

# App para fichaje remoto integrado con el TimeTrack de IR Soluciones

**Aitor Montalvo Cano**

Grado de Ingeniería Informática

Desarrollo multiplataforma de aplicaciones móviles

**Carlos Sánchez Rosa**

**Jordi Almirall López**

**Carles Garruiges Olivella**

30/05/2022



Esta obra está sujeta a una licencia de  
Reconocimiento-NoComercial-CompartirIgual  
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>App para fichaje remoto integrado con el TimeTrack de IR Soluciones</i>
<b>Nombre del autor:</b>	<i>Aitor Montalvo Cano</i>
<b>Nombre del consultor/a:</b>	<i>Carlos Sánchez Rosa Jordi Almirall López</i>
<b>Nombre del PRA:</b>	<i>Carles Garrigues Olivella</i>
<b>Fecha de entrega (mm/aaaa):</b>	05/2022
<b>Titulación:</b>	<i>Grado de Ingeniería Informática</i>
<b>Área del Trabajo Final:</b>	<i>Desarrollo multiplataforma de aplicaciones móviles</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave</b>	<i>Aplicación, fichaje, integrado</i>
<p><b>Resumen del Trabajo (máximo 250 palabras):</b> <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Este trabajo presenta la necesidad de desarrollar una aplicación para dispositivos móviles que integre las diferentes funcionalidades de control de presencia física disponibles en el web service "TimeTrack", perteneciente a la empresa IR Soluciones.</p> <p>Esta aplicación pretende principalmente agilizar y facilitar la autenticación, login y logout de los trabajadores de la empresa desde cualquier ubicación, sin necesidad de utilizar un dispositivo físico.</p> <p>Se implementarán diferentes tipos de fichaje, dependiendo del tipo de actuación en cada caso, que guardarán una localización que podrá ser verificada, así como la posibilidad de consultar el historial de los mismos y otras funcionalidades como poder consultar el horario o la disponibilidad de vacaciones de los trabajadores de la empresa.</p> <p>El objetivo es mostrar cómo se debe crear una aplicación de este tipo, desde su planteamiento hasta la finalización de su desarrollo.</p> <p>La metodología escogida para obtener un producto de calidad y cumplir los plazos de entrega ha sido Scrum y para su desarrollo se ha elegido el entorno IDE JetBrains WebStorm con Ionic y Vue Javascript.</p> <p>El resultado ha sido un producto de calidad, totalmente operativo que cumple las funcionalidades especificadas y que termina siendo una herramienta de trabajo imprescindible para la empresa y sus trabajadores.</p>	

**Abstract (in English, 250 words or less):**

This work presents the need to develop an application for mobile devices that integrates the different functionalities of physical presence control available in the "TimeTrack" web service, belonging to the company IR Soluciones.

This application is mainly intended to streamline and facilitate the authentication, login and logout of the company's workers from any location, without the need to use a physical device.

Different types of signing will be implemented, depending on the type of action in each case, which will save a location that can be verified, as well as the possibility of consulting their history and other functionalities such as being able to check the schedule or the availability of vacations company workers.

The objective is to show how an application of this type should be created, from its planning to the end of its development.

The methodology chosen to obtain a quality product and meet delivery deadlines has been Scrum and the JetBrains WebStorm IDE environment with Ionic and Vue Javascript has been chosen for its development.

The result has been a quality product, fully operational that meets the specified functionalities and ends up being an essential work tool for the company and its workers.

# Índice

1. Plan de trabajo .....	1
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo .....	2
1.5 Breve resumen de productos obtenidos .....	3
1.6 Breve descripción de los otros capítulos de la memoria.....	3
2. Análisis y diseño centrado en el usuario .....	5
2.1 Usuarios y contexto de uso .....	5
2.1.1 Métodos de indagación .....	5
2.1.2 Contexto de uso.....	7
2.1.3 Perfiles de usuario .....	7
2.1.4 Análisis de tareas.....	8
2.2 Diseño Conceptual .....	8
2.2.1 Escenarios de uso .....	8
2.3 Prototipado .....	10
2.3.1 Sketches .....	10
2.3.2 Prototipos de alta fidelidad.....	11
2.4 Evaluación de diseño .....	14
2.4.1 Planificación previa .....	14
2.4.2 Tareas a realizar .....	14
2.4.3 Preguntas para obtener información.....	14
2.4.4 Encuestas realizadas.....	15
3. Casos de uso .....	16
3.1 Caso de uso “Login” .....	16
3.2 Caso de uso “Imputar fichaje” .....	17
3.3 Caso de uso “Imputar descanso” .....	18
3.4 Caso de uso “Ver vacaciones” .....	19
3.5 Caso de uso “Solicitar vacaciones” .....	20
3.6 Caso de uso “Ver horarios” .....	21
3.7 Caso de uso “Ver perfil”.....	22
3.8 Diagrama de flujo general .....	23
4. Diseño de la arquitectura.....	24
4.1 Base de datos .....	24
4.2 Arquitectura del sistema .....	26
5. Desarrollo de la aplicación .....	27
5.1 Herramientas y configuración de entorno.....	27
5.1.1 JDK.....	27
5.1.2 Android SDK .....	27
5.1.4 Gradle .....	28
5.1.4 Variables de entorno .....	28
5.1.5 Node.js y NPM .....	28
5.1.6 Vue.js.....	29
5.1.7 Ionic 4 .....	29
5.1.8 Creación del proyecto .....	30
5.1.9 Cordova .....	32
5.1.10 JetBrains Webstorm .....	33

5.1.11 Android Studio .....	34
5.1.12 Android Virtual Device .....	35
5.2 Código y programación .....	35
5.2.1 Enrutamiento.....	35
5.2.2 Sesión de usuario .....	36
5.2.2 Otras dependencias utilizadas .....	37
5.2.2.1 Integración de Google Maps.....	37
5.2.2.2 Selectores de fecha y hora .....	37
5.2.2.3 Calendario .....	39
5.2.3 API PHP .....	39
5.2.4 Componentes Vue .....	41
5.2.1.1 Login .....	41
5.2.1.2 Historial.....	41
5.2.1.3 Fichaje .....	43
5.2.1.4 Perfil.....	44
5.2.1.5 CambiarPwd .....	44
5.2.1.6 Horarios .....	44
5.2.1.7 Calendario .....	45
5.2.1.8 PeticiónVacaciones .....	46
5.3 Análisis del estado del proyecto y correcciones .....	47
5.3.1 Estado del proyecto .....	47
5.3.2 Correcciones.....	47
5.3.3 Funcionalidades extra .....	48
5.3.3 Vista al futuro .....	49
6. Pruebas .....	51
6.1 Descripción de las pruebas .....	51
6.1.1 Localhost.....	51
6.1.2 Dispositivos móviles Android .....	51
6.1.3 Emulador Android .....	51
6.2 Pruebas unitarias .....	52
7. Aspecto final de la aplicación .....	53
8. Conclusiones.....	59
9. Glosario .....	60
10. Bibliografía .....	63
11. Anexos .....	64

## Lista de figuras

Ilustración 1. Diagrama de Gantt	3
Ilustración 2. Sketches a mano alzada	10
Ilustración 3. Prototipos de alta fidelidad	11
Ilustración 4. Diagrama de flujo "Login"	16
Ilustración 5. Diagrama de flujo "Imputar fichaje"	17
Ilustración 6. Diagrama de flujo "Imputar descanso"	18
Ilustración 7. Diagrama de flujo "Ver vacaciones"	19
Ilustración 8. Diagrama de flujo "Solicitar vacaciones"	20
Ilustración 9. Diagrama de flujo "Ver horarios"	21
Ilustración 10. Diagrama de flujo "Ver perfil"	22
Ilustración 11. Diagrama de flujo general	23
Ilustración 12. Diagrama UML base de datos reducida	24
Ilustración 13. Diagrama UML base de datos completa	25
Ilustración 14. Diagrama con la arquitectura del sistema	26
Ilustración 15. Versión JDK	27
Ilustración 16. Logotipo Android SDK	27
Ilustración 17. Variables de entorno	28
Ilustración 18. Logotipo Node.js	28
Ilustración 19. Versiones de Node.js y NPM	29
Ilustración 20. Logotipo Vue	29
Ilustración 21. Logotipo Ionic	29
Ilustración 22. Creación proyecto Vue.js	30
Ilustración 23. Proyecto Vue.js compilado	31
Ilustración 24. Imports Ionic dentro del "main.js" de Vue	31
Ilustración 25. Logotipo Cordova	32
Ilustración 26. Parámetros del proyecto en "config/index.js"	32
Ilustración 27. Logotipo Webstorm	33
Ilustración 28. Proyecto en IDE Webstorm	34
Ilustración 29. Logotipo Android Studio	34
Ilustración 30. Proyecto Vue en Android Virtual Device	35
Ilustración 31. Ejemplo de enrutamiento en Vue.js	36
Ilustración 32. Ejemplo del router de Vue.js	36
Ilustración 33. Ejemplo de inicio de sesión en Vue.js	36
Ilustración 34. Ejemplo de vue2-google-maps	37
Ilustración 35. Método de geolocalización	37
Ilustración 36. Ejemplo de vuejs-datepicker	38
Ilustración 37. Ejemplo de vue-datetime	38
Ilustración 38. Ejemplo de vue-fullcalendar	39
Ilustración 39. Ejemplo de creación de formulario en Vue.js	39
Ilustración 40. Ejemplo de envío por POST en Vue.js	40
Ilustración 41. Ejemplo del switch(\$acc) de la API	40
Ilustración 42. Ejemplos de la respuesta de la API	40
Ilustración 43. Ejemplo de Vue.js recibiendo respuesta de la API	40
Ilustración 44. Fichajes en el componente 'Historial'	41
Ilustración 45. Menú inferior del componente 'Historial'	42
Ilustración 46. Ejemplo del GIF del reloj en fichaje abierto	42
Ilustración 47. Ejemplo de opciones de cada fichaje al deslizar	43

Ilustración 48. Modal del componente 'Fichaje'	43
Ilustración 49. Ejemplo de la ficha con los horarios	45
Ilustración 50. Formulario de petición de vacaciones	46

# 1. Plan de trabajo

## 1.1 Contexto y justificación del Trabajo

Actualmente la empresa riojana IR Soluciones dispone de un servicio web llamado TimeTrack en el cual los trabajadores pueden gestionar diferentes tipos de fichaje para control de presencia física, peticiones de vacaciones, consulta de horarios, avisos de avería, actuaciones de soporte a clientes, solución de incidencias, etc.

En las instalaciones de la empresa, además, existen dispositivos físicos de fichaje que permiten la identificación mediante huella digital o tarjeta "lágrima" personal. En estos dispositivos físicos únicamente se permite fichar para registrar la entrada y la salida de las instalaciones.

Debido a esto, surge la necesidad de desarrollar una aplicación que permita integrar el resto de las funcionalidades disponibles en el TimeTrack desde cualquier ubicación y dispositivo móvil de una forma mucho más ágil y sencilla.

Hasta el momento del desarrollo de este proyecto, la empresa no dispone de ninguna aplicación en este sentido, así que se parte desde cero, integrando funcionalidades del servicio web y base de datos existentes.

## 1.2 Objetivos del Trabajo

El objetivo general es desarrollar una aplicación para la empresa IR Soluciones que integre todas las funcionalidades del servicio web TimeTrack, en la que los trabajadores de la empresa puedan fichar desde cualquier ubicación y dispositivo.

Objetivos críticos para el éxito del proyecto:

- Crear una interfaz de usuario que permita la autenticación de los trabajadores en la aplicación.
- Integrar un sistema de fichaje de entrada y salida de la jornada laboral, en general.
- Integrar un sistema de fichaje de entrada y salida para cada tarea o descanso que realice el trabajador.

- Identificar y registrar el origen de cada fichaje, ya sea el propio TimeTrack, la aplicación móvil o los dispositivos físicos mediante lágrima.

Otros objetivos no prioritarios:

- Crear un apartado que permita consultar los horarios de los trabajadores.
- Crear un apartado que permita consultar las vacaciones de los trabajadores y realizar una petición de vacaciones.
- Registrar la ubicación de cada fichaje mediante la integración de Google Maps en la aplicación.

### 1.3 Enfoque y método seguido

La aplicación es un producto totalmente nuevo en la empresa, ya que hasta el momento de su desarrollo no existía nada parecido como aplicación móvil, aunque sí que existe un servicio web contra el que se integrarán las funcionalidades.

La metodología escogida para el desarrollo del proyecto es Scrum, ya que es un marco de trabajo con buenas prácticas para desarrollo ágil de software que permite trabajar colaborativamente en caso necesario, adoptar una estrategia de desarrollo incremental, solapar las diferentes fases del desarrollo y obtener los mejores resultados.

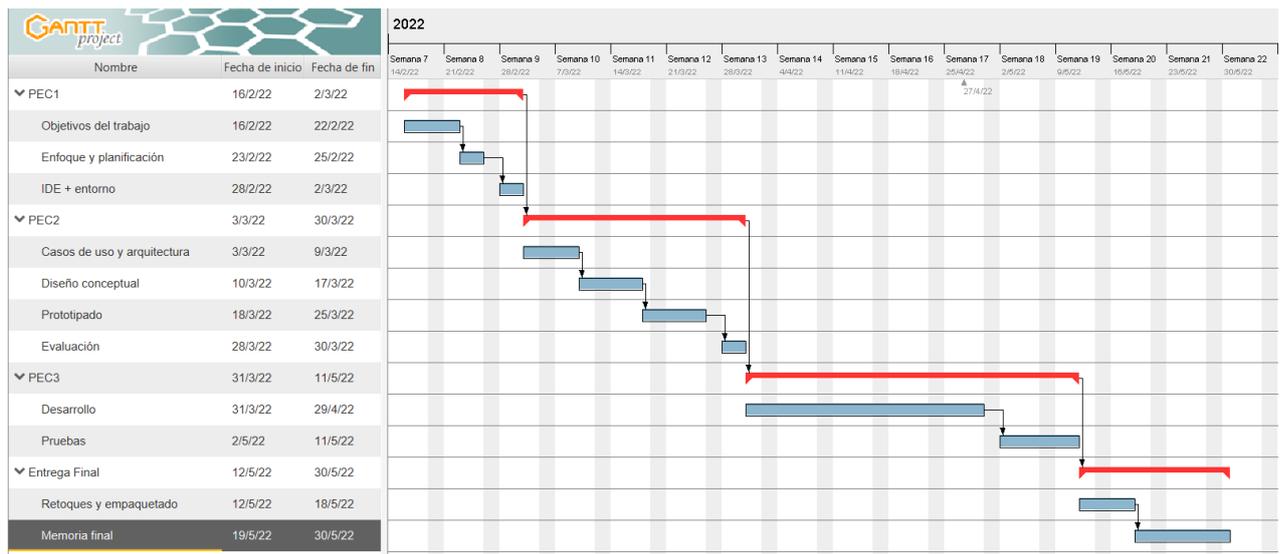
Una forma de poder visualizar los objetivos y el punto de desarrollo en el que se encuentra el proyecto es mediante la herramienta Trello, que permite hacer un listado de las tareas y pasos por los que debe pasar el desarrollo.

### 1.4 Planificación del Trabajo

El proyecto estará guiado por la metodología de trabajo Scrum, ya que, en este caso, al tener que permanecer en contacto continuo con el proveedor del servicio web TimeTrack, ya existente en la empresa, es la más idónea para el correcto desarrollo de la aplicación.

En cuanto a tecnología, la aplicación se desarrolla en un entorno de trabajo basado en un proyecto Vue.js con integración de Ionic, mediante el IDE de JetBrains WebStorm.

A continuación, se muestra un diagrama de Gantt donde se pueden visualizar las diferentes partes del desarrollo del proyecto:



**Ilustración 1. Diagrama de Gantt**

## 1.5 Breve resumen de productos obtenidos

Los productos obtenidos son:

- Una aplicación para la empresa IR Soluciones que permite registrar los fichajes y las diferentes funcionalidades del servicio web TimeTrack, que los trabajadores pueden utilizar desde cualquier ubicación y dispositivo.
- Un documento con la memoria finalizada de este TFG donde se detallará todo el proceso de elaboración de la aplicación.
- Un video de presentación de la aplicación, donde se explicarán sus funcionalidades y una idea acerca de los pasos que han sido necesarios para realizarla.
- Un documento con el manual de la aplicación para los usuarios.

## 1.6 Breve descripción de los otros capítulos de la memoria

### 2. Análisis y diseño centrado en el usuario (DCU)

#### 2.1 Usuarios y contexto de uso

Mediante los métodos de indagación se involucra a los usuarios para conocer sus características, necesidades y objetivos, así como el contexto de uso de la aplicación y los diferentes perfiles de usuario identificados.

## 2.2. Diseño Conceptual

Se elaboran los escenarios de uso y los flujos de interacción, que describen desde el punto de vista de cada usuario cómo se utilizará la aplicación en cada contexto concreto para determinar las funcionalidades necesarias y su diseño.

## 2.3. Prototipado

Se presentan los primeros bocetos elaborados a mano alzada y su posterior diseño como prototipos horizontales de alta fidelidad de las diferentes pantallas que tendrá la aplicación.

## 2.4. Evaluación de diseño

Se planifica y se define la evaluación del diseño prototipado y para corregir los errores encontrados.

## 3. Casos de uso

Son las definiciones de las funcionalidades que se necesitan implementar en la aplicación. Qué tiene que hacer cada una y cómo debe hacerlo.

## 4. Diseño de la arquitectura

Se define la arquitectura del sistema en el que se integra la aplicación, así como la estructura y diseño de la base de datos sobre la que se va a implementar.

## 5. Desarrollo de la aplicación

Se definen las herramientas y la configuración del entorno utilizados durante el desarrollo de la aplicación. Se entrega el código fuente implementado y la aplicación empaquetada. Se realiza un análisis del estado del proyecto y se establecen las posibles correcciones.

## 6. Pruebas

Se planifican y definen las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación.

## 7. Aspecto final de la aplicación

Se presentan capturas reales de la aplicación ya acabada al finalizar el proyecto

## 2. Análisis y diseño centrado en el usuario

### 2.1 Usuarios y contexto de uso

#### 2.1.1 Métodos de indagación

Para este proyecto he utilizado más de un método de indagación al mismo tiempo, debido a que me encuentro realizando una formación dentro de la empresa que requiere la aplicación.

- Observación e investigación contextual:
  - Planteamiento: observar directamente a los usuarios e investigar el contexto laboral en el que van a usar la aplicación y familiarizarse con él.
  - Desarrollo: se realiza sobre todo los primeros días de adaptación en la empresa. Se solicita información acerca de los métodos de fichaje (lágrima, TimeTrack, etc...), se observa la metodología de los usuarios a la hora de fichar y se les pregunta las dudas en caso necesario.
  - Resultados: se aclaran varios conceptos como el de “fichaje mediante lágrima” o qué funcionalidades están desarrolladas en el servicio web TimeTrack
  - Conclusiones: la observación es una fuente fiable y directa de información para los primeros pasos de un proyecto, aunque puede conllevar demasiado tiempo o incluso que el usuario, inconscientemente, altere las condiciones habituales del contexto.
- Shadowing:
  - Planteamiento: entender como los usuarios van a interactuar con la aplicación, comprender los procesos existentes y adquirir una visión en detalle del contexto de uso, acompañando al usuario en todo momento.
  - Desarrollo: durante el primer día de estancia en la empresa, realizo el seguimiento de mi “facilitador” asignado (trabajador y futuro usuario de la aplicación) en todos los pasos que realiza en su día a día, incluyendo entradas, salidas y fichajes en TimeTrack.
  - Resultados: el seguimiento se realiza de forma general para integración en la empresa, y en este caso, para el desarrollo del proyecto, sólo se obtiene algo de información acerca del uso de TimeTrack en los fichajes y al realizar petición de vacaciones.

- Conclusiones: En mi caso no ha sido de gran utilidad, pero si se realiza sin ser intrusivo con el usuario observado, puede ser una buena fuente de información inicial.
- Benchmarking:
  - Planteamiento: Analizar productos similares que ya existan en el mercado para encontrar puntos de partida, inspiración o soluciones a posibles problemas.
  - Desarrollo: se buscan en PlayStore aplicaciones similares. Se prueban varias de ellas, por ejemplo “Visual Presencia”, de la cual se extraen varias ideas para el diseño y planteamiento del proyecto.
  - Resultados: La idea de añadir un mapa de Google con la ubicación del usuario a la hora de realizar el fichaje surge del análisis de “Visual Presencia”.
  - Conclusiones: una buena forma de analizar y evaluar los productos que son competidores o que tienen funcionalidades parecidas, de la cual se obtienen buenas ideas para obtener una experiencia de usuario óptima.
- Encuestas en profundidad: se han realizado reuniones con el responsable del proyecto y otros trabajadores de la empresa para obtener información necesaria en cuanto a sus necesidades para el desarrollo de la aplicación.
  - Planteamiento: obtener mediante preguntas directas a los usuarios para conocer el alcance del proyecto, funcionalidades básicas necesarias de la aplicación y cualquier tipo de duda con respecto a su diseño o su desarrollo.
  - Desarrollo: se preparan cuestionarios y se realizan reuniones puntuales al inicio de cada fase del desarrollo de la aplicación con el jefe de proyecto de la empresa donde se aclaran las dudas y conceptos. Además, la empresa asigna a un programador Senior para supervisar mi proyecto con el cual se realizan reuniones semanales.
  - Resultados: se aclaran periódicamente conceptos implicados en cada fase del desarrollo como la metodología o plan de trabajo, el entorno de trabajo, el diseño de la aplicación y las funcionalidades de la aplicación.
  - Conclusiones: probablemente, el método que más ayuda e información me ha proporcionado. Las encuestas realizadas al inicio de cada fase y el seguimiento semanal han permitido una correcta organización, planteamiento y ejecución del proyecto.

### 2.1.2 Contexto de uso

En la empresa IR Soluciones están muy familiarizados con el entorno del sistema TimeTrack y con los fichajes a lo largo de la jornada laboral para tener un seguimiento de las horas de trabajo. No sólo registran las entradas y salidas, sino que también utilizan TimeTrack para contabilizar las horas que dedican a cada cliente, proyecto o labor concreta.

Hasta la creación de la aplicación, toda esta información se registraba exclusivamente a través del servicio web de TimeTrack, haciendo imprescindible la conexión al sistema a través de un navegador web.

Con la llegada de la aplicación móvil, se facilita al usuario la posibilidad de realizar estos fichajes desde cualquier ubicación, no sólo desde dentro de cualquiera de las oficinas de la empresa, sino que además los trabajadores que realizan salidas para ver a clientes o realizar intervenciones técnicas in situ, llevan las herramientas, funcionalidades e información necesarias en su dispositivo móvil.

### 2.1.3 Perfiles de usuario

Existen diferentes perfiles de usuario dentro de la base de datos de TimeTrack, como por ejemplo (admin, jefe de equipo, senior, junior, practicante, etc...), pero de momento, ya que es muy probable que más adelante la aplicación entre en producción, y para el desarrollo de este TFG, todos los usuarios tendrán el mismo perfil de uso y realizarán las mismas funcionalidades que los demás dentro de la aplicación.

#### Usuario básico

- Los usuarios se localizan en las oficinas de Logroño y Sevilla, pero ambos harán el mismo uso de la app sin que afecte al contexto.
- A nivel general, los usuarios de la empresa son personas jóvenes con alguna excepción puntual. Normalmente entre los 20 y 40 años.
- Los usuarios de la aplicación podrán ser masculinos o femeninos sin que esto sea relevante para su desarrollo.
- Los usuarios acreditan una gran experiencia con el uso de tecnologías y dispositivos móviles. Al tratarse de una empresa del sector TIC, los empleados son técnicos o ingenieros informáticos y en algún caso poseen titulaciones en administración.
- Los usuarios utilizarán la aplicación indistintamente, tanto dentro de las instalaciones de la empresa como fuera de ellas cuando sea necesario.

#### 2.1.4 Análisis de tareas

Las funcionalidades que la aplicación debe cubrir son las siguientes:

- Login: pantalla que permita iniciar sesión al usuario con su contraseña de TimeTrack. Tras iniciar sesión se cargará la pantalla principal.
- Historial (principal): pantalla donde aparecerán los accesos a las diferentes funcionalidades y donde se mostrará de la forma más intuitiva posible el historial de los fichajes realizados por el usuario.
- Vacaciones: pantalla que mostrará un calendario donde se podrá ver qué usuarios están de vacaciones o de permiso cada día.
  - Se podrá realizar una petición de vacaciones para que sea revisada por el supervisor.
- Horarios: pantalla que mostrará los horarios de trabajo de cada usuario.
- Perfil: pantalla que mostrará la información de perfil del usuario y las opciones para cambiar su contraseña y cerrar sesión.
- Fichajes: se podrán imputar fichajes de entrada y salida con la posibilidad de elegir día, horario y una breve descripción.
  - Entrada y salida de la jornada laboral.
  - Actuaciones, tareas, proyectos, etc.
  - Descansos o pausas.

## 2.2 Diseño Conceptual

### 2.2.1 Escenarios de uso

#### **Escenario 1 - Login**

Planteamiento: El usuario llega a la oficina y tras unos minutos trabajando se da cuenta de que al entrar no ha fichado en el dispositivo de control de presencia física como es habitual. Gracias a la nueva aplicación, puede realizar esta tarea de forma rápida y sin necesidad de moverse.

Acción: Entra en la aplicación, se autentica e introduce un fichaje nuevo con la hora a la que ha llegado a la oficina.

## **Escenario 2 - Fichaje**

Planteamiento: El usuario acaba de tener una reunión con el jefe de proyecto y otro compañero de trabajo y necesita imputar el fichaje para contabilizar el tiempo.

Acción: Entra en la aplicación, se autentica e introduce un fichaje nuevo especificando la hora de entrada y salida de la reunión.

## **Escenario 3 – Fichaje remoto**

Planteamiento: El usuario del departamento de sistemas acaba de realizar una intervención in situ en las oficinas de un cliente. Sin regresar a la oficina, debe dirigirse a una nueva intervención con otro cliente. Gracias a la nueva aplicación, puede imputar en tiempo real el fichaje de la intervención que acaba de realizar.

Acción: Entra en la aplicación, se autentica e introduce un fichaje nuevo especificando el cliente y la hora de entrada y salida.

## **Escenario 4 - Descanso**

Planteamiento: El usuario está en el bar más cercano a la oficina de trabajo realizando un descanso de 20 minutos. En mitad de su café diario recuerda que no ha registrado la salida. Gracias a la nueva aplicación puede realizar esta tarea desde el bar sin necesidad de volver a la oficina.

Acción: Entra en la aplicación, se autentica, cierra el fichaje anterior e imputa un nuevo descanso especificando la hora.

## **Escenario 5 - Vacaciones**

Planteamiento: El usuario necesita solicitar unos días de permiso para aprovechar las vacaciones de Semana Santa.

Acción: Entra en la aplicación, se autentica y accede al apartado de las vacaciones. Tras revisar en el calendario si los días que necesita están disponibles, procede a realizar una nueva petición y la registra para que sea supervisada.

## **Escenario 6 - Horarios**

Planteamiento: El usuario necesita programar una reunión con otros dos compañeros de la oficina para hablar sobre un proyecto y necesita saber en qué horarios están disponibles mañana.

Acción: Entra en la aplicación, se autentica y accede a los horarios. Verifica el horario del compañero y prepara la reunión.

## 2.3 Prototipado

Para el diseño de la aplicación a nivel gráfico, se ha buscado información de forma general sobre otras aplicaciones similares para obtener una referencia (benchmarking), se han mantenido conversaciones con los usuarios de TimeTrack (futuros usuarios de la aplicación) y con el jefe de proyectos de IR Soluciones.

### 2.3.1 Sketches

Estos son los primeros bocetos de las diferentes pantallas de la aplicación realizados a mano alzada sobre un folio en blanco.

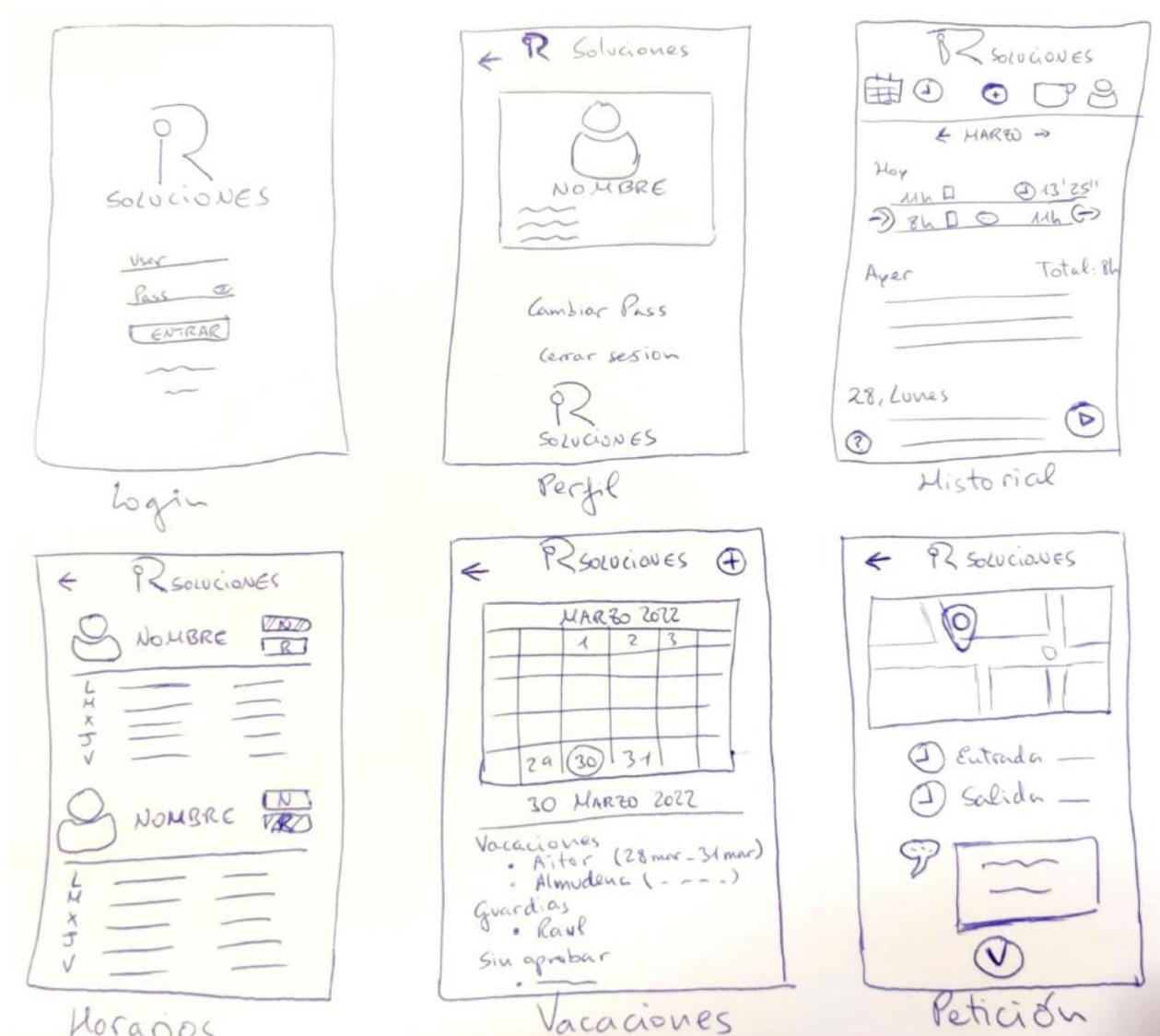


Ilustración 2. Sketches a mano alzada

### 2.3.2 Prototipos de alta fidelidad

Se presenta un diseño inicial realizado directamente en Balsamiq. Es el punto de partida puesto que por el momento los objetivos para el desarrollo de la aplicación son dinámicos y seguramente se modifiquen, descarten o añadan cosas en la entrega final de la aplicación.

Los prototipos presentados son los siguientes (por este orden): login, historial de fichajes (pantalla principal), horarios, vacaciones, perfil de usuario, nuevo fichaje y petición de vacaciones.

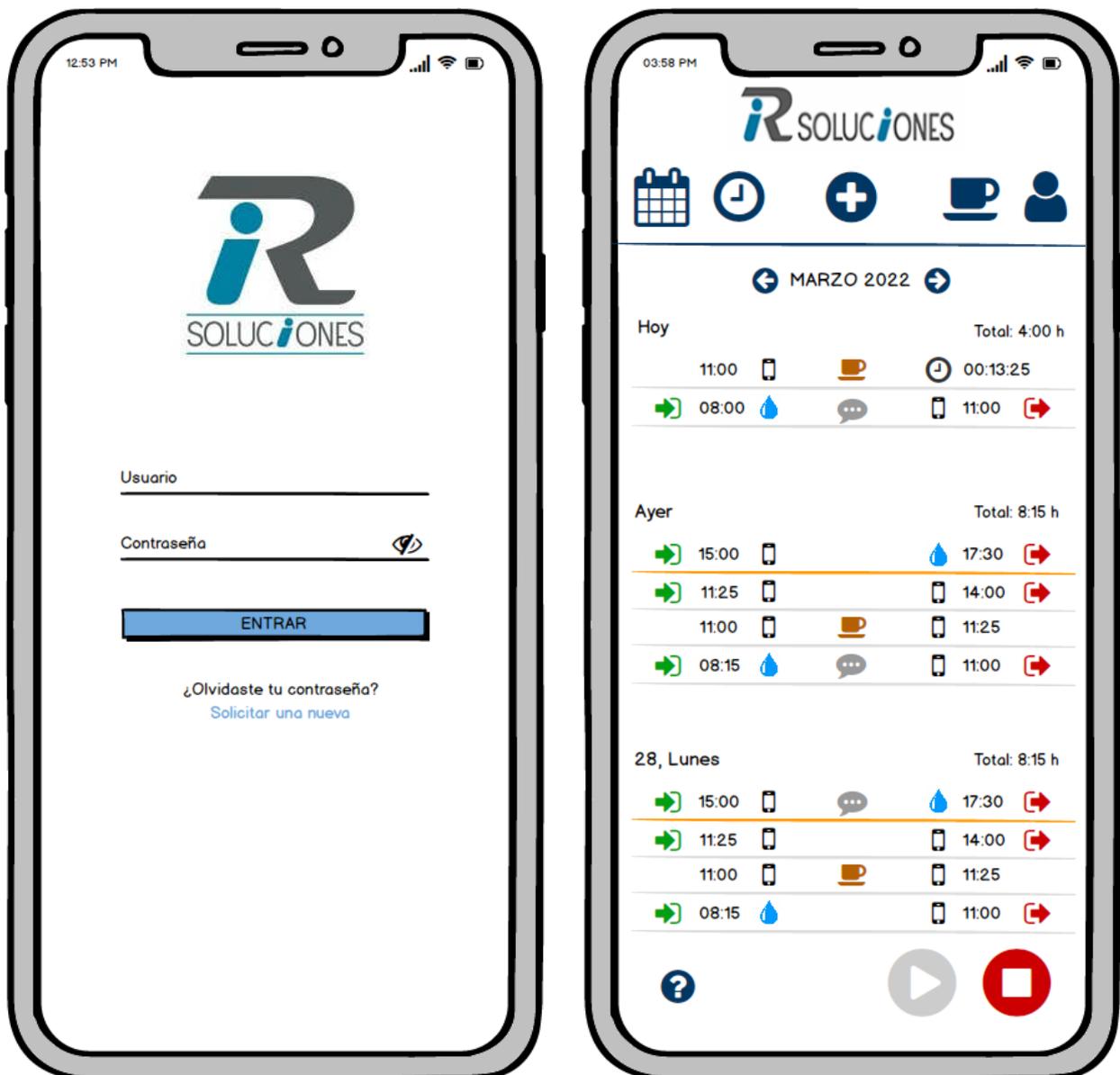


Ilustración 3. Prototipos de alta fidelidad

01:50 PM

**iR SOLUCIONES**

**Aitor Montalvo** Normal  
Retén

Lunes 08:30 - 14:00 15:00 - 17:30

Martes 08:30 - 14:00 15:00 - 17:30

Miércoles 08:30 - 14:00 15:00 - 17:30

Jueves 08:30 - 14:00 15:00 - 17:30

Viernes 08:30 - 14:00 15:00 - 17:30

**Almudena Gómez** Normal  
Retén

Lunes 08:30 - 14:00 15:00 - 17:30

Martes 08:30 - 14:00 15:00 - 17:30

Miércoles 08:30 - 14:00 15:00 - 17:30

Jueves 08:30 - 14:00 15:00 - 17:30

Viernes 08:30 - 14:00 15:00 - 17:30

01:52 PM

**iR SOLUCIONES**

MARCH 2022

S	M	T	W	T	F	S
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

30 de Marzo de 2022

**Vacaciones**

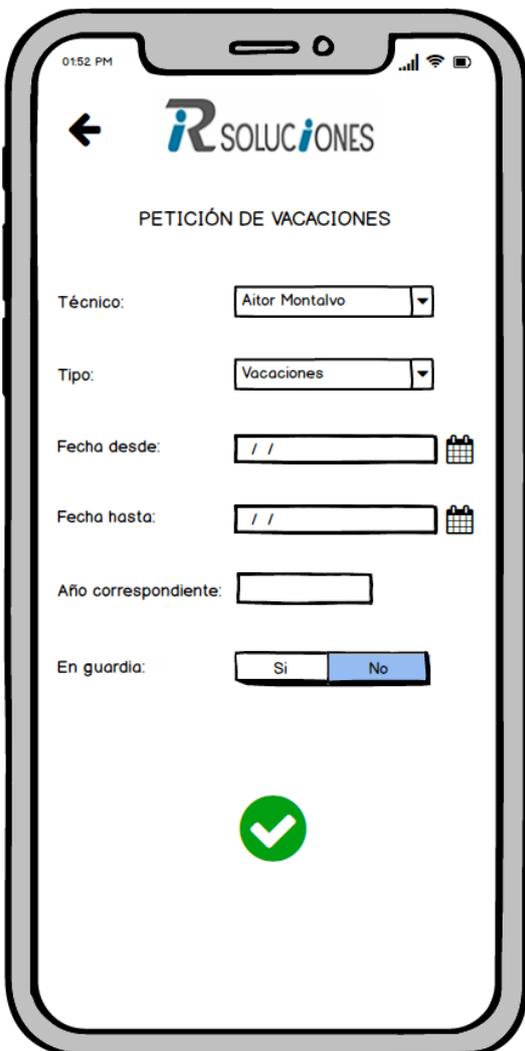
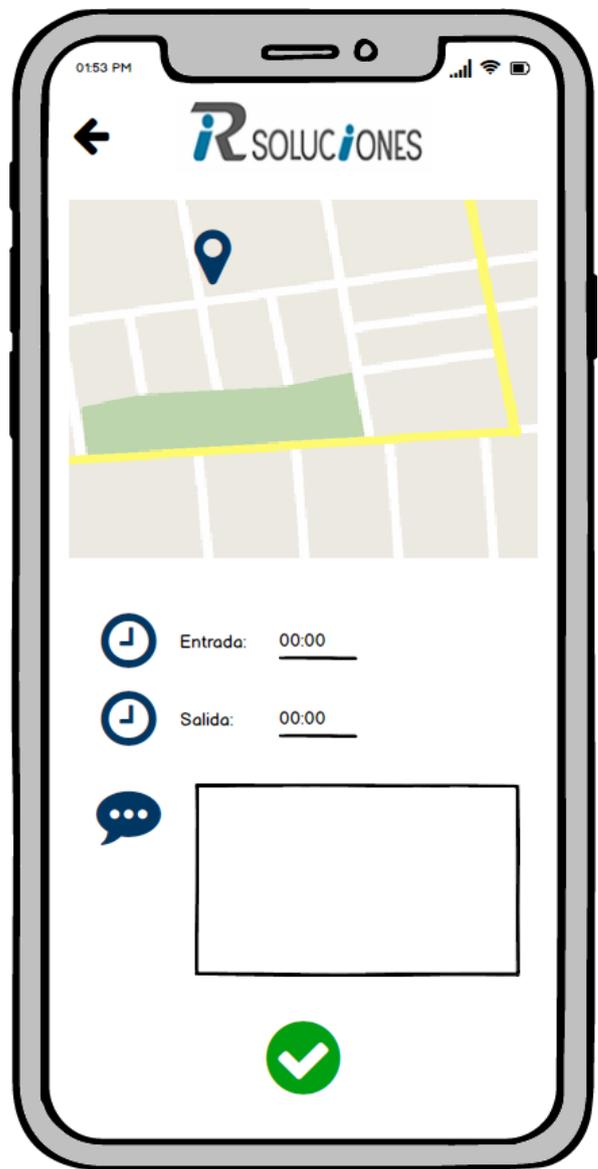
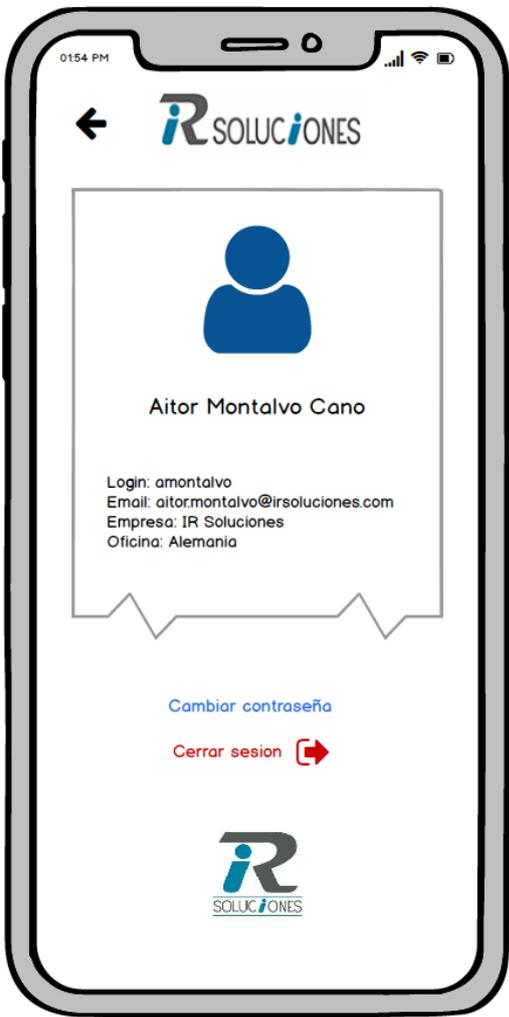
- Almudena Gomez (28 marzo - 31 marzo)
- Carlota Altí (30 marzo)

**Vacaciones en guardia**

- Raul Pérez (30 marzo)

**Pendiente de aceptación**

- Aitor Montalvo (29 marzo - 4 abril)



## 2.4 Evaluación de diseño

### 2.4.1 Planificación previa

Con estos prototipos de diseño como punto de partida se comenzará la fase de desarrollo, pasando a configurar el entorno y a programar el código fuente de la aplicación.

Cuando el desarrollo esté en una fase mínima viable como para poder interactuar con las funcionalidades básicas, se realizarán unos tests de prueba para que el usuario pueda experimentar una primera toma de contacto con la aplicación.

Los tests se realizarán con cada potencial usuario por separado y se realizará una observación minuciosa mientras que interactúa con la aplicación, realiza pruebas, detecta posibles errores y da su opinión acerca de cualquier aspecto técnico o de diseño.

### 2.4.2 Tareas a realizar

- Login
- Consulta de horarios
- Consulta y petición de vacaciones
- Imputar fichajes y descansos
- Consulta de perfil

### 2.4.3 Preguntas para obtener información

- ¿El menú de la pantalla principal es de fácil acceso? ¿Crees que sería más cómodo si estuviera en la parte inferior?
- ¿Consideras que la información sobre los fichajes se presenta de forma intuitiva en la pantalla principal?
- ¿Consideras que el procedimiento para imputar fichajes es intuitivo?
- ¿Consideras que la consulta de vacaciones es intuitiva?
- ¿Consideras que la información sobre los horarios de los compañeros se presenta de forma clara?
- ¿Echas de menos algo de información en alguna de las pantallas?
- ¿Los textos son fácilmente legibles?
- ¿Los iconos elegidos para cada funcionalidad son representativos?

#### 2.4.4 Encuestas realizadas

- *¿El menú de la pantalla principal es de fácil acceso? ¿Crees que sería más cómodo si estuviera en la parte inferior?*

La mayoría de los usuarios coinciden que el menú debe estar en la parte inferior para que sea más cómodo.

- *¿Consideras que la información sobre los fichajes se presenta de forma intuitiva en la pantalla principal?*

La mayoría de los usuarios coincide en que la información es bastante clara e intuitiva.

- *¿Consideras que el procedimiento para imputar fichajes es intuitivo?*

Algunos usuarios comentan que el formulario es lo más importante y que debe mostrarse de forma clara. El resto indica que es bastante intuitivo.

- *¿Consideras que la consulta de vacaciones es intuitiva?*

La mayoría coinciden afirmativamente

- *¿Consideras que la información sobre los horarios de los compañeros se presenta de forma clara?*

La mayoría coinciden afirmativamente. Se hacen algunas correcciones con respecto a la visualización de los mensajes de información sobre técnicos de vacaciones, horarios pendientes de confirmar o indicadores de horario normal y retén.

- *¿Echas de menos algo de información en alguna de las pantallas?*

Se añade información al componente 'Perfil' como la ciudad o la antigüedad del técnico.

- *¿Los textos son fácilmente legibles?*

La mayoría coincide afirmativamente.

- *¿Los iconos elegidos para cada funcionalidad son representativos?*

Tras consultarlo con varios usuarios se utilizan los iconos de Ionic y se eligen los que se van a utilizar en la aplicación.

## 3. Casos de uso

### 3.1 Caso de uso "Login"

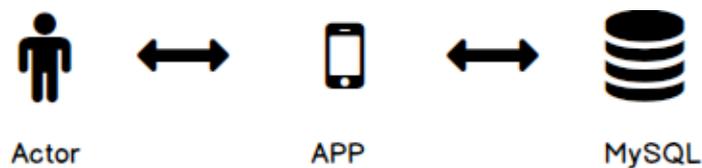
Prioridad → Alta

Descripción → La aplicación permitirá autenticarse contra TimeTrack

Precondiciones →

- El usuario debe tener descargada e instalada la aplicación en su dispositivo móvil.
- El usuario debe tener el dispositivo móvil conectado a internet.
- El usuario tiene abierta la aplicación.

Diagrama de flujo →



**Ilustración 4. Diagrama de flujo "Login"**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario abrirá la aplicación
- El usuario introducirá su usuario y su contraseña.
- La aplicación se conectará con la base de datos de TimeTrack para realizar el login con el usuario y contraseña proporcionados.
- La aplicación iniciará una sesión con el usuario logado.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar el login con el usuario y contraseña proporcionados.
- La aplicación muestra la pantalla de login con un error y la posibilidad de cambiar la contraseña.
- Se acaba el caso de uso.

Postcondiciones →

- La aplicación carga la pantalla principal.

### 3.2 Caso de uso “Imputar fichaje”

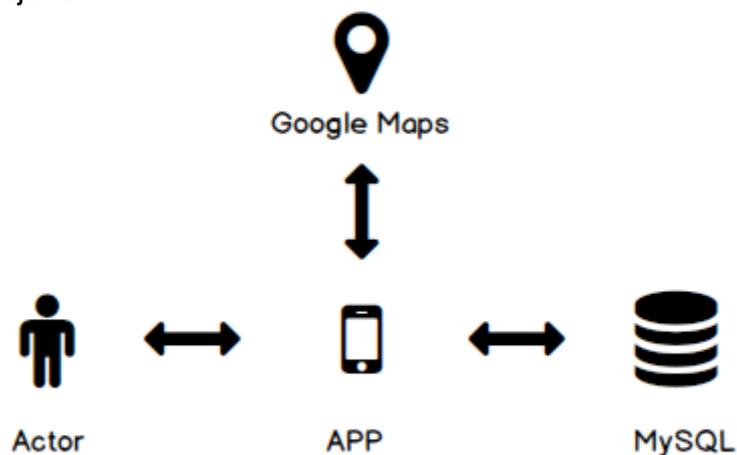
Prioridad → Alta

Descripción → La aplicación permitirá imputar un fichaje en TimeTrack

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla principal de la aplicación.

Diagrama de flujo →



**Ilustración 5. Diagrama de flujo “Imputar fichaje”**

- Actor: representa al usuario que se autentica en la aplicación.
- Google Maps: API de Google Maps
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para imputar fichaje.
- La aplicación conectará con Google Maps para localizar la ubicación del dispositivo móvil.
- La aplicación mostrará la pantalla de imputar fichaje con la ubicación y un formulario para rellenar los datos del fichaje.
- El usuario rellenará el formulario y pulsará sobre el icono para registrar el fichaje.
- La aplicación conecta con la base de datos de TimeTrack para registrar los datos.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar el registro en la base de datos.
- La aplicación muestra la pantalla de imputar fichaje con un error.
- Se acaba el caso de uso.

Postcondiciones →

- La aplicación carga la pantalla principal actualizada.

### 3.3 Caso de uso “Imputar descanso”

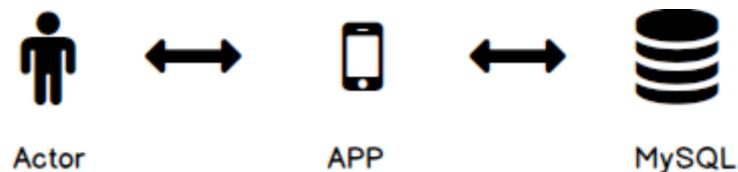
Prioridad → Baja

Descripción → La aplicación permitirá imputar un descanso en TimeTrack

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla principal de la aplicación.

Diagrama de flujo →



**Ilustración 6. Diagrama de flujo “Imputar descanso”**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para imputar fichaje.
- La aplicación mostrará la pantalla de imputar descanso con un formulario para rellenar los datos.
- El usuario rellenará el formulario y pulsará sobre el icono para registrar el descanso.
- La aplicación conecta con la base de datos de TimeTrack para registrar los datos.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar el registro en la base de datos.
- La aplicación muestra la pantalla principal
- Se acaba el caso de uso.

Postcondiciones →

- La aplicación carga la pantalla principal actualizada.

### 3.4 Caso de uso “Ver vacaciones”

Prioridad → Media

Descripción → La aplicación permite ver las vacaciones del resto de usuarios.

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla principal de la aplicación.

Diagrama de flujo →



**Ilustración 7. Diagrama de flujo “Ver vacaciones”**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para ver las vacaciones.
- La aplicación conecta con la base de datos de TimeTrack para obtener los datos.
- La aplicación mostrará la pantalla para ver las vacaciones con un calendario del mes actual.
- El usuario pulsará en cualquiera de los días del calendario.
- La aplicación mostrará debajo del calendario una lista con los usuarios que estén de vacaciones o permisos el día elegido.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar la conexión con la base de datos.
- La aplicación muestra la pantalla de ver las vacaciones con un error.
- Se acaba el caso de uso.

### 3.5 Caso de uso “Solicitar vacaciones”

Prioridad → Baja

Descripción → La aplicación permite solicitar vacaciones o permisos

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla de ver las vacaciones.

Diagrama de flujo →



**Ilustración 8. Diagrama de flujo “Solicitar vacaciones”**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para solicitar vacaciones.
- La aplicación mostrará la pantalla para solicitar vacaciones con un formulario.
- El usuario rellenará el formulario y pulsará en el icono para confirmar la petición.
- La aplicación conectará con la base de datos de TimeTrack para registrar los datos.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar el registro en la base de datos.
- La aplicación muestra la pantalla de ver las vacaciones con un error.
- Se acaba el caso de uso.

Postcondiciones →

- La aplicación mostrará la pantalla de ver las vacaciones actualizada.

### 3.6 Caso de uso “Ver horarios”

Prioridad → Media

Descripción → La aplicación permite ver los horarios de los usuarios

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla principal de la aplicación.

Diagrama de flujo →



**Ilustración 9. Diagrama de flujo “Ver horarios”**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para ver los horarios.
- La aplicación conectará con la base de datos de TimeTrack para obtener los datos.
- La aplicación mostrará la pantalla con los horarios de cada usuario.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar la conexión con la base de datos.
- La aplicación muestra la pantalla principal con un error.
- Se acaba el caso de uso.

### 3.7 Caso de uso “Ver perfil”

Prioridad → Media

Descripción → La aplicación permite ver el perfil del usuario logado

Precondiciones →

- El usuario está logado en la aplicación.
- El usuario está en la pantalla principal de la aplicación.

Diagrama de flujo →



**Ilustración 10. Diagrama de flujo “Ver perfil”**

- Actor: representa al usuario que se autentica en la aplicación.
- APP: aplicación.
- MySQL: base de datos de TimeTrack.

Flujo de éxito →

- El usuario pulsará sobre el icono para ver su perfil.
- La aplicación conectará con la base de datos de TimeTrack para obtener los datos.
- La aplicación mostrará la pantalla con el perfil del usuario.
- Se acaba el caso de uso.

Flujo alternativo de error →

- La aplicación no puede realizar la conexión con la base de datos.
- La aplicación muestra la pantalla principal con un error.
- Se acaba el caso de uso.

### 3.8 Diagrama de flujo general

A continuación, se presentan todos los casos de uso previamente identificados en un único diagrama de flujo:

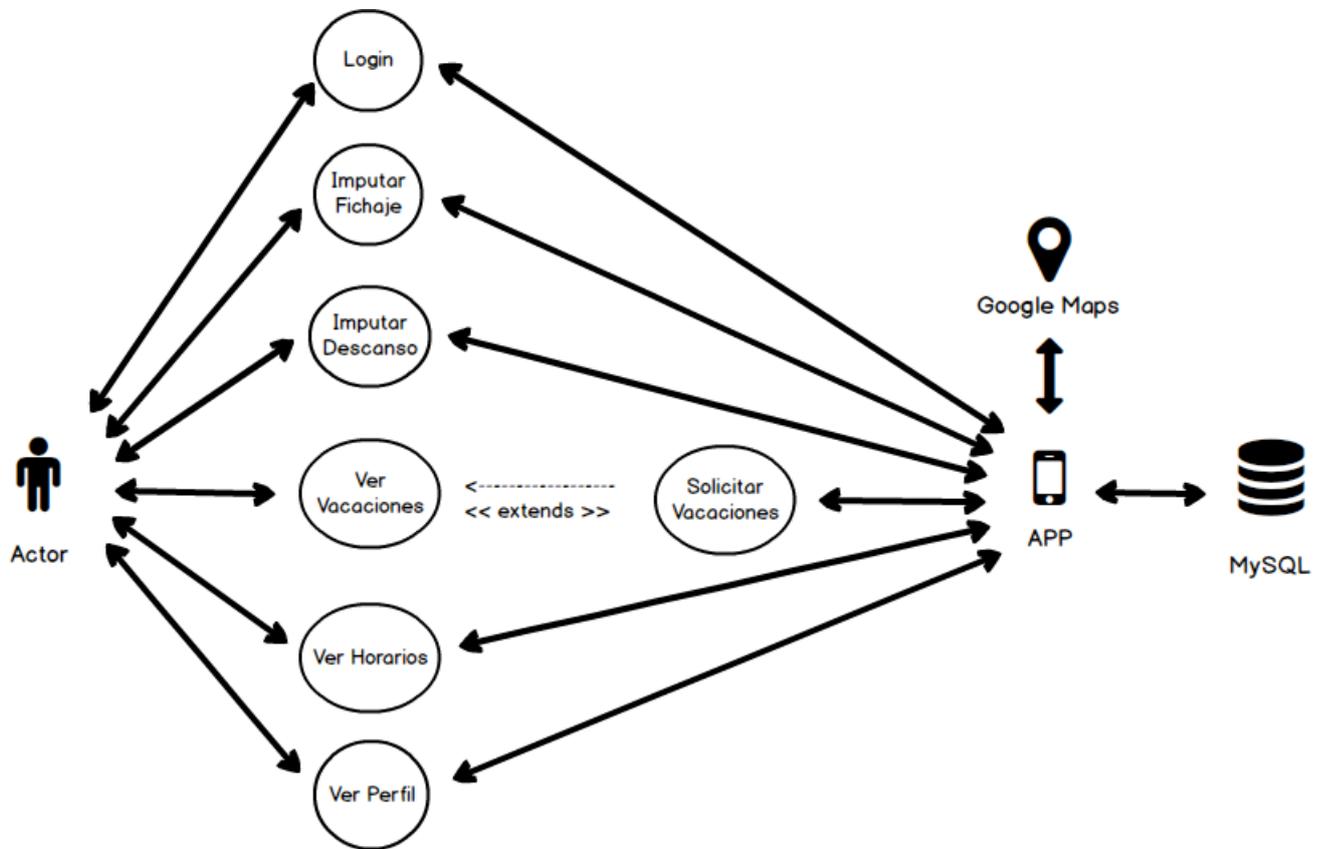


Ilustración 11. Diagrama de flujo general





## 4.2 Arquitectura del sistema

La aplicación desarrollada en este proyecto se integrará en el sistema que ya existe en la empresa IR Soluciones para completar la siguiente arquitectura:

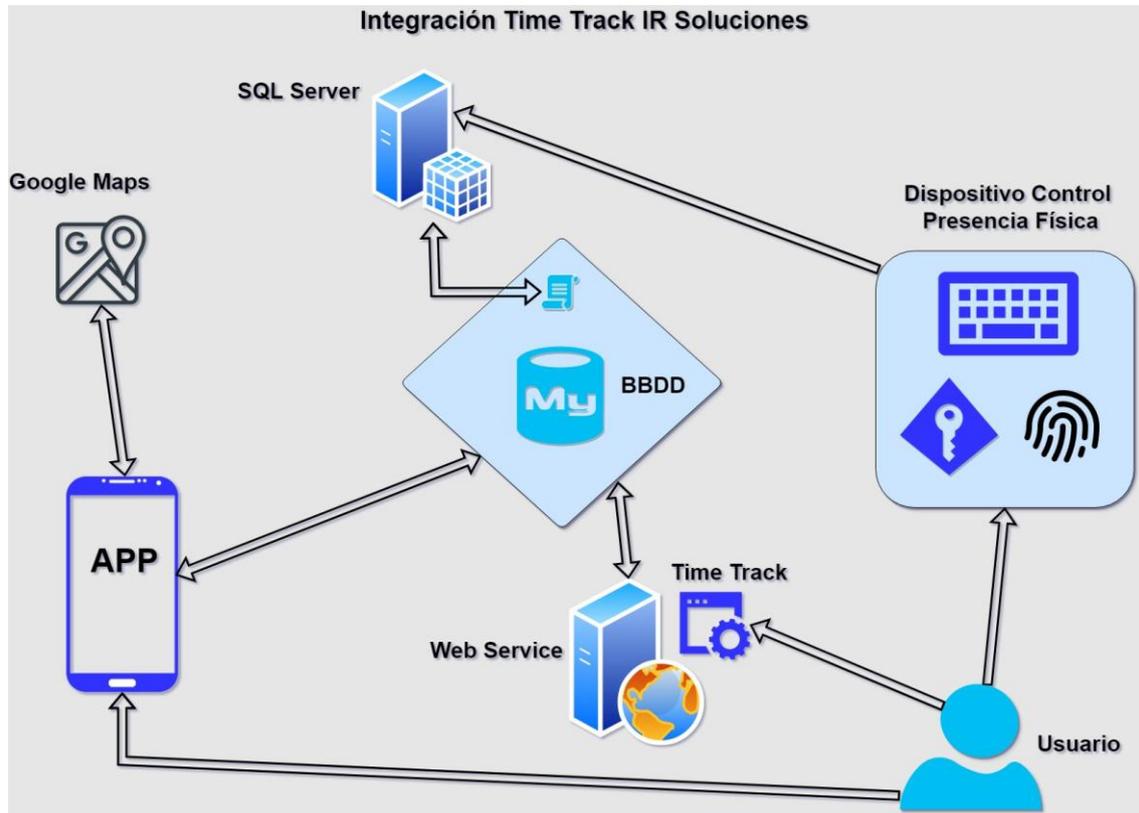


Ilustración 14. Diagrama con la arquitectura del sistema

Como se puede observar, el usuario del sistema tiene diferentes posibilidades para registrar un fichaje en la base de datos: el web service de TimeTrack, el dispositivo de control de presencia física y la aplicación móvil que se desarrolla en este proyecto.

La aplicación móvil registrará los fichajes y el resto de las funcionalidades en la base de datos MySQL, que comunica mediante un script con el SQL Server.

La aplicación, además, se comunica con Google Maps para obtener la geolocalización del dispositivo móvil en el momento de cada fichaje y poder registrarla en la base de datos.

Como dato aclaratorio, el dispositivo de control de presencia física únicamente registra el fichaje de entrada y salida del puesto de trabajo, sin posibilidad de implementar el resto de las funcionalidades, por este motivo se comunica directamente con el SQL Server.

## 5. Desarrollo de la aplicación

### 5.1 Herramientas y configuración de entorno

#### 5.1.1 JDK

Todo el desarrollo de este proyecto va a estar centrado en el lenguaje de programación JavaScript y desarrollado con las herramientas y utilidades que ofrece el Java Development Kit.

Tras comprobar multitud de problemas e incompatibilidades del resto de elementos del entorno con las últimas versiones de Java, finalmente se comprobó que para este proyecto lo mejor era utilizar JDK 8.



**Ilustración 15. Versión JDK**

Se puede descargar desde este enlace:

<https://www.oracle.com/java/technologies/javase/8u311-relnotes.html>

#### 5.1.2 Android SDK

Para poder exportar el proyecto a una aplicación Android “.apk”, es necesario tener instalado en el ordenador, además del JDK de Java, el SDK de Android.



La forma más fácil de instalarlo probablemente sea utilizando el SDK Manager del IDE Android Studio, del que hablaré más adelante.

Para este proyecto se utilizará la versión 32 de la API de Android y la versión 33 de las herramientas “build-tools” y “platform-tools” para el SDK de Android.

**Ilustración 16. Logotipo Android SDK**

Es necesario aceptar los contratos de licencia de Google, en este caso desde la consola de comandos, situándonos en el directorio “tools/bin” del SDK de Android e introduciendo el siguiente comando:

```
./sdkmanager --licenses
```

#### 5.1.4 Gradle

Gradle es una herramienta que permite la compilación de código abierto centrada en la flexibilidad y el rendimiento.

En este proyecto es necesaria Gradle para compilar la aplicación de Android mediante Cordova.

Se puede descargar desde la web: <https://gradle.org/install/>

#### 5.1.4 Variables de entorno

Para que todos los comandos y ejecuciones sobre JDK y Android SDK funcionen correctamente en el sistema operativo del ordenador que estemos usando para desarrollar la aplicación, es necesario configurar las variables de entorno. En este caso, sobre Windows:

ANDROID\_SDK\_ROOT → C:\Users\aitor.montalvo\AppData\Local\Android\Sdk  
ANDROID\_SDK\_HOME → C:\Android  
JAVA\_HOME → C:\Program Files\Java\jdk1.8.0\_311

Y añadir en "Path" lo siguiente:

```
C:\Program Files\Java\jdk1.8.0_311\bin  
C:\Users\aitor.montalvo\AppData\Local\Android\Sdk\tools\bin  
C:\Users\aitor.montalvo\AppData\Local\Android\Sdk\platform-tools  
C:\Users\aitor.montalvo\AppData\Local\Android\Sdk\emulator  
C:\Gradle\gradle-7.4.1\bin  
C:\Android
```

**Ilustración 17. Variables de entorno**

#### 5.1.5 Node.js y NPM

Para poder instalar y configurar el resto de los elementos que se utilizarán en el desarrollo de la aplicación ha sido necesario instalar previamente:

Node.js → es un entorno de ejecución para JavaScript de código abierto, construido con el motor de código abierto V8.

NPM → es un gestor de paquetes que permite instalar y gestionar versiones de paquetes y librerías JavaScript. Viene incluido en el entorno Node.js.



**Ilustración 18. Logotipo Node.js**

Como se puede observar en las ilustraciones, en este caso se han instalado la versión v16.14.2 de Node.js y la versión 8.5.0 de npm.

```
C:\Users\aitor.montalvo>npm -v  
8.5.0
```

```
C:\Users\aitor.montalvo>node -v  
v16.14.2
```

**Ilustración 19. Versiones de Node.js y NPM**

Se puede descargar desde este enlace:

<https://nodejs.org/es/download/>

### 5.1.6 Vue.js

Vue.js es un framework progresivo, accesible, versátil, reactivo, optimizado y bien documentado para construir interfaces de usuario con la librería principal enfocada en la capa de visualización.



Es una alternativa mucho más liviana y sencilla que otros frameworks parecidos como Angular. Tiene muchos plugins propios y mantenidos por la comunidad.

Para la instalación de Vue.js es necesario tener previamente instalado Node.js y utilizar el siguiente comando NPM:

```
npm install -g vue-cli
```

**Ilustración 20. Logotipo Vue**

### 5.1.7 Ionic 4

Ionic es un framework que permite el desarrollo de aplicaciones con tecnologías web, es decir, estándares abiertos altamente conocidos por la comunidad: HTML, CSS y Javascript.

Una de las ventajas fundamentales de Ionic y el desarrollo híbrido en general es que, al tratarse de tecnologías de desarrollo conocidas y usadas en el entorno web, cualquier desarrollador que trabaje habitualmente en internet puede reutilizar todo el conocimiento atesorado durante años.



**Ilustración 21. Logotipo Ionic**

Para la instalación de Ionic es necesario tener previamente instalado Node.js y utilizar el siguiente comando NPM:

```
npm install -g ionic
```

### 5.1.8 Creación del proyecto

Una vez instalados todos los elementos del entorno es el momento de crear el proyecto y ponerlos todos juntos en funcionamiento. Para ello es necesario seguir estos pasos:

#### Paso 1: Crear proyecto Vue.js

En este caso iniciamos un proyecto Vue.js basado en WebPack, una de sus plantillas para desarrollo de aplicaciones web, con este comando:

```
vue init webpack tfg
```

De todas las opciones que el proceso nos ofrece, es importante aceptar la instalación de vue-router, para crear rutas entre componentes de Vue, y no instalar ESLint, linter que examina el código JavaScript.

```
C:\Users\aitor.montalvo\WebstormProjects>vue init webpack tfg

? Project name tfg
? Project description TFG Aitor - IR Soluciones
? Author Aitor Montalvo Cano
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? No
? Set up unit tests No
? Setup e2e tests with Nightwatch? No
? Should we run `npm install` for you after the project has been created? (
recommended) npm

vue-cli · Generated "tfg".

# Installing project dependencies ...
# =====

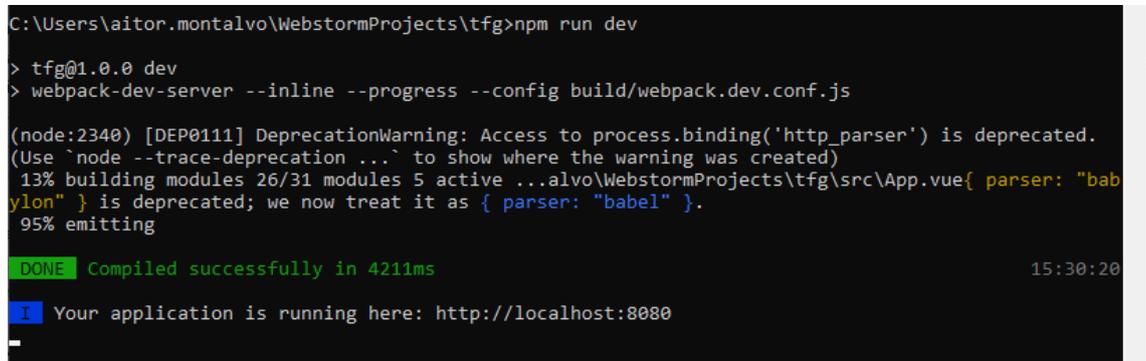
[ ] / idealTree:tfg: sill idealTree buildDeps
```

Ilustración 22. Creación proyecto Vue.js

Este proceso nos habrá generado una carpeta “tfg” con todos los archivos necesarios para empezar a construir la aplicación con Vue.js.

En este momento ya se podría lanzar la aplicación Vue a un servidor local (<http://localhost:8080>) situándonos en el directorio donde se haya creado la aplicación e introduciendo el siguiente comando:

`npm run dev`



```
C:\Users\aitor.montalvo\WebstormProjects\tfg>npm run dev
> tfg@1.0.0 dev
> webpack-dev-server --inline --progress --config build/webpack.dev.conf.js

(node:2340) [DEP0111] DeprecationWarning: Access to process.binding('http_parser') is deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
13% building modules 26/31 modules 5 active ...alvo\WebstormProjects\tfg\src\App.vue{ parser: "babel" } is deprecated; we now treat it as { parser: "babel" }.
95% emitting
DONE Compiled successfully in 4211ms 15:30:20
i Your application is running here: http://localhost:8080
```

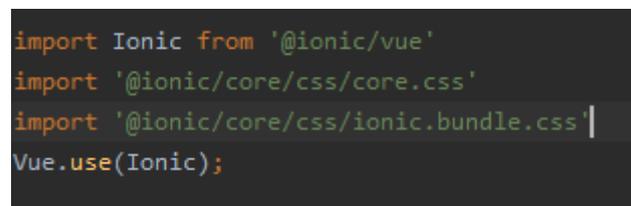
**Ilustración 23. Proyecto Vue.js compilado**

## Paso 2: Añadir Ionic al proyecto

Para poder utilizar las herramientas de Ionic necesitamos añadirlas al proyecto Vue que ya hemos generado. Para ello, la mejor forma es hacerlo es instalar el core de Ionic directamente dentro de la carpeta del proyecto Vue, de esta forma:

`npm install @ionic/vue @ionic/core`

Por último, añadir los import que hacen referencia a Ionic dentro del “main.js” del proyecto Vue.



```
import Ionic from '@ionic/vue'
import '@ionic/core/css/core.css'
import '@ionic/core/css/ionic.bundle.css'
Vue.use(Ionic);
```

**Ilustración 24. Imports Ionic dentro del “main.js” de Vue**

Además, es recomendable instalar también el siguiente paquete de iconos de Ionic, que se utilizarán en la aplicación.

`npm install ionicons@4.5.9-1 --save-dev`

### 5.1.9 Cordova

Cordova es una herramienta multiplataforma que permite construir y encapsular aplicaciones para sistemas móviles como Android, IOS o Windows Phone, a partir de lenguajes de programación web como HTML5, CSS3 y JavaScript.

Para el desarrollo de esta aplicación, integraremos Cordova dentro de nuestro proyecto Ionic+Vue.js para poder generar un “.apk” directamente y realizar las pruebas necesarias durante el desarrollo.



Ilustración 25. Logotipo Cordova

#### Paso 3: Integrar Ionic con Cordova

Lo primero de todo, es utilizar el siguiente comando para instalar el paquete de Cordova en nuestro ordenador:

```
npm install -g cordova
```

Para crear un proyecto Cordova, nos situamos dentro del directorio de nuestro proyecto e introducimos:

```
cordova create generarApk
```

Esto habrá creado una carpeta “generarApk” (se le puede llamar de otra forma) dentro de nuestro proyecto Vue+Ionic. Es necesario modificar los siguientes parámetros dentro del archivo “config/index.js” de nuestro proyecto:

```
build: {
  index: path.resolve(__dirname, '../www/index.html'),
  assetsRoot: path.resolve(__dirname, '../www'),
  assetsSubDirectory: 'static',
  assetsPublicPath: './',
}
```

Ilustración 26. Parámetros del proyecto en “config/index.js”

A partir de aquí, cada vez que se necesite generar un “.apk” de nuestra aplicación, necesitaremos construir primero la versión de producción desde la carpeta del proyecto Vue con el comando:

```
npm run build
```

Después, es necesario sustituir la carpeta “www” que está dentro de la carpeta “generarApk” por la homónima de nuestro proyecto Vue que acabamos de construir, y quedará configurado el núcleo de nuestra aplicación Android.

Para añadir la plataforma de Android en Cordova es necesario situarse en el directorio “generarApk” desde la consola e introducir el siguiente comando:

```
cordova platform add Android
```

Y finalmente:

```
cordova run Android
```

Esto nos proporcionará el archivo “.apk” en la ruta:

```
“C:\Users\aitor.montalvo\WebstormProjects\tfg\generarApk\platforms\android\app\build\outputs\apk\debug”
```

Este archivo lo podremos instalar para hacer pruebas en cualquier dispositivo Android o en un emulador Android, del que hablaré más adelante.

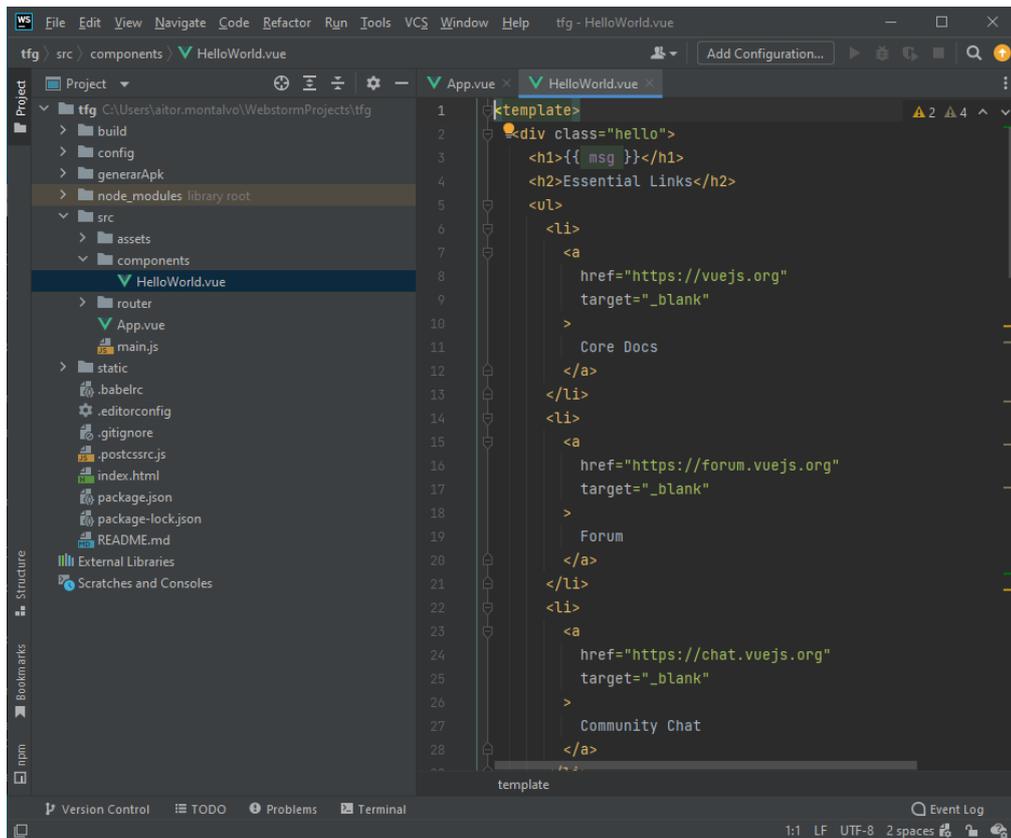
#### 5.1.10 JetBrains Webstorm



El IDE elegido para el desarrollo de la aplicación en este proyecto es Webstorm, de la empresa JetBrains, que dispone de un catálogo muy variado de IDEs enfocados a diferentes lenguajes como PhpStorm, Android Studio, IntelliJ, PyCharm o RubyMine, todos con similar apariencia y configuración.

**Ilustración 27. Logotipo Webstorm**

Una vez que ya se ha construido y configurado todo el entorno de trabajo con Java, Vue.js, Ionic y Cordova, se puede abrir un nuevo proyecto desde Webstorm y cargar todos los archivos generados previamente en el directorio “tfg” en el IDE para visualizarlos de forma estructural y poder comenzar a introducir código.



**Ilustración 28. Proyecto en IDE Webstorm**

Se puede ver en la captura como ya tenemos generado dentro del proyecto un componente de Vue con nombre “HelloWorld” que sirve como inicio de nuestro desarrollo y a partir del cual podemos empezar a construir la aplicación.

### 5.1.11 Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones Android y como ya he indicado anteriormente está basado en IntelliJ IDEA, otro IDE enfocado al desarrollo con Java. Ambos son desarrollados por la empresa JetBrains.

Para este proyecto, puesto que es necesario que la aplicación sea multiplataforma, no se usará Android Studio para el desarrollo del código, aunque se usará otro IDE de la misma empresa y similares características como Webstorm, del que ya he hablado.

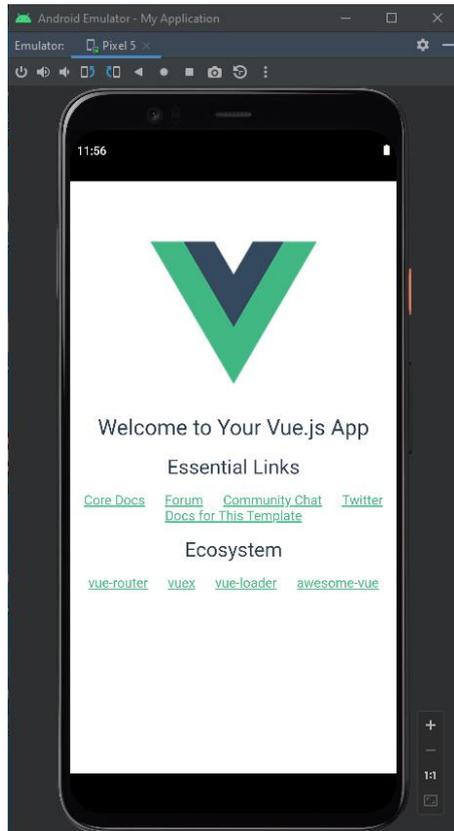


**Ilustración 29. Logotipo Android Studio**

El motivo por el cual es necesario mencionar a Android Studio en este proyecto, es que Android será el sistema en el que se harán las pruebas principales y este IDE además permite la descarga y configuración de muchas herramientas y complementos, además del propio SDK de Android, del que ya he hablado.

### 5.1.12 Android Virtual Device

A medida que se vaya desarrollando la aplicación, será necesario realizar pruebas en diferentes dispositivos móviles, para comprobar su correcto funcionamiento y detectar posibles errores.



Puesto que se ha configurado la opción de generar un “.apk” mediante Cordova, podremos instalarla directamente en cualquier dispositivo móvil o Tablet con Android, pero también existe la opción de utilizar un emulador de este sistema operativo en el propio ordenador en el que se está realizando el desarrollo, como, por ejemplo, el que nos proporciona el IDE Android Studio.

Para ello, se puede utilizar el Device Manager de Android Studio y configurar cualquiera de las opciones de simulación que permite. En este caso, se muestra un teléfono Google Pixel 5 de 6.0” con la versión 11 de Android, que también será necesario instalar desde Android Studio.

**Ilustración 30. Proyecto Vue en Android Virtual Device**

## 5.2 Código y programación

### 5.2.1 Enrutamiento

El paquete ‘vue-router’ (versión 3.0.1) de Vue que se ha utilizado en el proyecto se integra profundamente con el núcleo de Vue para facilitar la creación de páginas y permite hacer redirecciones entre componentes. Algunas características son:

- Mapeo de rutas anidadas
- Enrutamiento dinámico basado en componentes
- Efectos de transición
- Control de navegación detallado
- Enlaces con clases CSS activas automáticas
- Historial modo HTML5 o Hash
- Comportamiento de scroll personalizable
- Envío de propiedades entre componentes

El enrutamiento se haría de la siguiente forma desde el código del complemento origen:

```
this.$router.push({ name: 'Perfil' });
```

**Ilustración 31. Ejemplo de enrutamiento en Vue.js**

Un ejemplo del index.js del router de Vue con un par de complementos funcionando sería:

```
import Vue from 'vue'
import Router from 'vue-router'
import Login from '../components/Login'
import Perfil from '../components/Perfil'

Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/',
      name: 'Login',
      component: Login
    },
    {
      path: '/Perfil',
      name: 'Perfil',
      component: Perfil
    }
  ]
})
```

**Ilustración 32. Ejemplo del router de Vue.js**

### 5.2.2 Sesión de usuario

El paquete 'vue-session' (versión 1.0.0) de Vue permite generar y almacenar de forma local sesiones de usuario dentro de la aplicación.

Un ejemplo de creación de una sesión sería:

```
iniciarSesion(usuario) {
  this.$session.start();
  this.$session.set('usuario', usuario);
},
```

**Ilustración 33. Ejemplo de inicio de sesión en Vue.js**

Donde el parámetro 'usuario' que se pasa al método 'iniciarSesion' es un array con la información del usuario previamente recibida desde la base de datos MySQL.

## 5.2.2 Otras dependencias utilizadas

### 5.2.2.1 Integración de Google Maps

Para poder visualizar la ubicación del usuario en tiempo real y mostrarlo en un mapa de Google, se ha utilizado el paquete 'vue2-google-maps' (versión 0.10.7), que mediante una ApiKey de Google (la Key de Google utilizada es gratuita, únicamente para desarrollo) permite recibir unas coordenadas y crea el mapa con la ubicación dentro del contenedor especificado.

En este proyecto, se ha utilizado dentro del componente 'Fichaje' para mostrar la ubicación desde la cual el usuario está realizando el fichaje. La información del mapa es simplemente una mejora visual de la aplicación, que independientemente del mapa guarda las coordenadas de la ubicación en la base de datos.

Para su correcto funcionamiento es necesario que la aplicación tenga permiso de ubicación activado en el dispositivo móvil.

El componente se carga en el código de la siguiente forma:

```
<!-- GEOLOCALIZACION EN MAPA -->
<div id="mapa">
  <GmapMap
    :center='center'
    :zoom='16'
    style='width:100%; height: 25em;'
  >
  </GmapMap>
</div>
```

**Ilustración 34. Ejemplo de vue2-google-maps**

Y el método que recoge las coordenadas es el siguiente:

```
geolocate() {
  // EN ANDROID NECESITA PERMISO DE UBICACIÓN ACTIVADO
  navigator.geolocation.getCurrentPosition(position => {
    this.center = {
      lat: position.coords.latitude,
      lng: position.coords.longitude,
    };
  });
},
```

**Ilustración 35. Método de geolocalización**

### 5.2.2.2 Selectores de fecha y hora

En varias ocasiones, en este proyecto he necesitado utilizar selectores que me permitieran mostrar un calendario y elegir una fecha o un reloj y elegir una hora.

Por ejemplo, el componente principal de la aplicación 'Historial' utiliza 'vuejs-datepicker' (versión 1.6.2) como menú principal, el cual despliega un calendario, en este caso configurado para mostrar solamente los nombres de los meses, y permite navegar por ellos.

En el código se vería de la siguiente forma:

```
<datepicker
  @input ="cambiaMes('elegido') "
  input-class="vdp-datepicker"
  calendar-class="vdp-datepicker__calendar"
  :placeholder="getNombreMes(mesActivo)+' , '+anioActivo"
  name="selMes"
  id = "selMes"
  format = "MMMM, yyyy"
  :value="mesActivo+'/' +anioActivo"
  :language="es"
  :minimumView="'month'"
  :maximumView="'month'">

  <div slot="beforeCalendarHeader" class="calendar-
header">
    <strong>Elige un mes</strong>
  </div>
</datepicker>
```

**Ilustración 36. Ejemplo de vuejs-datepicker**

Otro caso, por ejemplo, es el del componente 'Fichaje', que usa 'vue-datetime' (versión 1.0.0-beta.14) para mostrar tanto el calendario como los relojes para elegir la fecha y las horas de entrada y salida de cada fichaje.

Un ejemplo del reloj de 'vue-datetime' se vería así en el código:

```
<datetime
  type="time"
  title="Hora de salida:"
  :phrases="{ok: 'Aceptar', cancel: 'Cancelar'}"
  :value="this.salida"
  ref="salida"
  input-id="salida"
  input-style=" text-align: right; width: 3em; margin:
1em; border:0; font-size: 1.2em;"
>
</datetime>
```

**Ilustración 37. Ejemplo de vue-datetime**

### 5.2.2.3 Calendario

Una de las funciones de la aplicación, es poder ver las vacaciones de cada técnico y para ello, el componente 'Calendario' usa 'vue-fullcalendar' (versión 1.0.9) para proporcionar un calendario navegable, en el cual además se cargan las vacaciones como eventos dentro de un array que se obtiene mediante una consulta a la base de datos.

Estos eventos, es decir, el array con las vacaciones de cada técnico, pueden ser de diferentes tipos (vacaciones normales, días libres por convenio y vacaciones de retén) y se muestran de diferentes colores dependiendo de cada uno de ellos.

En el código se llama al calendario de una etiqueta muy simple, ya que todo el código que lo genera está dentro del paquete instalado que, en este caso, también se ha editado para poder cargar diferentes tipos de vacaciones, y se vería de la siguiente forma:

```
<full-calendar :events="eventos" firstDay=1>
</full-calendar>
```

**Ilustración 38. Ejemplo de vue-fullcalendar**

### 5.2.3 API PHP

En este proyecto no se realiza ninguna conexión directa desde ninguno de los componentes de Vue para realizar consultas a la base de datos MySQL, si no que se realizan mediante una API programada en PHP que es quien se encarga de comunicar con la base de datos y devolver las respuestas al componente de Vue que la solicita.

Esta API está alojada en el servidor de la empresa IR Soluciones, que es con quien se comunica el proyecto, mediante una URL almacenada en una variable global en el 'main.js' del proyecto Vue.

Un ejemplo del funcionamiento de este proceso sería el siguiente:

- 1) El componente Vue genera un formulario y almacena en él los datos que se enviarán mediante POST.

```
let formdata = new FormData();
formdata.append('user', txUsuario);
formdata.append('pwd', txPwd);
```

**Ilustración 39. Ejemplo de creación de formulario en Vue.js**

- 2) El componente Vue realiza un POST, en el que llama a la API mediante la variable global que contiene la URL donde se aloja, añade a la URL la acción a realizar, que se envía mediante GET, y le pasa el formulario con los datos almacenados previamente.

```
this.$http.post(this.$url + 'login', formdata)
```

**Ilustración 40. Ejemplo de envío por POST en Vue.js**

- 3) La API recibe la llamada del proyecto Vue, realiza la conexión con la base de datos y, mediante un switch, realiza la función correspondiente a la acción recibida por GET.

```
$acc = $_GET['acc'];  
  
switch ($acc):  
    case 'insertar-tiempo':  
        ( ... )  
        break;  
    case 'eliminar-tiempo':  
        ( ... )  
        break;  
endswitch;
```

**Ilustración 41. Ejemplo del switch(\$acc) de la API**

- 4) La API, tras realizar la acción correspondiente, devuelve la información necesaria al proyecto Vue codificada mediante JSON. Dicha información, puede ser una simple respuesta booleana, como sucede en las acciones de eliminar o modificar datos, o un array de datos mucho más complejo como sucede en las acciones que devuelven registros al proyecto Vue.

```
$response->STATUS = 'ERROR';  
$response->ERROR = 'Datos de registro incorrectos';  
echo json_encode($response);
```

```
$tecnicos[$row->PK_ID_TECNICO] = array( ... );  
$response->DATA = $tecnicos;  
echo json_encode($response);
```

**Ilustración 42. Ejemplos de la respuesta de la API**

- 5) El componente del proyecto Vue que realizó la llamada, recibe la información devuelta por la API y realiza los métodos necesarios dependiendo en cada caso.

```
.then(result => result.data.STATUS === 'OK' ?  
this.iniciarSesion(result.data.DATA) : this.showToast(result.data.  
a.ERROR, 'danger'))
```

**Ilustración 43. Ejemplo de Vue.js recibiendo respuesta de la API**

## 5.2.4 Componentes Vue

A continuación, paso a detallar cada componente creado para este proyecto por separado. Para información más detallada se recomienda ver el anexo “Manual de usuario”.

### 5.2.1.1 Login

En este componente básicamente se presenta el logotipo de la empresa y un pequeño formulario que pide tu usuario y tu contraseña para poder acceder a la aplicación.

El ‘input’ donde se introduce la contraseña, oculta los caracteres introducidos por defecto y mediante un icono permite mostrarlos y volverlos a ocultar.

El componente llama a la API para comprobar el usuario y contraseña introducidos y si todo es correcto, crea la sesión con el perfil y la información del usuario y redirige al componente principal, ‘Historial’.

En este componente hay unas líneas de código comentado, ya que, en un primer momento se pensó en dar al usuario la opción de recuperar la contraseña en caso de ser olvidada, pero por el momento, la empresa ha decidido posponerlo para más tarde y pensar en una mejor forma de implementarlo. Es muy posible que en la entrega final del proyecto esté ya implementado.

### 5.2.1.2 Historial

Es el componente principal y núcleo de la aplicación, donde se puede ver el historial de los fichajes y navegar por ella.

Consta de un menú superior que permite navegar por los meses del año y un menú inferior que permite acceder al resto de funcionalidades de la aplicación.

En el contenido central del componente se presentan todos los fichajes que el usuario ha introducido previamente en la base de datos, ordenados por días y horas.

Vie, 6 May		Total: 08:33h				
	15:15				18:00	
	12:00				14:10	
	11:38				12:00	
	09:14				11:38	
	08:00				09:14	

Ilustración 44. Fichajes en el componente ‘Historial’

Como se puede observar en la imagen, cada día (generados con bucles v-for de Vue) muestran en la cabecera la información del día y el tiempo que se ha imputado en total ese día.

Cada línea del día es un fichaje imputado donde se muestra la hora de entrada y de salida (el icono verde y rojo solo se muestra en los fichajes que corresponden a tiempo trabajado), el dispositivo desde el cual se ha realizado el fichaje (lágrima, ordenador o dispositivo móvil) y en la parte central, el icono de una taza si el fichaje es un descanso para el café (cuyo tiempo no es computable y que además se resalta en azul), y un icono con forma de bocadillo, en caso de que el fichaje tenga asociado un comentario de la tarea realizada.

Además, se puede observar una línea naranja entre fichaje y fichaje, que llamaremos “vía del tren”, que aparece en caso de que entre uno y otro haya un espacio de tiempo sin fichajes imputados. Esta “vía del tren” debería aparecer normalmente en el descanso en el que los trabajadores salen a comer, por ejemplo, aunque también sirve de aviso en caso de que el usuario se haya equivocado al imputar los fichajes.

Con respecto al menú inferior, en la parte central se encuentra el acceso al perfil del usuario, que mostrará su foto en caso de que la haya subido al servidor de la empresa, un icono para acceder al componente ‘Calendario’ y otro para acceder al componente ‘Horarios’.



**Ilustración 45. Menú inferior del componente ‘Historial’**

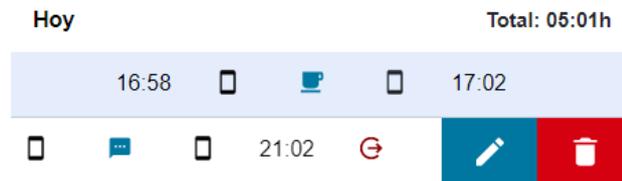
El menú inferior consta también de dos iconos flotantes, uno en cada esquina. En la derecha, un icono que abre un modal con el componente ‘Fichaje’ que permite imputar un nuevo fichaje, y en la izquierda un icono con el símbolo ‘Play’ (‘Stop’ cuando un fichaje está abierto) que despliega otros dos iconos (café y normal), que permiten directamente imputar un fichaje sin hora de salida, por lo que queda abierto y para diferenciarlo se indica con un GIF de un reloj girando.

Esta opción será útil para ir imputando los fichajes en tiempo real, mediante el ‘Play’ y el ‘Stop’.



**Ilustración 46. Ejemplo del GIF del reloj en fichaje abierto**

Otra función interesante implementada en este componente es que cada línea de cada fichaje imputado se puede deslizar hacia la izquierda, dejando ver así dos opciones más, una azul que abre un modal con el componente 'Fichaje' en modo edición, para modificar el fichaje en cuestión, y otra roja que permite eliminar el fichaje del todo, previa confirmación.



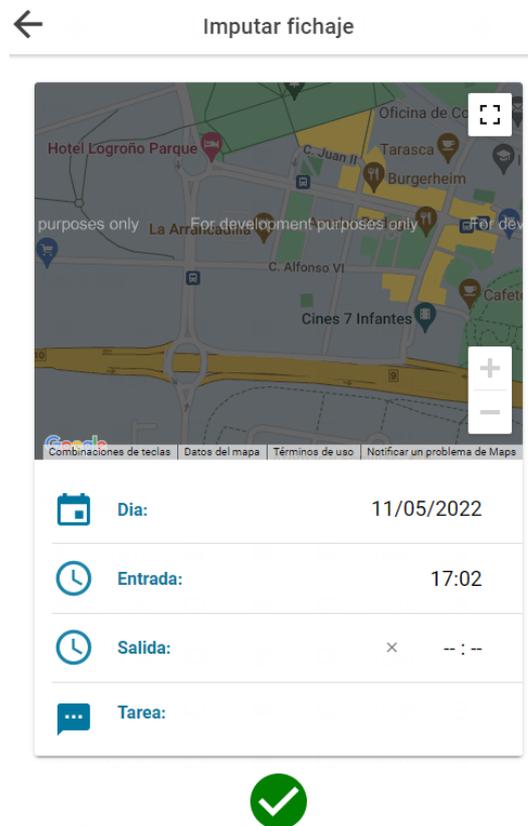
**Ilustración 47. Ejemplo de opciones de cada fichaje al deslizar**

### 5.2.1.3 Fichaje

Este componente se abre siempre en un modal, por lo que no se puede acceder a él mediante el router de Vue.

Básicamente es un formulario en el cual se puede elegir la fecha con un 'datepicker', la hora de entrada y la hora de salida con un 'timepicker' y añadir una descripción para la tarea realizada.

En la parte superior muestra un contenedor que dibuja un mapa de Google Maps, ubicando al usuario con las coordenadas obtenidas mediante geolocalización. Para que esto funcione correctamente, es necesario que el usuario de permisos a la aplicación para que pueda usar su ubicación (de momento manualmente).



**Ilustración 48. Modal del componente 'Fichaje'**

El mapa es meramente información visual, aunque el usuario se mueva por él o intente cambiar la información, las coordenadas que se guardarán serán las propias de la ubicación del usuario.

Si se acepta el formulario, el componente manda los datos del fichaje a la base de datos, incluidas las coordenadas, se guarda el registro, y regresa al componente 'Historial'.

Como curiosidad, es posible dejar 'vacío' (sin hora de salida) el fichaje para crearlo con el tiempo abierto, de la misma forma que el botón de 'Play' del componente 'Historial'.

#### 5.2.1.4 Perfil

En este componente se muestra básicamente la información del perfil del usuario como técnico de la empresa: su foto, en caso de que el usuario la haya subido previamente al servidor de la empresa, su usuario, su e-mail, la fecha de antigüedad, el equipo en el que trabaja y la ciudad donde trabaja.

Además, se da la opción al usuario de cerrar sesión, devolviéndole al componente 'Login' y de cambiar su contraseña, lo que abre un pequeño modal con el componente 'cambiarPwd'.

#### 5.2.1.5 CambiarPwd

Este componente, mencionado en el apartado anterior, se abre en un pequeño modal desde el componente 'Perfil' y permite al usuario mediante un pequeño formulario, cambiar su contraseña.

Al aceptar el formulario, se envían las contraseñas proporcionadas a la API, que hace las verificaciones correspondientes antes de cambiarla.

Al igual que en el componente 'Login', los campos permiten mostrar y ocultar los caracteres de las contraseñas.

#### 5.2.1.6 Horarios

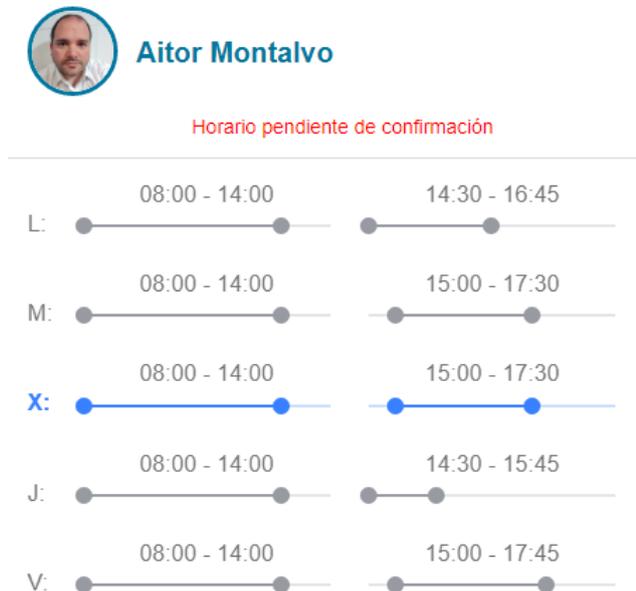
Este componente muestra los horarios de los técnicos que pertenecen a la empresa. Por defecto se mostrarán los técnicos que pertenecen al mismo equipo de trabajo que el usuario logado.

En el menú superior hay un primer selector, que permite elegir cambiar de equipo de trabajo y cargar los horarios de los técnicos del equipo seleccionado. Además, hay un segundo selector, que permite elegir a uno de los técnicos en concreto para filtrar los resultados.

Las fichas del contenido central están formadas por una cabecera con la foto, en caso de que el usuario la haya subido previamente al servidor de la empresa, el nombre del usuario y unos

indicadores en la parte derecha de color azul, que reflejan si el horario que se está mostrando es horario normal u horario de retén (guardia).

Las fichas de los técnicos que disponen de diferentes horarios, además de los indicadores, contienen un elemento deslizable, que permite cambiar de un tipo de horario a otro.



**Ilustración 49. Ejemplo de la ficha con los horarios**

Este apartado es el que más horas de programación ha llevado, ya que todo se carga dinámicamente en cada ficha para que los horarios se presenten visualmente de acuerdo con el día en el que se consulta. Se puede comprobar que el día actual se resalta de color azul, los 'ion-range' se recargan uno a uno con el valor del horario correspondiente, el 'slide' se mueve para mostrar el horario que el técnico tiene ese día (en caso de que tenga varios) y además se muestra información en la cabecera de la ficha en caso de que el horario esté sin confirmar por el responsable o que el técnico tenga vacaciones ese día.

Toda esta información se consulta previamente con la base de datos mediante llamadas a la API.

#### 5.2.1.7 Calendario

Este componente muestra un calendario generado con 'vue-fullcalendar', ya comentado previamente.

En la parte superior, al igual que en el componente 'Horarios', hay un selector que permite elegir el equipo de trabajo y recarga la

información de las vacaciones en el calendario. También se dispone de un botón flotante que permite abrir un modal con el componente 'peticionVacaciones'.

En la parte inferior se muestra una leyenda para identificar cada tipo de vacaciones.

Aunque el código del componente es aparentemente más sencillo, he tenido que modificar directamente los estilos y componentes del propio paquete instalado por NPM, para que pueda añadir los eventos de la forma requerida para este proyecto en concreto.

### 5.2.1.8 PeticionVacaciones

Este componente, mencionado en el apartado anterior, se abre en un pequeño modal desde el componente 'Calendario' y permite al usuario mediante un pequeño formulario, realizar una petición de vacaciones.

El formulario incluye dos 'datepicker' para seleccionar las fechas de inicio y final de las vacaciones.

Como curiosidad, las opciones para seleccionar el año al que pertenecen las vacaciones son dinámicas. Siempre se presenta en la parte central el año actual, en la parte izquierda el año anterior y en la derecha el siguiente. De esta forma no será necesario que el usuario escriba este dato.

Al aceptar el formulario, se envían los datos proporcionados a la API, que hace las verificaciones correspondientes antes registrar la petición.

<b>Tipo:</b>	<input checked="" type="radio"/> VACACIONES	<input type="radio"/> CONVENIO	<input type="radio"/> RETÉN
<b>Fecha desde:*</b>	dd/mm/aaaa		
<b>Fecha hasta:*</b>	dd/mm/aaaa		
<b>Año: ?</b>	<input type="radio"/> 2021	<input checked="" type="radio"/> 2022	<input type="radio"/> 2023
<b>En guardia: ?</b>	<input type="radio"/> SI	<input checked="" type="radio"/> NO	

Ilustración 50. Formulario de petición de vacaciones

## 5.3 Análisis del estado del proyecto y correcciones

### 5.3.1 Estado del proyecto

Actualmente, todas las funciones básicas que se pretendían implementar en un primer planteamiento del proyecto están implementadas y desarrolladas en la aplicación.

En cuanto al diseño, se ha respetado prácticamente en su totalidad, aunque se han cambiado algunos aspectos como por ejemplo el menú principal del componente 'Historial' que en un principio se situaba en la parte superior, pero tras pensarlo mejor y hablar con algunos de los usuarios, creo que en un dispositivo móvil que se maneja en la mano, es mucho más cómodo y accesible tener el menú de navegación en la parte inferior.

En resumen, estoy contento con el resultado obtenido hasta el momento y creo que el hecho de estar desarrollando la aplicación en la empresa IR Soluciones me ha servido para aprender mucho acerca de este entorno de trabajo y que el hecho de dedicar una jornada completa de 8 horas diarias me ha permitido cumplir los objetivos y poder estar enfocado plenamente en desarrollar la aplicación e investigar para implementar sus funcionalidades.

### 5.3.2 Correcciones

Con respecto a posibles correcciones, he detectado un pequeño mal funcionamiento que además es aleatorio en el segundo selector del componente 'Horarios', que filtra por técnico los resultados. A veces, al filtrar los resultados por técnico, algunos objetos de la ficha no se comportan de la misma forma que cuando se presentan todos los resultados en bucle, como si no se terminaran de cargar correctamente. Sin embargo, el comportamiento es aleatorio porque en la segunda o tercera elección se comportan correctamente. Intentaré revisar este comportamiento.

Sobre todo, me gustaría investigar un poco más acerca de la solicitud de permisos por parte de la aplicación, para que no sea necesario que el usuario active el permiso de ubicación manualmente, ya por el momento no he conseguido hacer que esto funcione, aunque si he encontrado información acerca de cuál es el código que hay que añadir, no tengo claro y no he conseguido encontrar en qué parte de mi proyecto (Vue, Ionic, Cordova...) hay que implementarlo exactamente para que funcione.

### 5.3.3 Funcionalidades extra

Además de los objetivos principales planteados inicialmente para este proyecto, se han implementado varias actualizaciones y funcionalidades extras interesantes para la aplicación. Algunas de ellas son:

- En el componente 'Historial', se añade una opción de "copiar" dentro de las opciones que se ofrecen al desplazar cada fichaje a la izquierda, añadiéndola a las ya existentes, "editar" y "eliminar".
- En el componente 'Historial', se añade una opción de "play" dentro de las opciones que se ofrecen al desplazar cada fichaje a la izquierda, añadiéndola a las ya existentes, "editar" y "eliminar". En este caso, sólo se debería mostrar si es el último fichaje del día, para que se pueda poner en "play".
- En el componente 'Historial', se implementa una "via del tren" de color roja, que se muestra cuando dos fichajes imputados tienen tiempos solapados. En principio esto no es posible ya que se realizan comprobaciones antes de imputar el fichaje, pero se implementa esta opción por si hubiera algún error o fallo de diseño.
- En el componente 'Historial', si se intenta imputar un fichaje mientras que el "play" está activo (hay un fichaje sin hora de salida), se pide confirmación para cerrarlo.
- En el componente 'Historial', en la cabecera de cada día, se añade la duración total de trabajo y la duración neta, sin tener en cuenta los descansos.
- En el componente 'Fichaje', se añade la posibilidad de imputar un fichaje como descanso, además de como fichaje normal.
- En el componente 'Calendario', se modifica la leyenda para que se puedan filtrar los eventos de las vacaciones por tipo para que sea más sencilla su visualización y búsqueda.
- En el componente 'Calendario', se muestran también las vacaciones pendientes de confirmación con un texto en rojo para diferenciarlas de las ya confirmadas.
- En la API PHP, al realizar una nueva petición de vacaciones, se comprueba que el rango de las fechas de la nueva petición no coincida con ningún registro que ya exista, se tiene en cuenta los días festivos y fines de semana para contabilizar los días de vacaciones solicitados y se verifica que el técnico tiene

días disponibles para pedir vacaciones haciendo un recuento de los registros existentes.

- En la API PHP, al imputar un nuevo fichaje desde “play”, se comprueba que las horas de entrada y salida de los diferentes tiempos no se solapen o si existen fichajes imputados posteriores a la hora actual.
- En el componente ‘Login’, se introduce un mensaje de error en caso de que el dispositivo no esté conectado a internet.

### 5.3.3 Vista al futuro

Muy probablemente, la aplicación desarrollada en este proyecto termine entrando en producción en la empresa IR Soluciones y para ello será necesario repasar todo el código de arriba abajo, incluso añadir nuevas funcionalidades.

De cara al futuro, además de las correcciones planteadas en el punto 5.3.2 de esta memoria, sería interesante tener en cuenta lo siguiente:

- En el componente ‘Login’, estudiar la posibilidad de que el usuario pueda recuperar su contraseña en caso de que se le haya olvidado. Pensaba implementarlo para este proyecto, pero al menos por el momento, hasta que la empresa encuentre la forma de hacer esto efectivo, no ha sido necesario.
- Revisar el código del paquete instalado ‘vue-fullcalendar’ para ver si de alguna manera se puede implementar en el componente ‘Calendario’ que se pinten de un color los días festivos en el calendario, ya que estos se guardan en la base de datos de TimeTrack.
- Añadir funcionalidades y vistas para diferentes perfiles de usuario, como por ejemplo administradores o jefes de proyecto. Algunas de ellas pueden ser, aceptar las peticiones de vacaciones o ver los fichajes del resto de trabajadores. Hasta el momento, todas las funcionalidades están diseñadas para un único perfil de usuario, el técnico.
- Implementar la posibilidad de realizar fichajes contra proyectos en concreto o clientes en concreto, que variarían dependiendo del usuario y su perfil. Hasta el momento, todos los fichajes que no son de descanso se realizan bajo la misma petición de trabajo.

- Implementar una solución que permita de alguna forma guardar los datos de inicio de sesión del usuario para que el login sea automático cada vez que se ejecuta la aplicación.
- Implementar el inicio de sesión en la aplicación mediante huella digital, utilizando el banco de huellas del dispositivo.

## 6. Pruebas

### 6.1 Descripción de las pruebas

Como ya se ha especificado en apartados anteriores, durante el desarrollo de la aplicación se han realizado pruebas de tres formas diferentes.

#### 6.1.1 Localhost

El desarrollo del código de la aplicación se ha realizado en Webstorm y tanto las pruebas del diseño como el control de errores que daba el código se han realizado a tiempo real y en una primera instancia de forma local.

Con el comando “npm run dev” se compila el proyecto y se lanza al servidor local “localhost:8080”, en el cual, a través de cualquier navegador, en este caso he utilizado Google Chrome, se puede probar la aplicación e interactuar con ella.

Además, se han utilizado tanto la consola como las propias herramientas que ofrece Google Chrome para desarrolladores para poder debuggear y observar el comportamiento de la aplicación en los momentos más complicados del desarrollo.

#### 6.1.2 Dispositivos móviles Android

Debido a que “localhost” se ejecuta en un navegador de un ordenador Windows, las pruebas que se realizan en este medio no son fieles al cien por cien, ni las pruebas de diseño, ni las pruebas de comportamiento.

Por esto, es necesario de vez en cuando comprobar que el comportamiento de la aplicación es el deseado también en un dispositivo móvil con sistema operativo Android.

En mi caso, dispongo de dos teléfonos OnePlus (versión 3 y 8 Pro) con diferentes resoluciones y versiones de Android, que me han servido para comprobar que tanto el funcionamiento como el diseño de la aplicación eran el deseado.

#### 6.1.3 Emulador Android

Como ya se ha comentado anteriormente en el apartado 5.1.11, se ha utilizado también el emulador Android Virtual Device que viene en Android Studio para realizar pruebas en dispositivo virtual, instalando la aplicación en un teléfono Google Pixel 5 de 6.0” con la versión 11 de Android.

## 6.2 Pruebas unitarias

Una de las pruebas que he repetido en varias ocasiones, por ejemplo, es pedir a diferentes técnicos y futuros usuarios de la aplicación, que la instalen en sus dispositivos, y que comprueben si todo funciona correctamente, si ven algo que les llame la atención, si consideran que echan algo en falta o que algo no funciona como debería.

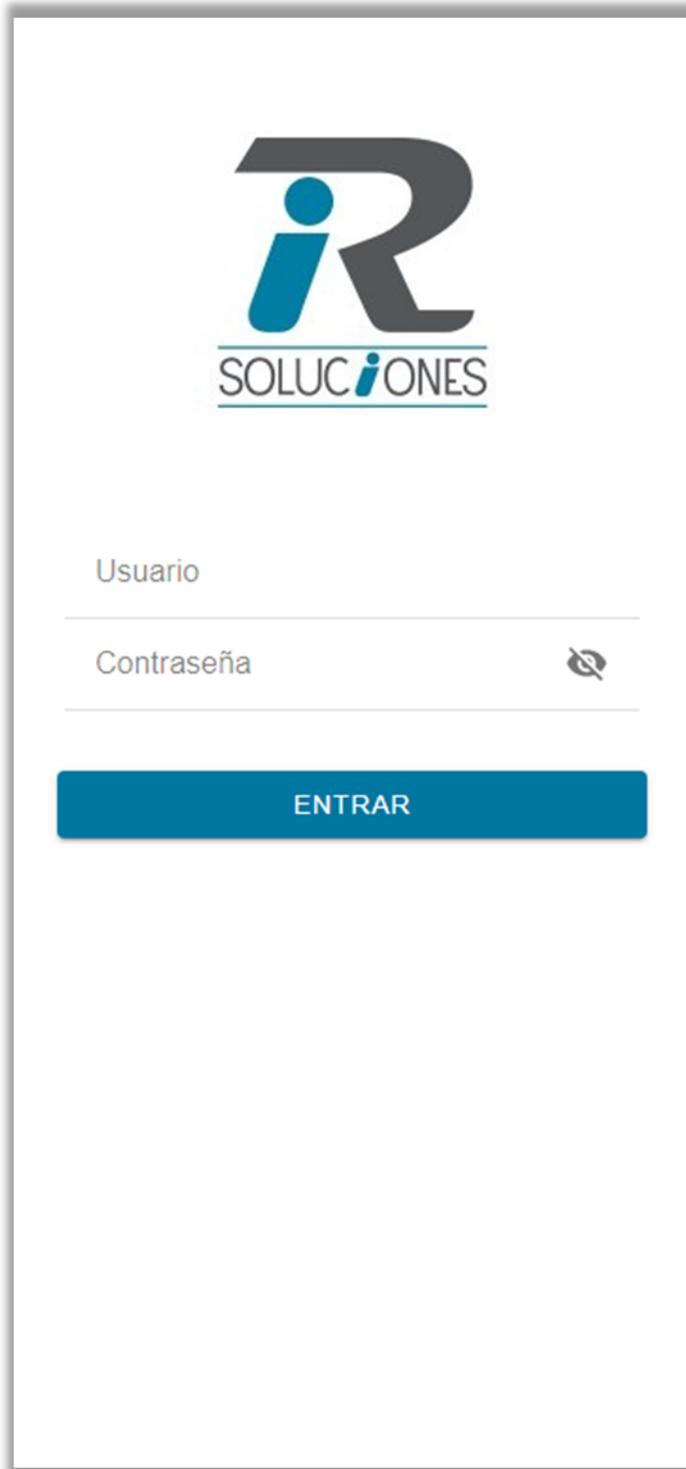
No se han realizado pruebas en específico de cada funcionalidad, si no que el desarrollo y las pruebas iban de la mano, diariamente mediante prueba y error hasta que cada funcionalidad estaba en el punto deseado.

No se han realizado pruebas unitarias como tal para este proyecto.

## 7. Aspecto final de la aplicación

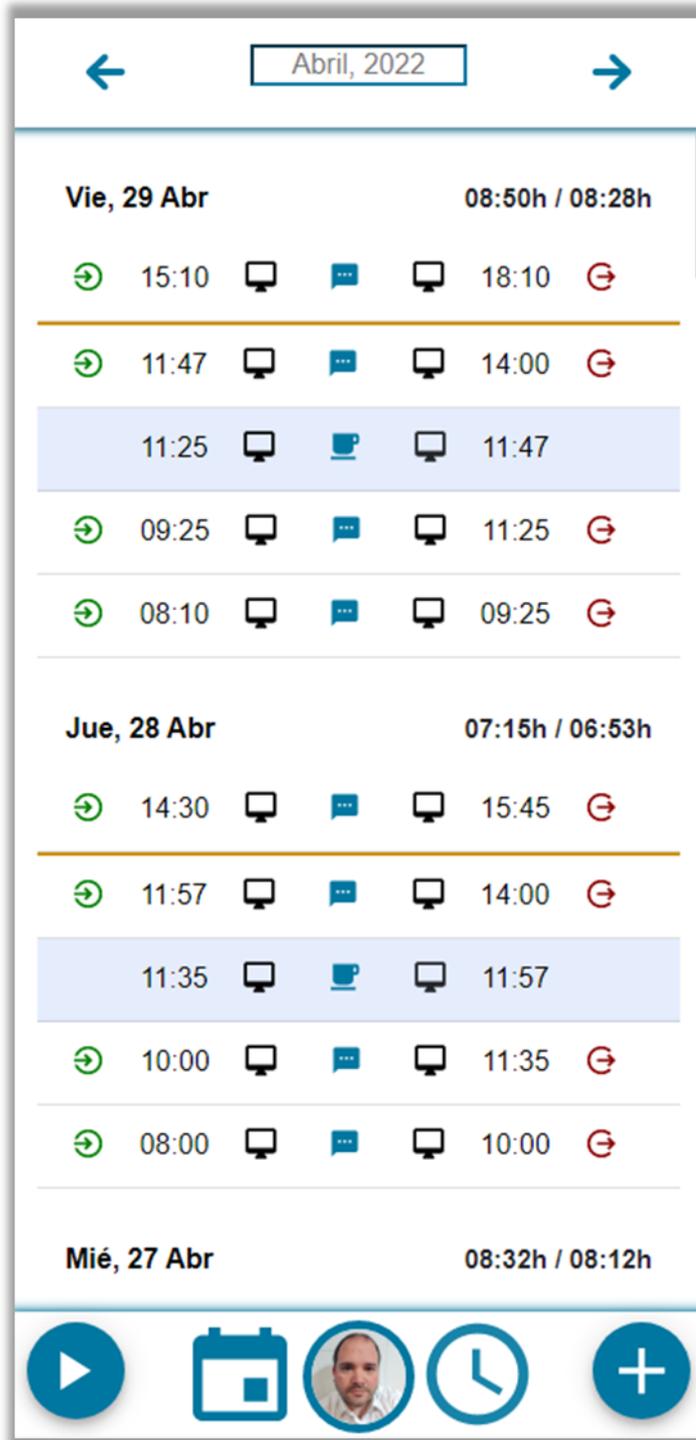
A continuación, se presentan capturas con el aspecto de la aplicación una vez finalizado el proyecto.

- Pantalla de Login



The screenshot shows a mobile application login screen. At the top center is the logo for 'iR SOLUCIONES', featuring a stylized 'i' and 'R' in blue and grey, with the word 'SOLUCIONES' in blue below it. Below the logo are two input fields: 'Usuario' and 'Contraseña'. The 'Contraseña' field has a small eye icon to its right, indicating a toggle for password visibility. Below the input fields is a large blue button with the text 'ENTRAR' in white capital letters.

- Pantalla de Historial



- Pantalla de Horarios

← Software 2 - Ruben ▾ Tecnico ▾



**Ruben Alutiz**

**NORMAL**

RETÉN

L:	08:15 - 14:15	15:45 - 18:15
M:	08:35 - 14:05	15:45 - 18:15
X:	08:35 - 14:15	
J:	08:35 - 14:05	15:45 - 18:15
V:	08:15 - 14:20	



**Carlos De la Torre**

**NORMAL**

RETÉN

L:	08:45 - 14:00	14:30 - 17:30
M:	08:20 - 14:00	14:30 - 17:30

The image shows two screenshots of a scheduling interface. The top screenshot is for 'Ruben Alutiz', a technician in 'Software 2'. It displays his schedule for the week of Monday to Saturday. The schedule is shown as horizontal bars with start and end times. For Monday (L), the schedule is 08:15 - 14:15 and 15:45 - 18:15. For Tuesday (M), it is 08:35 - 14:05 and 15:45 - 18:15. For Wednesday (X), it is 08:35 - 14:15. For Thursday (J), it is 08:35 - 14:05 and 15:45 - 18:15. For Friday (V), it is 08:15 - 14:20. The interface includes a back arrow, a dropdown menu for 'Software 2 - Ruben', another dropdown for 'Tecnico', a profile picture, the name 'Ruben Alutiz', and two buttons: 'NORMAL' (highlighted in dark blue) and 'RETÉN' (light blue). The bottom screenshot is for 'Carlos De la Torre'. It displays his schedule for Monday and Tuesday. For Monday (L), the schedule is 08:45 - 14:00 and 14:30 - 17:30. For Tuesday (M), it is 08:20 - 14:00 and 14:30 - 17:30. The interface includes a profile picture, the name 'Carlos De la Torre', and two buttons: 'NORMAL' (highlighted in dark blue) and 'RETÉN' (light blue).

- Pantalla de Vacaciones

← Software 2 - Ruben ▾

⊕

<<< Mayo 2022 >>>

L	M	X	J	V	S	D
25	26	27	28	29	30	1
David D						
			Laura F			
2	3	4	5	6	7	8
David D						
	Jose G					
9	10	11	12	13	14	15
David D						
16	17	18	19	20	21	22
David D				Laura F		
23	24	25	26	27	28	29
Raul P						
30	31	1	2	3	4	5

Retén
  Vacs
  Libre
  Guardia
  Pte

- Pantalla de Perfil

←



**Aitor Montalvo**

**Login:** amontalvo

---

**E-mail:** aitor.montalvo@irsoluciones.com

---

**Incorporación:** 2022-03-10

---

**Equipo:** Software 2 - Ruben

---

**Ciudad:** Logroño

---

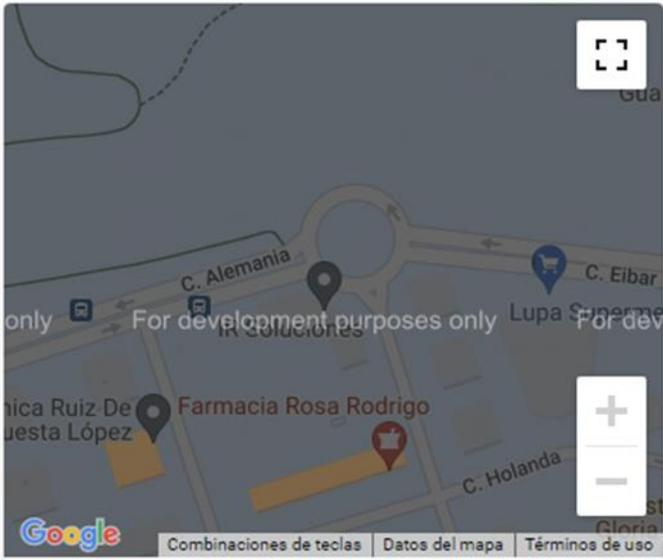
**Cambiar contraseña**

**Cerrar sesion** ↻



- Pantalla de Fichaje

← Imputar fichaje



Google Combinaciones de teclas Datos del mapa Términos de uso

**Tipo:**

**Dia:** 16/05/2022

**Entrada:** 13:18

**Salida:** × -- : --

**Tarea:**

✓

## 8. Conclusiones

Como resumen, estoy bastante satisfecho con el resultado del desarrollo de la aplicación, de mi estancia en la empresa IR Soluciones y en general, de este TFG.

Valoro muy positivamente la elección que realicé hace unos meses de pedir una excedencia en mi trabajo para poder realizar la estancia en esta empresa y poder así dedicarle una jornada laboral completa al proyecto y al TFG. Me ha servido para empaparme de una metodología de trabajo profesional y aplicarla al desarrollo de la aplicación.

En cuanto a los objetivos, creo que el resultado también es bastante positivo. Se han logrado cumplir todos los objetivos iniciales críticos y no prioritarios y se han implementado bastantes funcionalidades extras en la aplicación, aunque es cierto que se pueden todavía pulir muchas de ellas.

En cuanto a la metodología, creo que desde el primer momento se ha planteado correctamente y se han seguido las pautas marcadas por la UOC para las entregas, lo cual ha facilitado el correcto desarrollo del proyecto. Además, el hecho de realizar el TFG en un entorno laboral ha permitido también realizar un seguimiento por parte de la empresa que ha ayudado a garantizar el éxito del proyecto.

En cuanto al futuro, como se ha comentado en el apartado 5.3.3 de esta memoria, existe una gran posibilidad de que la aplicación desarrollada entre en producción en la empresa IR Soluciones, para lo cual, será necesario revisarla completamente y además, estudiar la posibilidad de añadir nuevas funcionalidades como por ejemplo que la aplicación pida permisos de ubicación, que se pueda guardar la contraseña en local para realizar el login automáticamente o incluso que se pueda realizar el login mediante huella digital.

Además, por parte de la empresa, también se me ha ofrecido la posibilidad de pasar a formar parte del equipo de programadores al finalizar la formación práctica.

En definitiva, creo que ha sido una experiencia muy positiva bajo cualquier punto de vista.

## 9. Glosario

IR Soluciones → empresa del sector TIC que necesita desarrollar la aplicación del proyecto, donde estoy realizando mi TFG y prácticas de formación.

TimeTrack → servicio web disponible en la empresa IR Soluciones para realizar las funcionalidades específicas detalladas en el proyecto y muchas más.

Scrum → metodología y marco de trabajo para desarrollo ágil de aplicaciones.

Java → lenguaje de programación creado por la empresa Sun Microsystems.

SDK → conjunto de herramientas de desarrollo de software para un sistema concreto.

JDK → conjunto de herramientas, utilidades, documentación y ejemplos para desarrollo de aplicaciones con Java.

JavaScript → lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Gradle → paquete de herramientas de compilación avanzadas que automatizan y administran el proceso de compilación.

V8 → motor de código abierto para JavaScript creado por la empresa Google.

Node.js → entorno de ejecución para JavaScript de código abierto, construido con V8.

NPM → gestor de paquetes y librerías para JavaScript.

Ionic → plataforma SDK de código abierto para desarrollo multiplataforma de aplicaciones móviles.

Framework → entorno o marco de trabajo con conceptos, prácticas y criterios estandarizados que sirve como referencia para resolución de problemas.

Vue.js → framework progresivo para construir interfaces de usuario con la librería principal enfocada en la capa de visualización.

Cordova → entorno de desarrollo de aplicaciones móviles multiplataforma de código abierto desde lenguajes estándar como HTML, CSS o JavaScript.

Lint → herramienta que examina el código y ayuda al desarrollador a corregir errores de sintaxis o malas prácticas.

Android → sistema operativo móvil basado en el núcleo Linux y en otros software de código abierto.

IOS → sistema operativo móvil creado por la empresa Apple para sus dispositivos que no permite su instalación en software de terceros.

APK → extensión de archivo correspondiente a un paquete de instalación para el sistema operativo Android. Variante del formato JAR de Java.

IDE → entorno de desarrollo integrado, aplicación informática que proporciona servicios integrales para facilitar el desarrollo de software.

WebStorm → IDE para JavaScript y tecnologías relacionadas creado por la empresa JetBrains.

CSS → lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

HTML → lenguaje de marcado para la elaboración de páginas web que permite indicar la estructura del documento mediante etiquetas.

PHP → lenguaje de código abierto adecuado para el desarrollo web que se ejecuta en servidor y puede ser incrustado en HTML.

PhpStorm → IDE para PHP y tecnologías relacionadas creado por la empresa JetBrains.

API → la interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos que se ofrece para ser utilizada por otro software con una capa de abstracción.

API Key → identificador único que permite autenticar solicitudes asociadas a diferentes servicios que comunican entre un proyecto y un servidor.

GoogleMaps → servidor de aplicaciones de mapas de Google que ofrece fotografías por satélite, navegación entre ubicaciones, imágenes a pie de calle, etc.

Array → estructura de datos utilizada en programación para almacenamiento contiguo que contiene una serie de elementos del mismo tipo, dentro de una matriz o vector.

MySQL → sistema de gestión de base de datos relacional de código abierto utilizada en entornos de desarrollo web.

URL → localizador de recursos uniforme. Identificador que se le da a un recurso único en la web para su direccionamiento.

POST → método de envío de información a través de una solicitud HTTP al servidor de manera no visible para el usuario.

GET → método de envío de información a través de parámetros añadidos a la URL de manera visible para el usuario.

Switch → mecanismo de control de selección utilizado en programación para permitir que el valor de una variable o expresión cambie el flujo de la ejecución del programa.

JSON → formato de texto sencillo pensado para el intercambio de datos cuya sintaxis está basada originalmente en JavaScript, pero independiente de cualquier lenguaje de programación.

GIF → formato gráfico digital utilizado en la web para imágenes o animaciones.

Modal → elemento o componente web que aparece sobre el contenido de la página o pantalla que se está visualizando.

Datepicker → elemento web que permite seleccionar una fecha.

Timepicker → elemento web que permite seleccionar una hora.

## 10. Bibliografía

NPM → <https://codingpotions.com/npm-tutorial>

V8 → [https://es.wikipedia.org/wiki/V8\\_\(Ingenieria\\_JavaScript\)](https://es.wikipedia.org/wiki/V8_(Ingenieria_JavaScript))

Node.js → <https://es.wikipedia.org/wiki/Node.js>

JavaScript → <https://es.wikipedia.org/wiki/JavaScript>

Gradle → <https://gradle.org/install/>

Ionic → <https://profile.es/blog/que-es-ionic/>

Vue.js → <https://es.vuejs.org/v2/guide/>

Android → <https://es.wikipedia.org/wiki/Android>

IOS → <https://es.wikipedia.org/wiki/IOS>

Webstorm → <https://www.jetbrains.com/es-es/webstorm/learn/>

Vue-router → <https://router.vuejs.org/introduction.html>

Vue-session → <https://www.npmjs.com/package/vue-session>

Vue2-google-maps → <https://www.npmjs.com/package/vue2-google-maps>

Vuejs-datepicker → <https://www.npmjs.com/package/vuejs-datepicker>

Vue-datetime → <https://www.npmjs.com/package/vue-datetime>

Vue-fullcalendar → <https://www.npmjs.com/package/vue-full-calendar>

## 11. Anexos

- Manuales
  - Instrucciones de uso del web service TimeTrack
  - Instrucciones de uso de la APP
  - Instrucciones para instalación y compilación
- Autoinforme de competencias transversales
- Gráficos:
  - Diagrama de arquitectura
  - Diagramas de flujo
  - Diagramas UML de la base de datos
  - Prototipos de alta fidelidad
  - Aspecto final de la APP
- Base de datos
  - Esquema HTML de la base de datos
- Código fuente comentado (tfg.zip)
- API PHP comentada (api.zip)
- APK con la APP empaquetada para Android (timetrack.apk)
- Video presentación del TFG y la APP