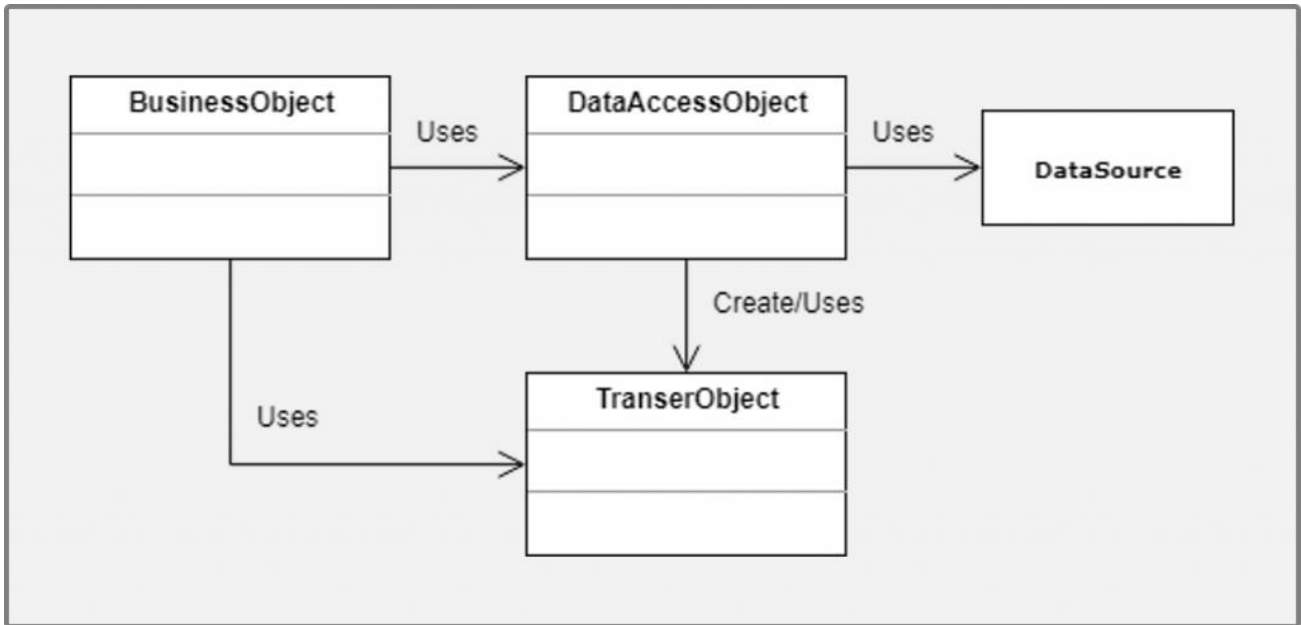


Figura 15 Diagrama del patrón DAO

5.2.2 DTO

Muy ligado al patrón DAO, el patrón DTO tiene por finalidad el crear un objeto plato (POJO) con una serie de atributos que puedan ser enviados o recuperados del servidor en una sola invocación, de tal forma que un DTO puede contener información de múltiples fuentes y concentrarlas en una única clase simple. Debe cumplir dos características principales: debe ser serializable y de solo lectura, es decir, no debe contener operaciones relacionadas con la lógica del negocio o métodos que realicen cálculos sobre los datos [\[20\]](#).

5.2.3 Façade

Se trata de un patrón estructural, el cual proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. La fachada define una interfaz de alto nivel que hace el subsistema más fácil de usar, minimizando la comunicación y dependencias entre subsistemas [21].

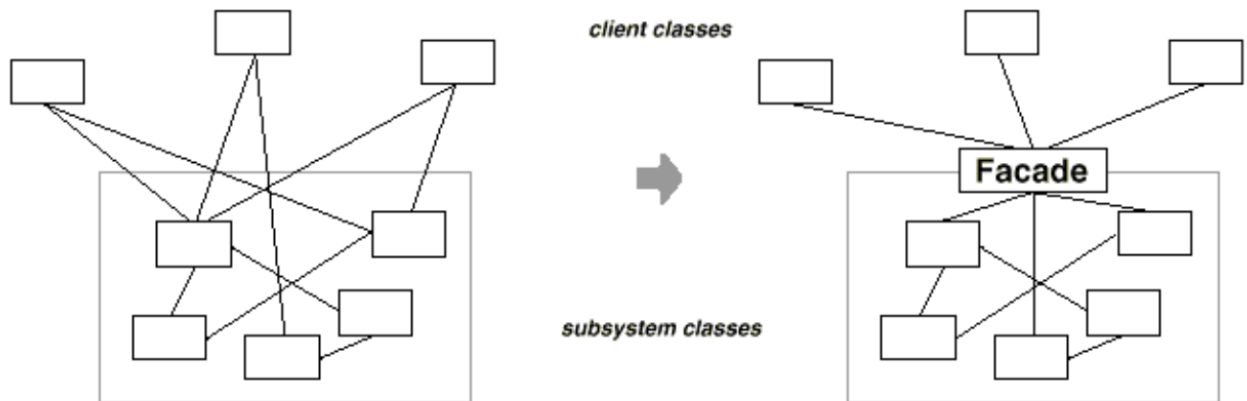


Figura 16 Patrón Façade

5.3 Despliegue

Como el sistema consta de una aplicación Android que toma y genera datos en una API, además de generar el APK de la aplicación Android debemos desplegar nuestra API.

Se ha utilizado la plataforma Heroku para el despliegue de la API, además de la plataforma Atlas MongoDB donde tener alojada la base de datos.

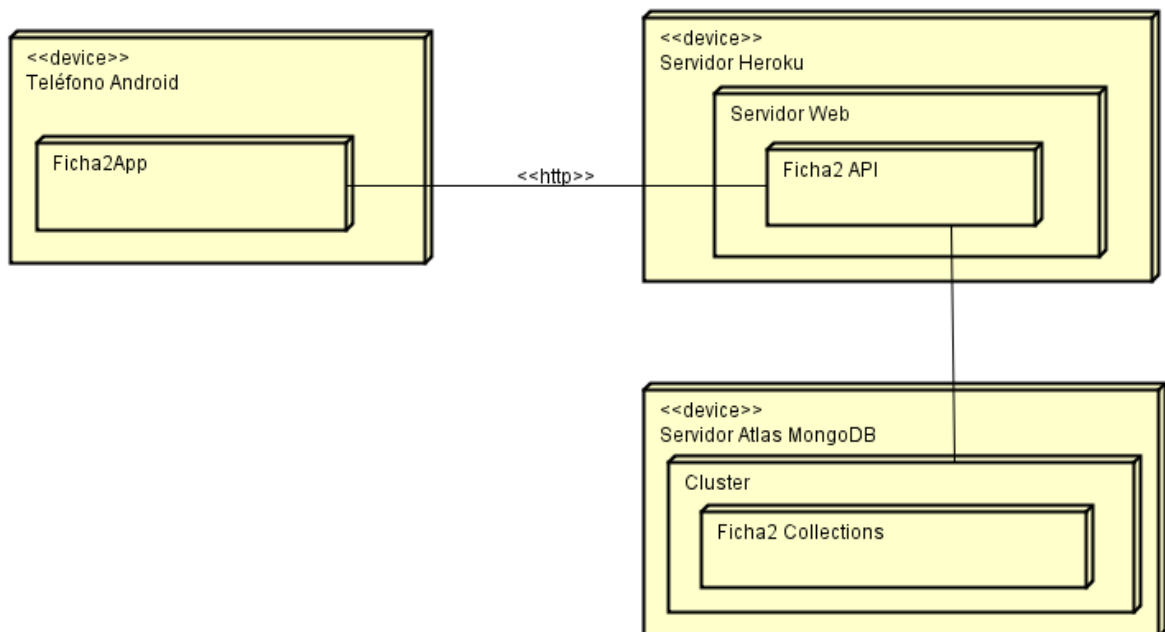


Figura 17 Diagrama de despliegue del sistema

5.4 Diseño de la base de datos

Con MongoDB se puede integrar datos relacionados en una sola estructura o documento, haciendo que la aplicación pueda realizar menos peticiones a la base de datos para obtener la información necesaria. Este tipo de esquemas con datos integrados se los conoce como “desnormalizados” [22].



Figura 18 Ejemplo de esquema "desnormalizado" en MongoDB

Sin embargo, con MongoDB se pueden tener también esquemas normalizados, es decir, esquemas en los que las relaciones entre estructuras se realizan mediante referencias entre los documentos.

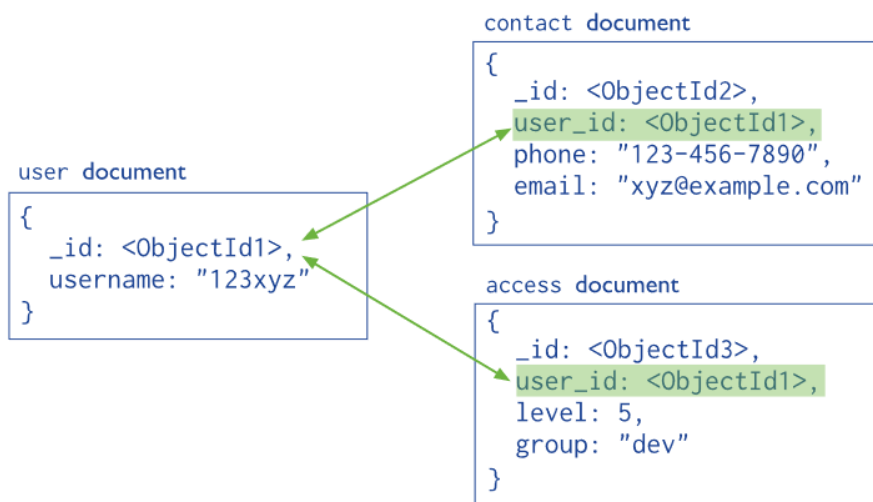


Figura 19 Ejemplo del esquema normalizado en MongoDB

Aunque para este proyecto se podía haber usado la forma “desnormalizada” mediante datos integrados, finalmente se optó por emplear las referencias al considerar que el beneficio de la reducción de las peticiones no era suficiente frente al inconveniente de tener duplicada información de los usuarios.

A continuación, se muestra el modelo Entidad-Relación una vez finalizado el proyecto.

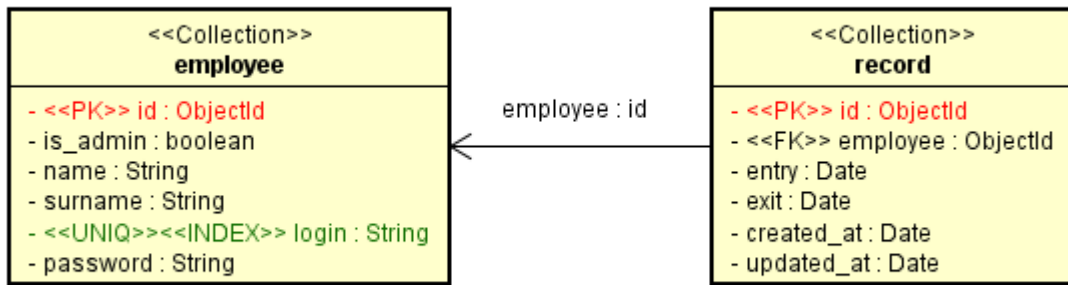


Figura 20 Diagrama entidad-relación del sistema

5.5 Diagramas de secuencia

A continuación, se muestra la secuencia normal de los dos principales casos de uso. En ambos casos, la petición HTTP pasa por el middleware *auth.middleware.js* antes que los Controller para identificar el usuario que está realizando la petición mediante el token JWT enviado en la cabecera Authorization.

5.5.1 Registro completo diferido

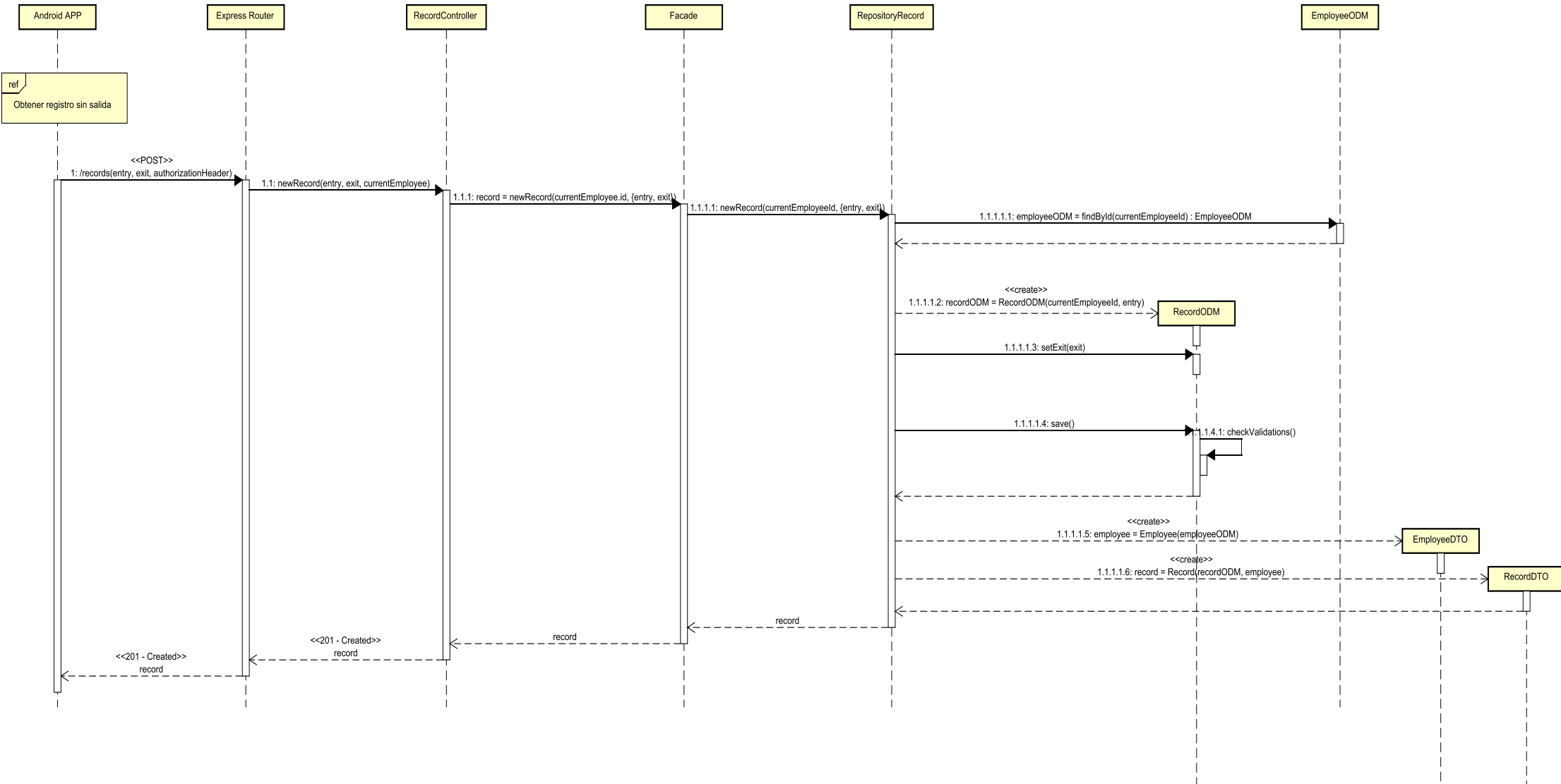


Figura 21 Diagrama de secuencia - Registro completo diferido

5.5.1.1 Obtener registro sin salida

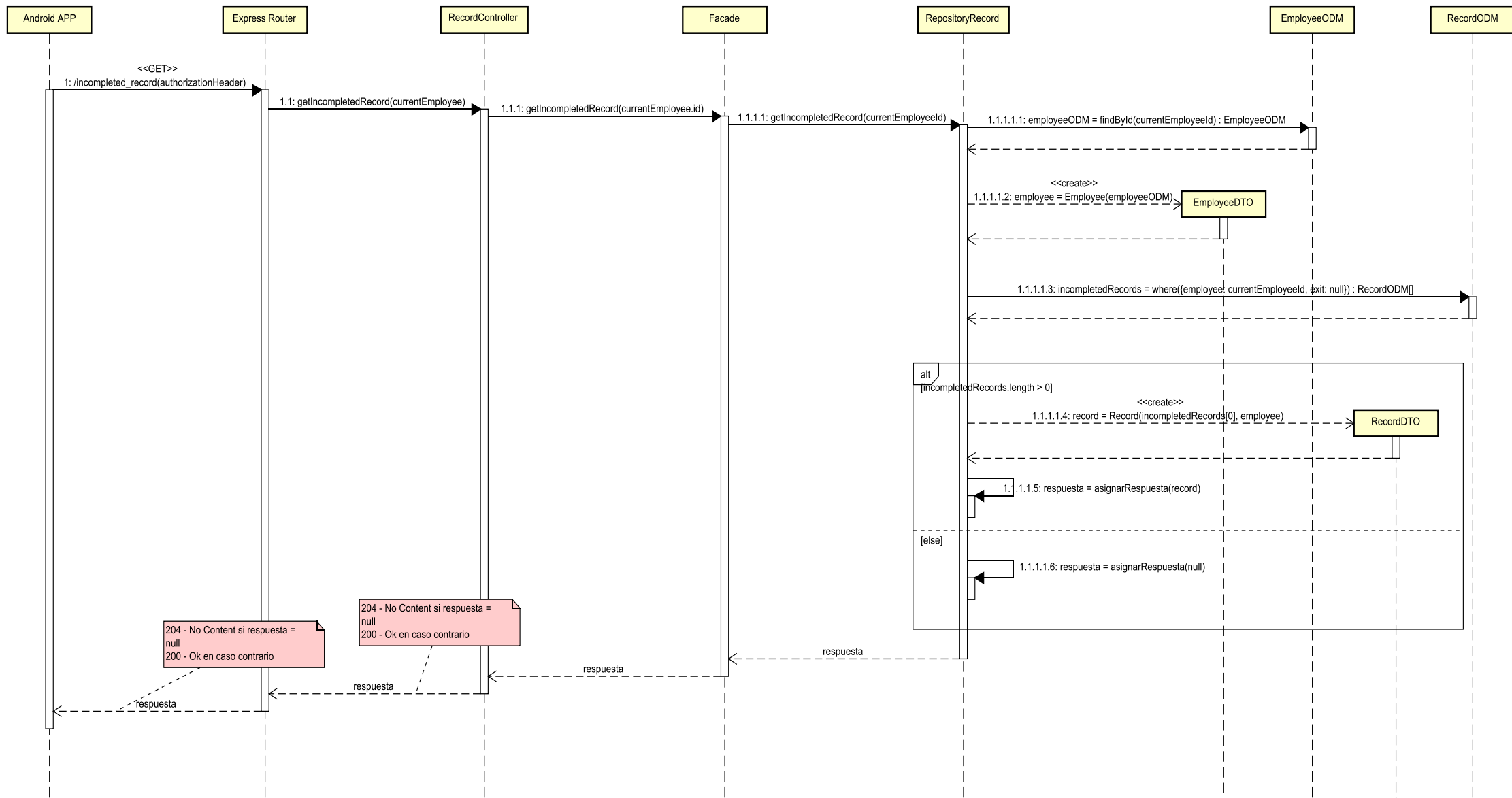


Figura 22 Diagrama de secuencia - Obtener registro sin salida

5.5.2 Obtener registros de un empleado

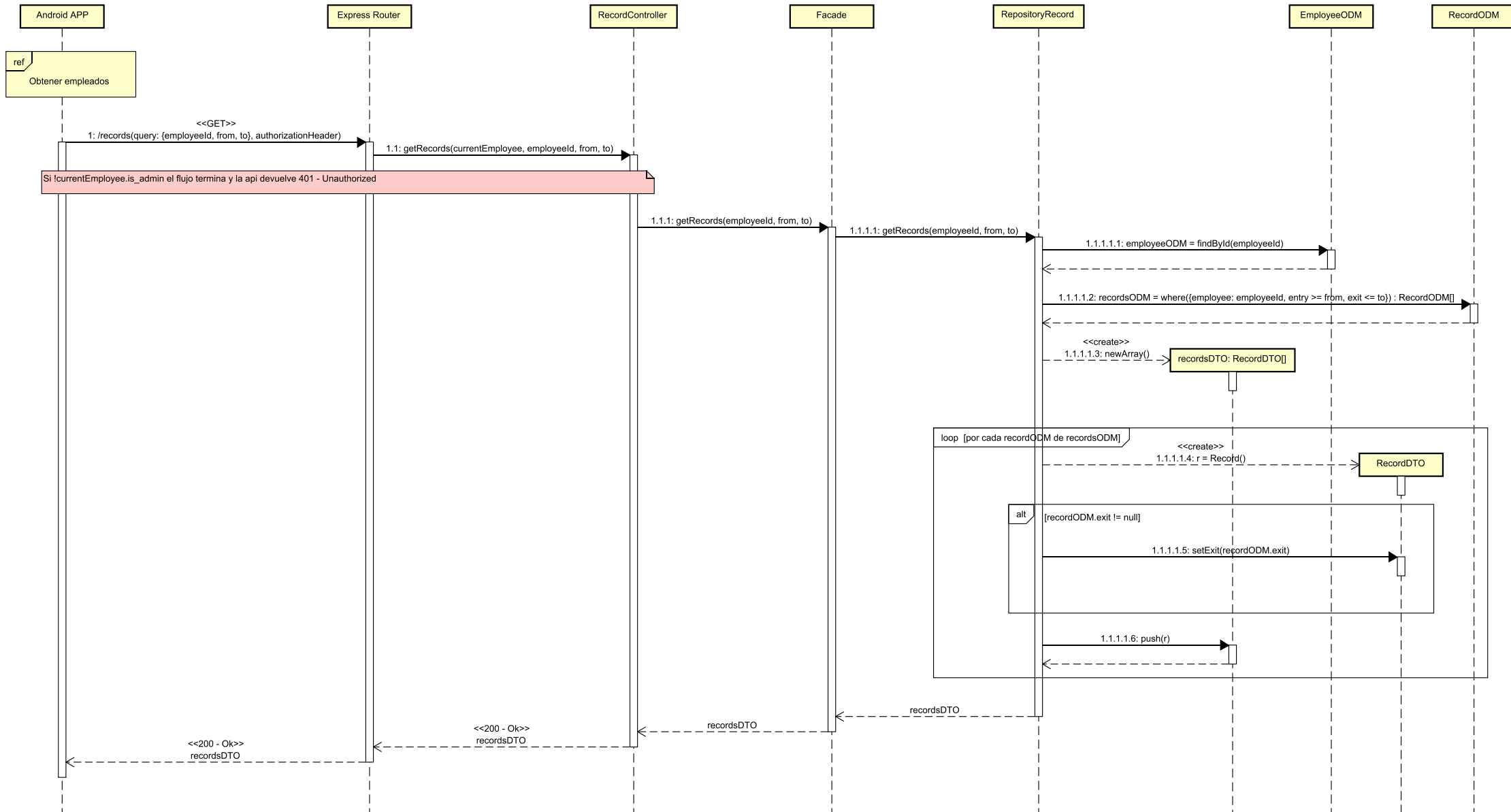


Figura 23 Diagrama de secuencia - Obtener registros de un empleado

5.5.2.1 Obtener empleados

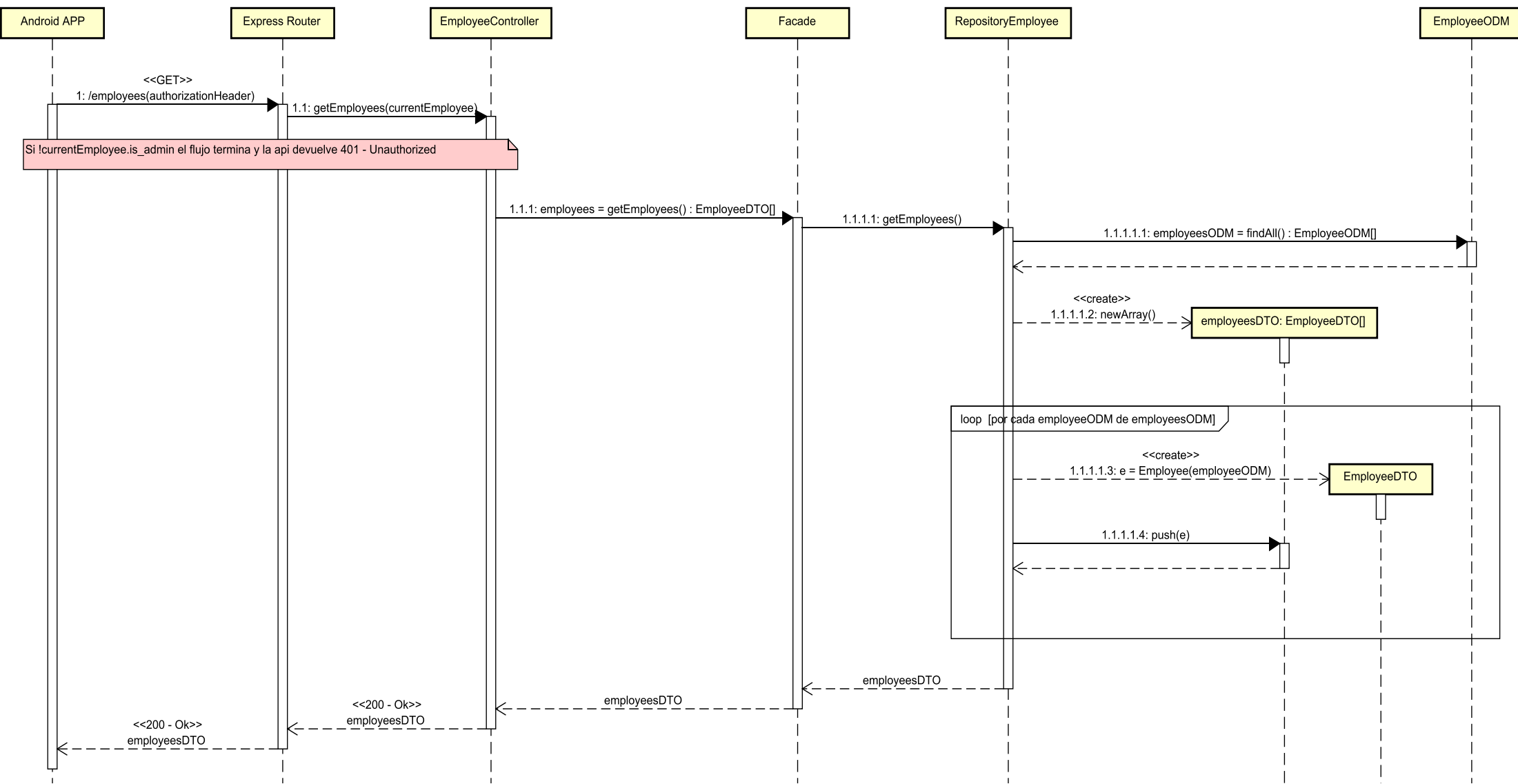


Figura 24 Diagrama de secuencia - Obtener empleados

5.6 Interfaces gráficas

En este apartado, se mostrarán las vistas de la aplicación Android implementadas.

5.6.1 Splash screen

Esta es la vista que se muestra mientras se inicia la aplicación.



Figura 25 Splash Screen

5.6.2 Login

En esta vista el usuario podrá iniciar sesión introduciendo su usuario y contraseña.



Figura 26 Vista de iniciar sesión

5.6.3 Vista principal o de acceso rápido

Esta es la vista que se muestra al usuario una vez ha iniciado sesión. En ella, el usuario puede visualizar los registros realizados en el día actual ordenadas por hora de entrada. Además, tiene un botón para realizar el registro de entrada/salida para el momento actual. Este botón cambiará de color dependiendo de la acción que se vaya a realizar, rojo para registrar una salida, y verde para registrar una entrada. También, este botón estará desactivado si existe un registro de entrada realizado en un día pasado el cual no tenga su correspondiente salida.

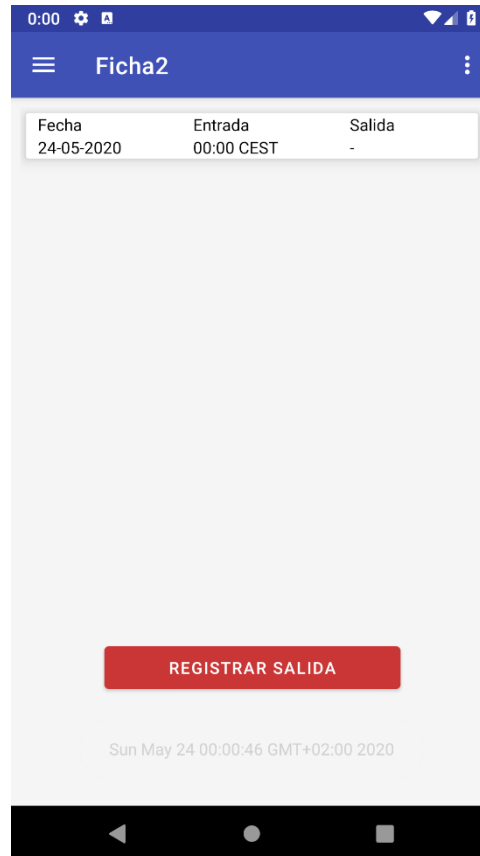


Figura 27 Vista del acceso rápido al registro de entrada-salida

5.6.4 Vista del registro diferido de entrada-salida

En la vista del registro diferido, el usuario podrá:

- Visualizar si tiene algún registro sin completar y completarlo. En la parte superior, justo encima de **Fecha**, se muestra si hay o no un registro incompleto. En caso de haberlo, el selector de **Completar registro pendiente** se podrá activar, rellenando los campos de **Fecha** y **Entrada**. Una vez activado el selector e introducida la hora de salida, el registro se completará.
- Registrar de forma diferida la hora de entrada. Para ello, el usuario deberá rellenar los campos de **Fecha** y **Entrada**.
- Realizar un registro completo de forma diferida. Para ello, el usuario deberá rellenar todos los campos (**Fecha**, **Entrada** y **Salida**).

Fecha	Entrada	Salida
24-05-2020	00:00 CEST	-

Fecha

Entrada
Zona horaria: CEST

Salida
Zona horaria: CEST

ENVIAR

Figura 28 Vista del registro diferido de entrada-salida

5.6.5 Vista para consultar registros

Un usuario que no sea administrador, en la sección **Consultar registros** visualizará la siguiente pantalla. En ella podrá seleccionar en qué rango de tiempo quiere consultar sus registros, además de poder descargar el informe en PDF. Por defecto, los registros se agrupan por día, mostrando las horas totales trabajadas en el día. Para poder ver los detalles de cada día, el usuario deberá pulsar el día del que quiere ver los registros, visualizándose tal y como se muestra en la imagen.

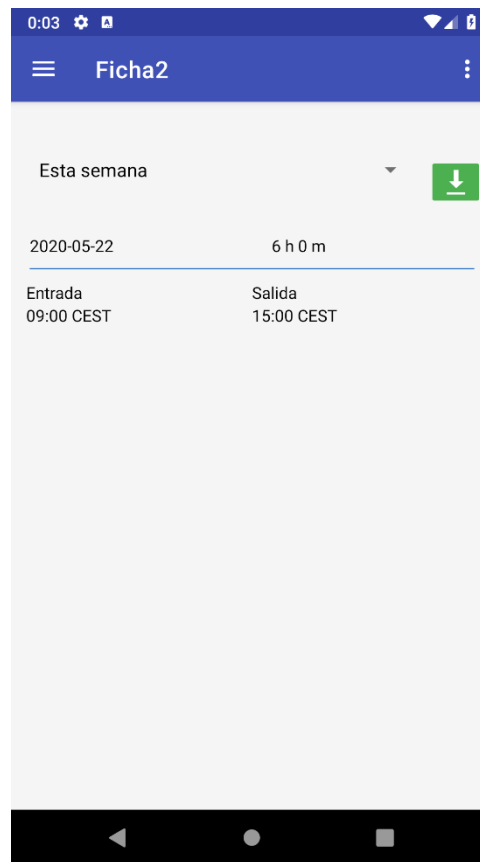


Figura 29 Vista para consultar mis registros

5.6.6 Vista cambio de contraseña

En este apartado, el usuario podrá modificar su contraseña, debiendo introducir su contraseña actual junto con la nueva y su duplicado.



The screenshot shows a mobile application interface for changing a password. At the top, there is a blue header bar with the text "Ficha2" and a menu icon on the left and a three-dot menu icon on the right. Below the header, the title "Nueva contraseña" is displayed in a large, bold, black font. The main content area is light gray and contains three input fields, each with a label and a toggle icon (an eye) to the right of the input line. The labels are "Contraseña actual", "Contraseña", and "Repetir contraseña". Below the input fields, there is a blue button with the text "ENVIAR" in white, bold, uppercase letters. At the bottom of the screen, there is a black navigation bar with three icons: a back arrow, a home circle, and a recent apps square.

Figura 30 Vista para cambiar la contraseña

5.6.7 Vista de un administrador para consultar registros

Un usuario que sea administrador, además de poder realizar las acciones descritas en el apartado 0, podrá seleccionar de qué usuario quiere consultar los registros, además de poder descargar dichos registros en formato PDF.

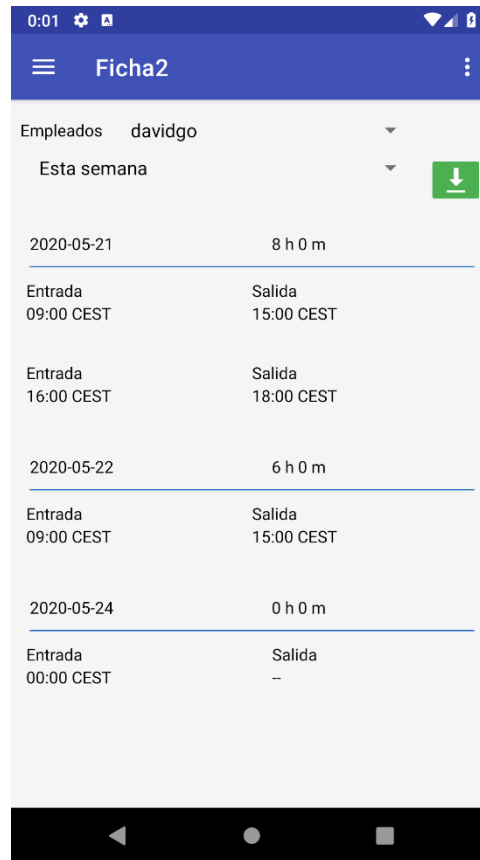
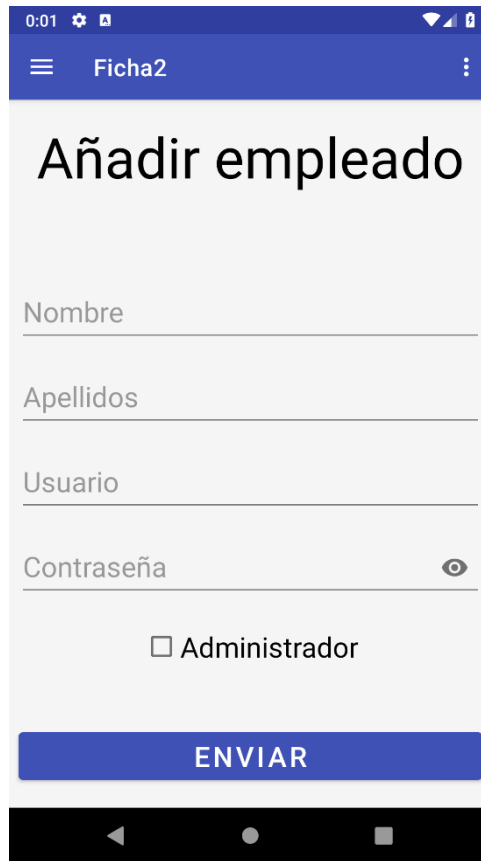


Figura 31 Vista de un administrador para consultar los registros realizados

5.6.8 Vista de un administrador para añadir empleado

Un usuario que sea administrador tendrá acceso esta vista desde donde podrá añadir un nuevo usuario al sistema. Para ello, deberá rellenar los campos, pudiendo indicar si el nuevo usuario será o no un nuevo administrador.



0:01


Ficha2

Añadir empleado

Nombre

Apellidos

Usuario

Contraseña 

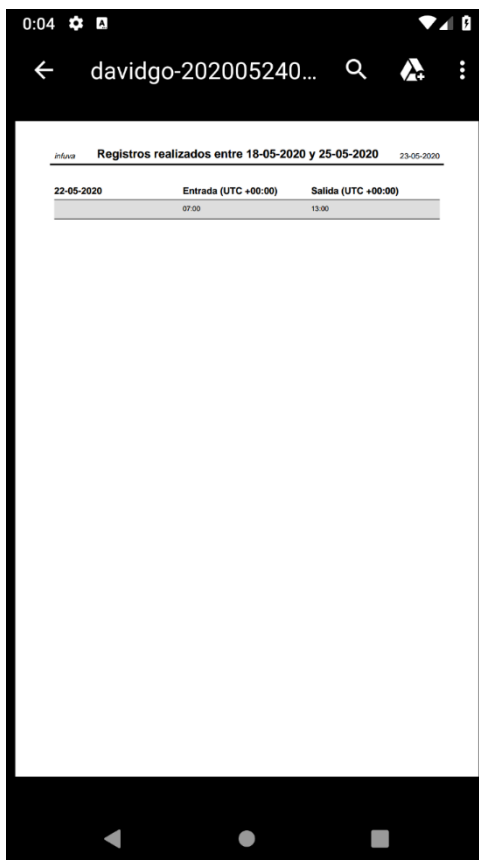
Administrador

ENVIAR

Figura 32 Vista de un administrador para añadir nuevo usuario

5.6.9 Ejemplo de PDF con los registros realizados

A continuación se muestra un ejemplo de un PDF que se generaría al pulsar la opción de descargar en la sección de **Consultar registros**. En la esquina superior izquierda se indica el usuario a quien pertenecen los registros, en la esquina superior derecha la fecha de creación del PDF. Todas las horas mostradas en el PDF se indican en formato UTC.



The screenshot shows a mobile application interface. At the top, the status bar displays the time 0:04 and various icons. Below the status bar, there is a navigation bar with a back arrow, the text 'davidgo-202005240...', a search icon, a refresh icon, and a menu icon. The main content area displays a PDF document with the following table:

Registros realizados entre 18-05-2020 y 25-05-2020		
	Entrada (UTC +00:00)	Salida (UTC +00:00)
22-05-2020	07:00	13:00

Figura 33 Ejemplo de un PDF generado con los registros realizados

5.7 Pruebas

Durante el desarrollo de los Sprints, se fueron realizando diferentes pruebas para comprobar que las funcionalidades que se iban desarrollando funcionaban correctamente. Además, tras la finalización del último Sprint, se llevaron a cabo una serie de pruebas de usabilidad para determinar si la aplicación es lo suficientemente intuitiva para alguien sin conocimientos previos sobre la aplicación.

5.7.1 Pruebas durante los Sprints

Estas pruebas se realizaron usando la aplicación Postman y el emulador que viene integrado con Android Studio para emular el dispositivo Nexus 5X con la versión 9 de Android (API 29).

5.7.1.1 Sprint 1

Sprint 1	Prueba 1	HU-9 Añadir nuevos empleados y administradores
Descripción	Comprobación del endpoint POST /register mediante Postman, haciendo una petición con los datos necesarios sin usar un login ya existente.	
Resultado esperado	Respuesta con código 201 - Created , y la información del nuevo empleado.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 1	Prueba 2	HU-9 Añadir nuevos empleados y administradores
Descripción	Comprobación del endpoint POST /register mediante Postman, haciendo una petición con los datos necesarios, pero usando un login ya existente.	
Resultado esperado	Respuesta con código 409 - Conflict , y un mensaje informativo.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 1	Prueba 3	HU-9 Añadir nuevos empleados y administradores
Descripción	Comprobación mediante el emulador de que al pulsar sobre "Añadir empleado" y enviar los datos, el nuevo empleado ha sido creado.	
Resultado esperado	Se muestra en la aplicación un mensaje de que el empleado ha sido añadido correctamente.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 1	Prueba 4	HU-7 Identificarme en la aplicación
Descripción	Comprobación del endpoint POST /login mediante Postman, haciendo una petición con las credenciales correctas, y otra con credenciales erróneas.	
Resultado esperado	- Respuesta con código 200 - Ok y el token JWT para la petición con credenciales correctas. - Respuesta con código 401 - Unauthorized y un mensaje informativo.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 1	Prueba 5	HU-7 Identificarme en la aplicación
Descripción	Comprobación mediante el emulador de que la aplicación Android se comunica correctamente con la API enviando las credenciales.	
Resultado esperado	Si se envían unas credenciales correctas, entonces se muestra la vista principal.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

5.7.1.2 Sprint 2

Sprint 2	Prueba 6	HU-1 Registrar mi entrada/salida
Descripción	Comprobación de los endpoints POST /quick_entry y POST /quick_exit mediante Postman, haciendo las siguientes peticiones con el token JWT en la cabecera Authorization . - Primero una petición a /quick_entry , después otra a /quick_exit - Primero una petición a /quick_entry , después otra a /quick_entry - Una petición a /quick_exit (sin tener registrada una entrada previamente) - Primero una petición a /quick_entry . Al día siguiente, una petición a /quick_exit	
Resultado esperado	- Respuesta de la primera petición: 200 - Ok . Respuesta de la segunda: 200 - Ok - Respuesta de la primera petición: 200 - Ok . Respuesta de la segunda: 409 - Conflict - Respuesta de la petición: 400 - Bad Request - Respuesta de la primera petición: 200 - Ok . Respuesta de la segunda: 409 - Conflict	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 2	Prueba 7	HU-1 Registrar mi entrada/salida
Descripción	Comprobación mediante el emulador de que la aplicación Android envía las peticiones correctamente.	
Resultado esperado	Se muestra un mensaje de que el registro se ha realizado correctamente.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 2	Prueba 8	HU-2 Registrar mi entrada/salida en diferido
Descripción	Comprobación de los endpoints POST /records y POST /manual_exit mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization . 1) Una petición a /records con fecha actual y hora de entrada y salida. 2) Una petición a /records con fecha del día anterior con hora de entrada y salida. 3) Una petición a /records con fecha del día anterior con solo hora de entrada. 4) Una petición a /records con fecha del día posterior con hora de entrada y salida. 5) Teniendo un registro con solo entrada del día anterior: 5.1) Una petición a /manual_exit con fecha del día anterior y hora de salida. 5.2) Una petición a /manual_exit con fecha actual y hora de salida. 5.3) Una petición a /records con fecha actual y solo hora de entrada. 6) Teniendo un registro del día actual con entrada a las 9:00 y salida a las 14:00: 6.1) Una petición a /records con fecha actual y hora de entrada posterior a las 14:00. 6.2) Una petición a /records con fecha actual y entrada a las 13:30 y salida a las 15:00.	
Resultado esperado	1) Respuesta con código 201 - Created y el registro creado. 2) Respuesta con código 201 - Created y el registro creado. 3) Respuesta con código 400 - Bad Request . 4) Respuesta con código 400 - Bad Request . 5.1) Respuesta con código 201 - Created y el registro con la salida. 5.2) Respuesta con código 409 - Conflict . 5.3) Respuesta con código 409 - Conflict . 6.1) Respuesta con código 201 - Created . 6.2) Respuesta con código 409 - Conflict .	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 2	Prueba 9	HU-2 Registrar mi entrada/salida en diferido
Descripción	Comprobación mediante el emulador de que la aplicación Android envía las peticiones correctamente y muestra los mensajes de error cuando procede.	
Resultado esperado	Se muestran los mensajes de registro creado satisfactoriamente y los mensajes de error cuando se realiza alguno de los casos descritos en la Prueba 8 .	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

5.7.1.3 Sprint 3

Sprint 3	Prueba 10	HU-3 Consultar mis registros
Descripción	<p>Comprobación de los endpoints GET /records y GET /incompleted_record mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization.</p> <p>1) Habiendo creado registros para varios días de la semana, pero todos con hora de entrada y salida:</p> <ol style="list-style-type: none"> 1.1) Una petición a /records?from=YYYY-MM-DD&to=YYYY-MM-DD con la fecha de from anterior a la de to. 1.2) Una petición a /records?from=YYYY-MM-DD&to=YYYY-MM-DD con la fecha de from posterior a la de to. 1.3) Una petición a /records. 1.4) Una petición a /incompleted_record <p>2) Habiendo creado registros para varios días de la semana, pero dejando un registro sin hora de salida:</p> <ol style="list-style-type: none"> 2.1) Una petición /incompleted_record <p>3) Sin tener creado registros previamente:</p> <ol style="list-style-type: none"> 3.1) Una petición a /records. 	
Resultado esperado	<ol style="list-style-type: none"> 1.1) Respuesta con código 200 - Ok y la lista de registros realizados entre from y to. 1.2) Respuesta con código 400 - Bad Request. 1.3) Respuesta con código 200 - Ok y la lista de todos los registros realizados. 1.4) Respuesta con código 204 - No content. 2.1) Respuesta con código 200 - Ok y el registro que solo tiene la hora de entrada. 3.1) Respuesta con código 200 - Ok, pero vacía. 	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 3	Prueba 11	HU-3 Consultar mis registros
Descripción	Comprobación mediante el emulador que las peticiones se realizan correctamente, que en la vista principal se muestren los registros realizados durante el día actual, y que en la vista para realizar registros diferidos se muestre el único registro sin completar (si lo hubiera).	
Resultado esperado	Al realizar un nuevo registro mediante los botones "Entrada/Salida ahora", la lista de los registros del día actual se actualiza. Si hubiera un registro sin completar, se muestra en la vista donde se realizan los registros diferidos.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 3	Prueba 12	HU-4 Consultar los registros de otros empleados
Descripción	Comprobación de los endpoints GET /records y GET /employees mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization . 1) Habiendo iniciado sesión con la cuenta de un administrador: 1.1) Petición a /records?from=YYYY-MM-DD&to=YYYY-MM-DD&employee=id siendo id el identificador del empleado del que se quiere obtener sus registros. 1.2) Petición a /employees . 2) Habiendo iniciado sesión con la cuenta de un empleado normal: 2.1) Petición a /records?from=YYYY-MM-DD&to=YYYY-MM-DD&employee=id siendo id el identificador del empleado del que se quiere obtener sus registros. 2.2) Petición a /employees	
Resultado esperado	1.1) Respuesta con código 200 - Ok y la lista de registros realizados entre from y to por el empleado "id" . 1.2) Respuesta con código 200 - Ok y la lista de empleados. 2.1) Respuesta con código 401 - Unauthorized . 2.2) Respuesta con código 401 - Unauthorized .	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 3	Prueba 13	HU-4 Consultar los registros de otros empleados
Descripción	Comprobación mediante el emulador que se puede consultar los registros de un empleado siendo administrador, pero no siendo un empleado normal.	
Resultado esperado	Si se ha iniciado sesión con la cuenta de un administrador, en la vista para consultar los registros aparece la lista de los empleados. Si se ha iniciado sesión con la cuenta de un empleado normal, en la vista para consultar los registros no aparece la lista de los empleados.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

5.7.1.4 Sprint 4

Sprint 4	Prueba 14	HU-5 Descargar un PDF con mis registros
Descripción	Comprobación del endpoint GET /report mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization . 1) Habiendo creado registros para varios días de la semana, pero todos con hora de entrada y salida: 1.1) Una petición a /report?from=YYYY-MM-DD&to=YYYY-MM-DD con la fecha de from anterior a la de to . 1.2) Una petición a /report?from=YYYY-MM-DD&to=YYYY-MM-DD con la fecha de from posterior a la de to . 1.3) Una petición a /report . 2) Sin tener creados registros en el intervalo indicado: 2.1) Petición a /report?from=YYYY-MM-DD&to=YYYY-MM-DD	
Resultado esperado	1.1) Respuesta con código 200 - Ok y un fichero PDF con los registros realizados en el periodo indicado. 1.2) Respuesta con código 400 - Bad Request . 1.3) Respuesta con código 404 - Bad Request . 2.1) Respuesta con código 200 - Ok y un fichero PDF con un mensaje indicando que no existen registros en el periodo indicado.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 4	Prueba 15	HU-5 Descargar un PDF con mis registros
Descripción	Comprobación mediante el emulador que se tras acceder a la vista de consultar registros, se pueden descargar en PDF.	
Resultado esperado	Se genera un PDF con los registros realizados en el intervalo indicado en el selector de la aplicación. Además, la aplicación da la opción de visualizar el PDF una vez descargado.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 4	Prueba 16	HU-6 Descargar un PDF con los registros de otros empleados
Descripción	Comprobación del endpoint GET /report mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization . 1) Habiendo iniciado sesión con la cuenta de un administrador: 1.1) Una petición a /report?from=YYYY-MM-DD&to=YYYY-MM-DD&employee=id siendo id el identificador del empleado del que se quiere obtener sus registros. 2) Habiendo iniciado sesión con la cuenta de un empleado normal: 2.1) Petición a /report?from=YYYY-MM-DD&to=YYYY-MM-DD&employee=id siendo id el identificador del empleado del que se quiere obtener sus registros.	
Resultado esperado	1.1) Respuesta con código 200 - Ok y un fichero PDF con los registros realizados en el periodo indicado por el empleado " id ". 2.1) Respuesta con código 401 - Unauthorized .	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 4	Prueba 17	HU-6 Descargar un PDF con los registros de otros empleados
Descripción	Comprobación mediante el emulador que, tras acceder a la vista de consultar registros, se pueden descargar en PDF los registros realizados por otro empleado.	
Resultado esperado	Si se ha iniciado sesión con la cuenta de un administrador, se genera un PDF con los registros realizados en el intervalo indicado por el empleado seleccionado. Además, la aplicación da la opción de visualizar el PDF una vez descargado.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

5.7.1.5 Sprint 5

Sprint 5	Prueba 18	HU-8 Cambiar mi contraseña
Descripción	Comprobación del endpoint PUT /change_password mediante Postman, realizando las siguientes peticiones con el token JWT en la cabecera Authorization . 1) Petición a /change_password enviando la antigua contraseña, la nueva y la repetición de la nueva contraseña. 2) Petición a /change_password enviando la antigua contraseña, la nueva y la repetición sin coincidir. 3) Petición a /change_password sin enviar alguno de los tres datos.	
Resultado esperado	1) Respuesta con código 204 - No content . 2) Respuesta con código 400 - Bad Request . 3) Respuesta con código 400 - Bad Request .	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 5	Prueba 19	HU-8 Cambiar mi contraseña
Descripción	Comprobación mediante el emulador que se realiza el cambio de contraseña habiendo iniciado sesión previamente.	
Resultado esperado	Se muestra un mensaje de que la contraseña ha sido modificada en caso de que los datos introducidos sean correctos.	
Resultado obtenido	El esperado.	
Modificaciones	Ninguna.	

Sprint 5	Prueba 20	HU-10 Conectarse desde cualquier sitio
Descripción	Tras haber desplegado la API en Heroku y conectar la API con MongoDB Atlas, se intenta iniciar sesión con una de las cuentas.	
Resultado esperado	Se ha iniciado sesión correctamente.	
Resultado obtenido	Fallo de conexión. <u>Causa:</u> no se actualizó la dirección de la API a la que se tiene que conectar la aplicación.	
Modificaciones	Se modificó la variable ROOT_URL del fichero ServerURLs.java	

5.7.2 Pruebas de usabilidad

Según el estándar ISO 25000 [23], la usabilidad de un producto software es su capacidad para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en:

- **Capacidad para reconocer su adecuación.** Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- **Capacidad de aprendizaje.** Capacidad del producto que permite al usuario aprender su aplicación.
- **Capacidad de ser usado.** Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- **Protección contra errores de usuario.** Capacidad del sistema para proteger a los usuarios de cometer errores.
- **Estética de la interfaz de usuario.** Capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.
- **Accesibilidad.** Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

A los participantes de las pruebas se les explicó la finalidad de la aplicación además de un breve resumen de los conocimientos mínimos que deben de tener, como por ejemplo que los registros no se pueden editar ni borrar, o que dos registros no se pueden solapar.

Para documentar los resultados obtenidos se sigue el siguiente esquema. En cuanto a las métricas, se tomaron las siguientes:

Para cada acción se tomaron tres métricas con las que se pudo determinar si era necesario realizar cambios en la aplicación Android. Estas métricas son:

- Éxito de la acción.
- Tiempo, en segundos, que tarda el participante en realizar la acción.
- Número de fallos que comente el participante. Esto se determinará comparando el número de pulsaciones que realiza el participante para completar la acción frente al número mínimo de pulsaciones necesarias para realizarla.

Prueba de Usabilidad – Iniciar sesión y registrar entrada	
Descripción de la acción a realizar	El participante inicia sesión en la aplicación con las credenciales que se la ha facilitado. A continuación, procede a registrar la entrada para el momento actual.
Métricas	- Éxito: Sí - Tiempo: 4 segundos (Baja) - Pulsaciones: 1/1 (sin contar las necesarias para el login)
Observaciones	Ninguna.
Modificaciones realizadas	Ninguna.

Prueba de Usabilidad – Consultar sus registros	
Descripción de la acción a realizar	Teniendo la aplicación cerrada previamente, pero sin haber cerrado sesión, el participante procede a consultar sus registros del mes actual.
Métricas	- Éxito: Sí - Tiempo: 4 segundos (Baja) - Pulsaciones: 5/4
Observaciones	El participante pulsó sobre “Registro Manual” por error, pero de seguido pulsó correctamente en “Mis Registros”.
Modificaciones realizadas	Ninguna.

Prueba de Usabilidad – Registro diferido completo	
Descripción de la acción a realizar	Partiendo del escenario anterior, el participante procede a realizar un registro en diferido para el día anterior con entrada a las 9:00 y salida a las 15:30
Métricas	- Éxito: Sí - Tiempo: 17 segundos (Media) - Pulsaciones: 16/14
Observaciones	La diferencia de pulsaciones se debe a que el participante confundió el 3 con el 15.
Modificaciones realizadas	Ninguna.

Prueba de Usabilidad – Registro de salida rápido	
Descripción de la acción a realizar	Partiendo del escenario anterior, el participante procede a registrar su salida para el momento actual.
Métricas	- Éxito: Sí - Tiempo: 4 segundos (Baja) - Pulsaciones: 3/3
Observaciones	Ninguna.
Modificaciones realizadas	Ninguna.

Prueba de Usabilidad – Completar registro	
Contexto	El participante iniciará sesión con otras credenciales. El día anterior, en esta cuenta se ha realizado un registro de entrada para dejar preparado el escenario para la prueba.
Descripción de la acción a realizar	El participante inicia sesión con las nuevas credenciales. A continuación, procede a completar el registro incompleto del día anterior registrando la salida.
Métricas	- Éxito: Sí - Tiempo: 34 segundos (Alta) - Pulsaciones: 11/8 (sin contar las necesarias para el login)
Observaciones	El participante pensó que el botón del “Registrar salida” de la vista principal era para completar el registro, pulsando varias veces al no distinguir que el botón estaba desactivado.
Modificaciones realizadas	- Cambiar el texto de los botones de la vista principal por “Salida ahora” y “Entrada ahora”. - Cambiar el color al botón cuando está desactivado para que refleje dicho estado.

Prueba de Usabilidad – Informe de los registros de otro empleado	
Descripción de la acción a realizar	Partiendo del escenario anterior, el participante procede a consultar los registros de otro empleado realizados el mes pasado y los descargará formato PDF.
Métricas	- Éxito: Sí - Tiempo: 8 segundos (Baja) - Pulsaciones: 7/7
Observaciones	Ninguna.
Modificaciones realizadas	Ninguna.

Prueba de Usabilidad – Añadir nuevo administrador	
Descripción de la acción a realizar	Partiendo del escenario anterior, el participante procede a añadir un nuevo administrador al sistema.
Métricas	- Éxito: Sí - Tiempo: 19 segundos (Media) - Pulsaciones: 4/4 (sin contar las necesarias para introducir los datos)
Observaciones	El participante introdujo un login que ya existía.
Modificaciones realizadas	Ninguna.

Tras estas pruebas, las modificaciones que se realizaron fueron las siguientes:

- Cambiar el texto de los botones de la vista principal de “Registrar salida” a “Salida ahora”, y “Registrar entrada” por “Entrada ahora”.
- Cambiar el color de los botones de la vista principal cuando están desactivados para reflejar este estado. Se ha realizado aclarando un poco el color cuando están desactivados.
- Cambiar el texto de la opción del menú lateral “Mis registros” por “Consultar registros”.
- Se aumentó el tiempo que permanecían visibles los mensajes de información.

6 Conclusiones y trabajo futuro

Con este trabajo, se ha conseguido desarrollar un sistema completo con el que una empresa podría cumplir con el Real Decreto-ley 8/2019, permitiendo al empresario tener un seguimiento de las horas de entrada y salida de sus empleados, y a estos el poder registrar su hora de entrada y salida, junto con la posibilidad de generar informes de sus registros.

Con este sistema, las pequeñas empresas y comercios se pueden despreocupar de la conservación de los registros ya que quedarán almacenados en una base de datos en la nube.

Tras realizar las pruebas de usabilidad, se ha comprobado que la aplicación resulta bastante intuitiva para el usuario medio, por lo que el objetivo de desarrollar una aplicación de fácil uso se ha cumplido.

Por otra parte, este trabajo me ha servido para ser consciente de la dificultad que supone la planificación y ejecución de un proyecto completo. Además, he podido poner en práctica los conocimientos obtenidos durante la carrera, tanto directa como indirectamente, ya que, con la base adquirida gracias a todas las asignaturas, la parte del aprendizaje de nuevas tecnologías se hizo más sencillo.

6.1 Trabajo futuro

A medida que iba avanzando en el trabajo se me iban ocurriendo nuevas funcionalidades que se podrían añadir al sistema. A continuación, se listan aquellas funcionalidades que se podrían incorporar en siguientes versiones del sistema:

- **Firmar los registros.** Mensualmente, por ejemplo, se mandaría una notificación a todos los usuarios para que revisaran sus registros, y si están conformes los firmarían. Por otro lado, un administrador deberá dar el visto bueno a los registros de los empleados una vez que estos los hayan firmado. De esta forma, se podría asegurar la conformidad por ambas partes de las horas que han quedado registradas.
- **Calendario de vacaciones/festivos.** Con esto se podría centralizar en un mismo sistema toda la gestión de las vacaciones de los empleados, días festivos, etc. De esta forma, sería un administrador quien asignaría las vacaciones, mientras que los empleados solo podrían visualizar en un calendario sus festivos y días de vacaciones.
- **Desarrollar aplicación multiplataforma.** La idea sería utilizar algún framework, como Ionic, que permita el desarrollo de aplicaciones multiplataforma. El objetivo de esto es conseguir que el sistema se pueda usar en cualquier dispositivo.
- **Envío automático de informe.** Si se añadiera la dirección de correo electrónico a los datos de los empleados, se podría hacer que mensualmente, por ejemplo, se mandara a todos los empleados un informe con sus registros de entrada/salida realizados en este periodo.
- **Dar de baja empleados.** Aunque para este trabajo no se planteó esta posibilidad, esta funcionalidad debería de añadirse. Debido a que el Real Decreto-ley obliga a mantener los registros durante al menos 4 años, al dar de baja a un empleado no se eliminaría su información ni sus registros, sino que la cuenta del empleado se desactivaría.
- **Programar copia de seguridad de la base de datos.** Habría que configurar MongoDB Atlas para que realice una copia de seguridad cada X días para tener total seguridad de que no se pierden los registros.

7 Bibliografía

- [1] Real Decreto-ley 8/2019. Real Decreto-ley 8/2019, de 8 de marzo, de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo. BOE núm. 61, Jefatura del Estado, Madrid, España, 12 de marzo de 2019. Disponible en: www.boe.es/eli/es/rdl/2019/03/08/8
- [2] Control Laboral. Último acceso: 14/06/2020. [Online]. Disponible en: controllaboral.es
- [3] Hora Laboral. Último acceso: 14/06/2020. [Online]. Disponible en: horalaboral.com
- [4] Git. Último acceso: 14/06/2020. [Online]. Disponible en: git-scm.com
- [5] Github. Último acceso: 14/06/2020. [Online] Disponible en: github.com
- [6] Android, Wikipedia. Último acceso: 14/06/2020. [Online] Disponible en: es.wikipedia.org/wiki/Android
- [7] Alfonso Casas, (25/02/2019). *iPhone vs Android: cuota de mercado*. Último acceso: 14/06/2020. [Online] Disponible en: pcworld.es/articulos/smartphones/iphone-vs-android-cuota-de-mercado-3692825/
- [8] José García Nieto, (23/10/2018). *Así es como Android se ha comido el mercado en diez años*. Último acceso: 14/06/2020. [Online] Disponible en: xatakamovil.com/sistemas-operativos/asi-como-android-se-ha-comido-mercado-diez-anos
- [9] NPM. Último acceso: 14/06/2020. [Online] Disponible en: npmjs.com
- [10] W3Schools, (n.d.). *What is npm?* Último acceso: 14/06/2020. [Online] Disponible en: w3schools.com/whatis/whatis_npm.asp
- [11] NodeJs. Último acceso: 14/06/2020. [Online] Disponible en: nodejs.org/es
- [12] Jesús Lucas, (04/09/2019). *Qué es NodeJS y para qué sirve*. Último acceso: 14/06/2020. [Online] Disponible en: openwebinars.net/blog/que-es-nodejs
- [13] Rownok Ara, (n.d.). *Top 20 Best NodeJS frameworks for developers in 2020*. Último acceso: 14/06/2020. [Online] Disponible en: ubuntupit.com/best-nodejs-frameworks-for-developers
- [14] Mozilla, (10/06/2020). *Introducción a Express/Node*. Último acceso: 14/06/2020. [Online] Disponible en: developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction
- [15] F.J. Moreno Arboleda, J.E. Quintero Rendón, R. Rueda Vásquez, (2016). *Una comparación de rendimiento entre Oracle y MongoDB*. Ciencia e Ingeniería Neogranadina. Último acceso: 14/06/2020. [Online] Disponible en: redalyc.org/jatsRepo/911/91145342002/html/index.html
- [16] MongoDB Atlas. Último acceso: 14/06/2020. [Online] Disponible en: mongodb.com/cloud/atlas
- [17] Scrum Manager, (23/03/2014). *Historia de usuario*. Último acceso: 14/06/2020. [Online] Disponible en: scrummanager.net/bok/index.php/Historia_de_usuario
- [18] Adriana Peralta, (2003). *Metodología SCRUM*. Último acceso: 14/06/2020. [Online] Disponible en: fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf
- [19] Óscar Blancarte, (10/12/2018). *Data Access Object (DAO) Pattern*. Último acceso: 14/06/2020. [Online] Disponible en: oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern
- [20] Óscar Blancarte, (30/11/2018). *Data Transfer Object (DTO) – Patrón de diseño*. Último acceso: 14/06/2020. [Online] Disponible en: oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/
- [21] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.

[22] MongoDB, (n.d.). *Data Model Design*. Último acceso: 14/06/2020. [Online] Disponible en: docs.mongodb.com/manual/core/data-model-design

[23] ISO 25000, (n.d.). *Usability*. Último acceso: 14/06/2020. [Online] Disponible en: iso25000.com/index.php/en/iso-25000-standards/iso-25010/61-usability