

# MEMORIA DE PROYECTO FIN DE GRADO

DESARROLLO DE APLICACIONES WEB

---



Escuela de profesiones digitales

Hugo García Flores

16/06/2022

---

# Índice

Abstract .....	2
Justificación del proyecto .....	3
Introducción .....	5
Objetivos .....	6
Desarrollo .....	6
Conclusiones .....	21
Bibliografía .....	22

# **ABSTRACT**

Llevar un registro sobre las horas de trabajo es esencial para evaluar las condiciones de trabajo de las personas empleadas. El acto de fichar en el trabajo y llevar un registro de la jornada laboral es obligatorio en todas las empresas, y un correcto sistema de fichaje es obligatorio por parte de las organizaciones. El presente trabajo busca desarrollar una aplicación para la gestión de las jornadas laborales, posibilitando una revisión completa de los horarios de cada miembro de la empresa, lo que ayuda a los responsables y líderes de equipo a acceder a la información de forma rápida, organizada y precisa.

En el documento se explican todos los aspectos relevantes en cuanto a la ejecución y clarificación del proyecto. Está dividido en siete partes, la primera trata sobre la justificación de la realización del proyecto la cuál está resumida al principio de este abstracto. Seguidamente, en la introducción, se escenifican las ideas principales en las que se basa el proyecto y luego explico los objetivos del mismo. Después se pasa al desarrollo, sección dividida en subsecciones donde se explican cada una de las pantallas de la aplicación, junto a sus componentes y su funcionamiento. Donde la función principal de la aplicación es controlar las jornadas laborales de los empleados que se encuentran dentro de una empresa para mantener constancia del tiempo que trabaja cada empleado.

Finalmente, se exponen las conclusiones y bibliografía.

## **Abstract en inglés**

An essential practice to evaluate the work conditions of the employees is to store the information regarding the number of hours dedicated to the development of their work. The action of clocking in and out of work is mandatory for every enterprise and having a correct system to manage this data is required for all organizations. The present project seeks the development of an application that helps and facilitates this registration by creating a correct and complete management system for each employee which helps the team leaders to access that information in a fast, precise, and organized way.

Inside the following document, all the relevant aspects regarding the execution and clarification of the given project are explained. It is divided into seven parts, first the justification of the realization of the project, which is summarized at the beginning of this abstract. Moreover, inside the introduction, the main ideas on which the project is based are staged; then I explain the objectives of the task. Furthermore, I have explained the methodology and development of the project, which is divided into subsections where every screen of the application is explained, with the corresponding components and functionalities. Where the principal functionality is to have control of the working hours of the employees who are within a company to keep track of the time each employee works.

Finally, the conclusions and bibliography are presented.

## **JUSTIFICACIÓN DEL PROYECTO**

En general se pueden identificar diferentes tipologías de actividades dentro de una organización: las productivas, que son las que forman parte de las operaciones de la organización y de su cadena de valor, los proyectos, los cuales responden a necesidades

puntuales, y las no productivas o de soporte, que son las necesarias para mantener y gestionar la organización.

En este caso nos encontramos con un proyecto que responde a una necesidad puntual como puede ser el control de los tiempos en los que un empleado está o no trabajando. Los motivos por los cuales he decidido la realización del proyecto son varios.

Para empezar, guarda relación con lo que se espera de un trabajador y en cómo se integra en el conjunto de una empresa. Esta cuestión la considero fundamental para el ámbito laboral en el que nos vamos a desenvolver a lo largo de nuestras vidas, y proporcionar medios que nos faciliten monitorear y controlar el trabajo, o más concretamente en este caso, las horas empleadas en el desarrollo del mismo por parte de los empleados, es fundamental para asegurar que cada empleado esté haciendo su función durante el tiempo correspondiente y que, en caso contrario, haya una justificación para ello.

La aplicación permite guardar un registro de cuántas horas se realizan una vez comenzada la jornada laboral, la cual se inicia en el momento en el que el operario inicia sesión en la aplicación y presiona al botón de “iniciar jornada”. Es similar a como cuando fichas en cualquier trabajo, con la ventaja de contar con muchas otras posibilidades que facilitan el seguimiento del trabajo del empleado. Se permite llevar la cuenta del número de ausencias dadas y el poder justificarlas en caso de contar con la razón de estas; también es posible registrar los momentos en los que se ausenta debido a pausas para comer o descansos asignados, entre otras posibilidades; también podremos revisar nuestras sesiones de trabajo, pudiendo ver tanto las horas de la jornada como la fecha de la misma.

En este la necesidad que llega a ocasionar este seguimiento se cubre bien con el propio proyecto y también gracias al avance tecnológico que permite desarrollar nuevos productos mejores y más económicos.

# **INTRODUCCIÓN**

La idea principal en la que se basa el proyecto es que se cumpla la ley de registro horario. Todas las empresas deben llevar el registro de horario de sus empleados. Los empleados y sindicatos deben conocer mensualmente tanto las horas realizadas, como las horas extras que cada empleado ha realizado.

La empresa está obligada a guardar el registro horario durante cuatro años y siempre debe estar disponible para los empleados y los sindicatos.

Otra de las ideas principales es aumentar la productividad de la empresa y reducir el absentismo mediante el control horario y gestión de turnos, gracias al registro de horas de trabajo de los empleados.

Además, los primeros beneficiados de tener que fichar en el trabajo son los empleados. Al registrar de manera diaria las horas que pasan en su puesto laboral, los trabajadores pueden conocer mejor cómo es su productividad y así poder planificar tanto su vida profesional como personal mejor.

Se estima que el sistema de control horario también influye en la autoestima del trabajador. Al saber que existe un control sobre su jornada laboral, el empleado tiende a ser más eficaz y productivo en su puesto laboral, redundando, por tanto, en algo beneficioso para la compañía para la que trabaja.

Gracias al sistema de control horario, la empresa puede conocer el horario preferido de los trabajadores y reubicarlos en el turno que prefieran, ya que la aplicación registra la hora en la que se inició la jornada. De esa manera se fomentará un buen ambiente laboral.

# **OBJETIVOS**

Los objetivos que tiene la aplicación son:

- Guardar un registro de cuando se comienza la jornada laboral.
- Registrar el número de horas trabajadas.
- Anotar las ausencias del empleado o empleada.
- Registrar los motivos de la ausencia.
- Registrar todas las sesiones de trabajo.
- Otros registros, como pueden ser descansos o pausas para comer.

Los objetivos personales que he alcanzado al desarrollar esta aplicación han sido:

- Aprender el desarrollo y los ciclos de vida de un proyecto real
- Aprender a realizar un código más limpio y organizado
- Hacer un código más seguro
- Trabajar a partir del código que ha hecho otra persona
- Aprender nuevas tecnologías como Angular, Ionic o .NET
- Aprender el desarrollo de una API y una mejor utilización de la misma

# **DESARROLLO**

Como esta sección será la más grande voy a dividirla en subsecciones donde iré explicando cada una de las pantallas de la aplicación, junto a sus componentes y su funcionamiento al detalle.

## **Resumen del funcionamiento de la aplicación**

La función principal de la aplicación es, como ya he detallado en las secciones anteriores del documento, controlar las jornadas laborales de los empleados que se encuentran dentro de una empresa para mantener constancia del tiempo que trabaja cada empleado y así saber si ha realizado sus horas asignadas o si ha trabajado menos de lo que le toca o, en cambio, ha realizado horas extra. De cada sesión de trabajo se guardarán las horas y minutos de dicha sesión, la hora a la que empezó y finalizó, así como la localización de la misma. Obviamente cada sesión va asignada al empleado que inició sesión al entrar a la aplicación, que a su vez está asignado a una empresa dentro de la misma.

El empleado también cuenta con un botón para pulsar cuando vaya a realizar un descanso, ya sea para comer u otra razón, mediante el cual podremos “poner en pausa” dicha sesión e iniciar una sesión de descanso que quedará registrada como una sesión a parte.

En la aplicación no solo podemos iniciar o finalizar una sesión sino que también el empleado podrá ver todas las sesiones de los últimos meses que ha realizado (fecha que es modificable, por lo tanto, podrá ver todas sus sesiones desde la fecha que él o ella desee).

También se podrán ver las ausencias que tiene programadas el empleado junto con su motivo, la fecha de la misma y si han sido aprobadas por su supervisor o no. Además, el empleado podrá crear una nueva ausencia, que más tarde deberá ser aprobada o no por su supervisor.

## **Tecnologías utilizadas en la aplicación**

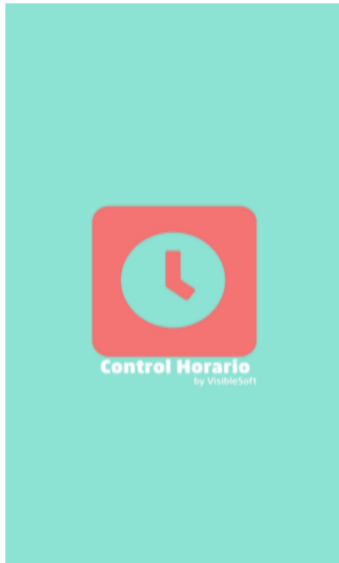
- AngularJS (framework web de JavaScript)
- Ionic (SDK para el desarrollo de aplicaciones móviles)
- SASS
- CSS3
- HTML5
- JavaScript (TypeScript)
- .NET



- SQL Server

## **Antes de entrar a la aplicación**

Antes de entrar a la aplicación nos encontramos con una pantalla con un color de fondo y un logo, aparte del nombre de la aplicación. A esto se le conoce como “Splash Screen”, la pantalla que sale antes de que se cargue la aplicación.

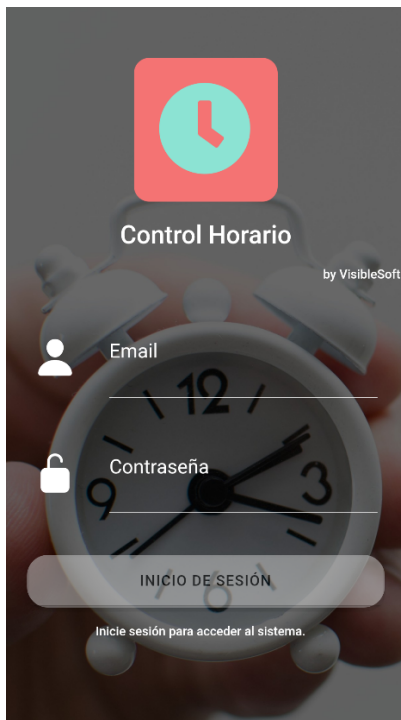


Para que salga esta splash screen en vez de la autogenerada por Ionic, hay que cambiar ciertos archivos en la carpeta .android ( y .ios si también lo quieres en dispositivos IOS). Para ello hay que entrar al directorio app/src/main/res y allí habrá que cambiar todas las imágenes de Splash autogeneradas, que se encuentran dentro de las carpetas llamadas drawable, por las nuestras propias que deberán ser del mismo tamaño que las originales. Esto hay que hacerlo por cada tamaño de dispositivo, que está asociado a cada una de las carpetas drawables.

## **Primera pantalla: Pantalla de Login**

La pantalla de Login sería la primera pantalla que nos encontraríamos al entrar a la aplicación y en un primer vistazo podemos ver, que el diseño de esta, cuenta con lo típico que te encontrarías en cualquier login, un formulario con dos campos, uno de email y otro

de contraseña, un botón para realizar la petición, una imagen de fondo y el logo de la empresa.



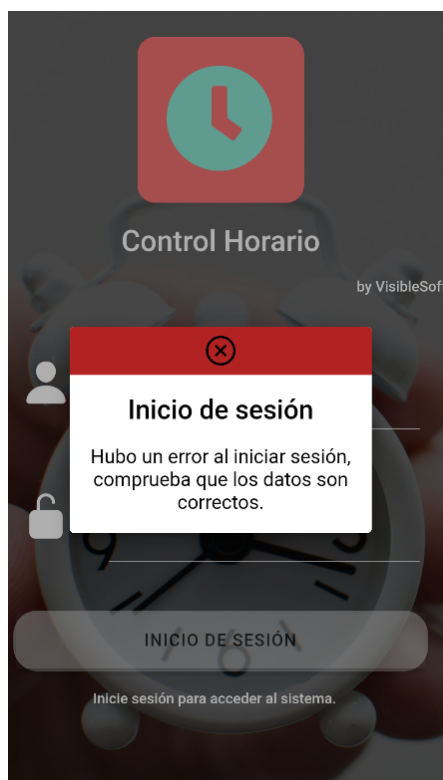
Ambos campos se tratan de ion-input validados tanto en la validación propia de HTML5, como la validación de formularios de Angular y una validación en el backend de la aplicación. Si el texto ingresado por el usuario no cumple con las validaciones del campo en cuestión, podremos apreciar como el color de texto de la etiqueta se vuelve de color rojo para ayudar al usuario a saber que tiene que cambiar lo que ha escrito en ese campo.

Como puede apreciarse en la imagen, el botón de Iniciar Sesión está deshabilitado hasta que el usuario haya ingresado en ambos campos un texto que cumpla con todas las validaciones, esto se logra gracias a la funcionalidad de Angular que te permite alterar atributos de las etiquetas HTML5 según el valor o valores de las propiedades del archivo de TypeScript que hayas asignado, en este caso, que el formulario sea válido.

Una vez ingresados todos los datos correctamente en sus campos, podremos darle al botón de Iniciar Sesión para entrar a la aplicación, lo que disparará la llamada a varias funciones para tratar con el funcionamiento del login.

Lo primero de todo lo que ocurrirá será que los valores ingresados por el usuario en los campos se asignarán a una clase llamada “usuario” que cuenta con las propiedades email y password.

Después de esto se realizarán varias peticiones HTTP al backend para comprobar que los datos sean correctos, siendo la primera de estas una petición de tipo POST, donde se le introducirá como cuerpo de la petición el email y la contraseña que ha ingresado el usuario al enviar el formulario. Si los datos son correctos se devolverá una respuesta de tipo 200 OK, y como cuerpo de la respuesta un JWT o JSON Web Token que permitirá al usuario entrar a la aplicación y realizar todas las acciones que se encuentran dentro de la misma, en caso contrario, si los datos son incorrectos, la petición devolverá un error 401 (unauthorized) y llamará a un componente de Ionic llamado PopOver, que te informará de que los datos que has ingresado no son correctos.



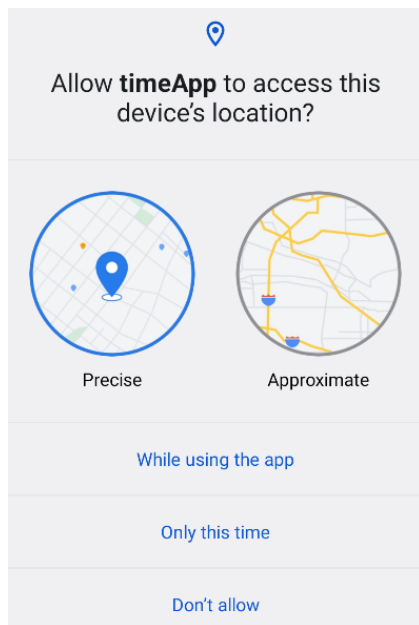
Una vez sabemos que los datos que ha ingresado el usuario son correctos, se realizarán otras dos peticiones HTTP al servidor. La que ocurre primero se llama store UserData, una petición de tipo GET que recibe como cuerpo de la petición el JWT que hemos recibido de la petición anterior, la de login, y nos devuelve los datos del usuario que ha iniciado sesión. Una vez contamos con estos datos, la misma función que llama a la petición HTTP almacena en la memoria temporal del dispositivo los datos del usuario, los datos del JWT y el JWT decodificado, también conocido como payload.

La siguiente petición que se realizará será una de tipo GET para obtener los, que yo he llamado parámetros, de la empresa del usuario que ha iniciado sesión, que son los siguientes:

```
"id": 16,  
"company": "prueba",  
"address": "Calle falsa 123",  
"city": "Madrid",  
"province": "Madrid",  
"zipCode": "28765",  
"cif": "A123456789",  
"vacationDays": 6,  
"geolocation": true,  
"restsRegistered": true,  
"navbarBackgroundColor": "#f4a42c",  
"navbarColorClass": "navbar-dark",  
"logo": "data:image/png;base64,iVBORw0KGAAACBjSFJNAAB6JgAAgIQAAAPoAAAAP8A/wD/oL2nkwAAJa1JREFUeNrtnQdY+P1pFjB2KQoaS6xRozF2QcTeokbALk0U
```

De todos ellos, de momento solo utilizaremos dos de ellos antes de avanzar a la pantalla principal de la aplicación, el llamado “geolocation” que nos informa sobre si la empresa quiere que se use la localización o no, y el “logo” que se trata del logo de la empresa en formato base64.

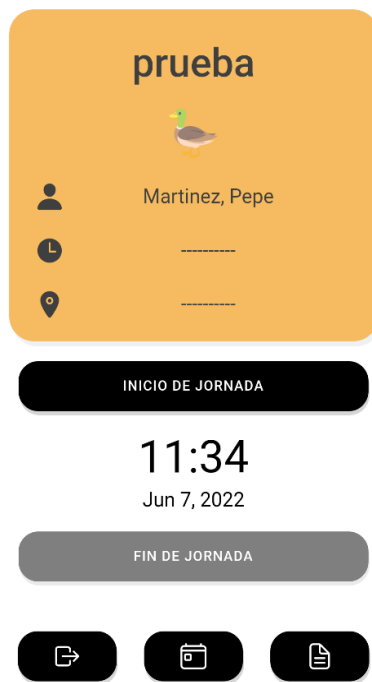
Una vez finalizada la petición anterior, se realizarán dos procesos, siendo el primero de ellos pedir al usuario los permisos para usar la localización de su dispositivo si es la primera vez que este ingresa a la aplicación. El segundo de ellos se trata de almacenar en la memoria temporal del dispositivo el logo de la empresa asociada con el usuario que inició sesión.



Una vez hecho todo esto, se avanzará al llamado “Home”, la pantalla principal de la aplicación.

## **Pantalla principal: Home Page**

La pantalla de Home es la pantalla principal de la aplicación y donde haremos la mayoría de los procesos contados en las primeras secciones del documento. Antes de entrar en detalle a cada una de sus funciones, así sería como la veríamos nada más haber iniciado sesión.



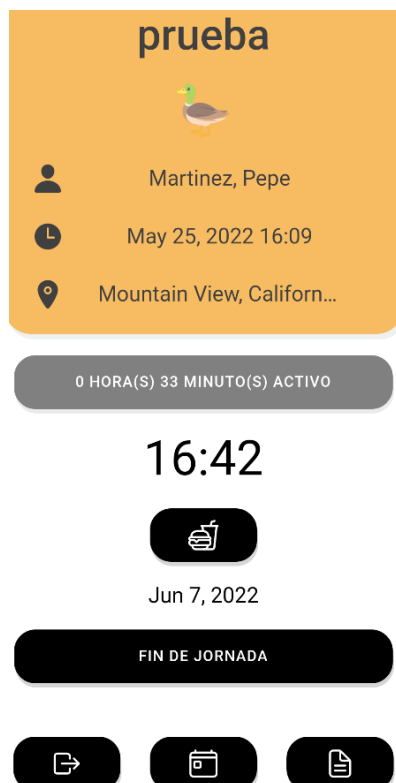
Aquí podemos ver varias cosas, la primera y más llamativa es el contenedor que, en este caso, es de color naranja. Aunque en este caso veamos el contenedor de color naranja y los botones y textos de color negro, hay que aclarar que estos colores son elegidos por la empresa, por lo que si iniciamos sesión con otro usuario de otra empresa distinta que tenga seleccionados otros colores distintos, estos cambiarían de color, pudiendo elegir como colores de botón entre blanco o negro y el color de contenedor, cualquier color.

Lo primero que vemos dentro de dicho contenedor es “prueba”, que se trata del nombre de la empresa a la que está asociado el usuario que inició sesión, parámetro el cual recibimos gracias a la petición HTTP realizada al hacer el login en la aplicación, al igual que su nombre, que sería el segundo texto que vemos y los colores del contenedor y los botones mencionados anteriormente. También cabe resaltar la imagen que, en este caso es un pato, que se encuentra muy a la vista en el contenedor. Esto se trata del logo de la empresa del usuario (si la empresa no tiene seleccionado un logo se le asignará el logo de la aplicación por defecto).

Antes de explicar el tercer y cuarto texto del contenedor, seguiré con el resto, pues la explicación de estos, vendrá más adelante.

En esta pantalla podemos distinguir cinco botones y un texto bastante llamativo en el centro, el cual se corresponde a la hora y fecha actuales, una información bastante relevante cuando se trata de una aplicación como esta, que sirve para controlar los horarios de los empleados.

Entre los cinco botones podemos ver dos más grandes que el resto, el botón de “Iniciar jornada” y el de “Finalizar jornada”, que se encuentra desactivado en este momento. Más abajo podemos ver, empezando desde la izquierda, el botón correspondiente al log out, que se encargaría de borrar la memoria temporal de nuestro dispositivo, la cual de momento contiene información como, la información del usuario o el JWT generado al hacer login y de devolvernos a la pantalla anterior, la de login. El segundo botón es el correspondiente a las ausencias, al interactuar con él podremos ver la pantalla de ausencias, que explicaré más adelante en el documento. Y el tercer botón también corresponde a otra pantalla, esta vez la de ver todas las sesiones que ha realizado el usuario desde la fecha que él desee, también la explicaré más adelante. Para seguir con la explicación iniciemos una jornada para ver qué ocurre.



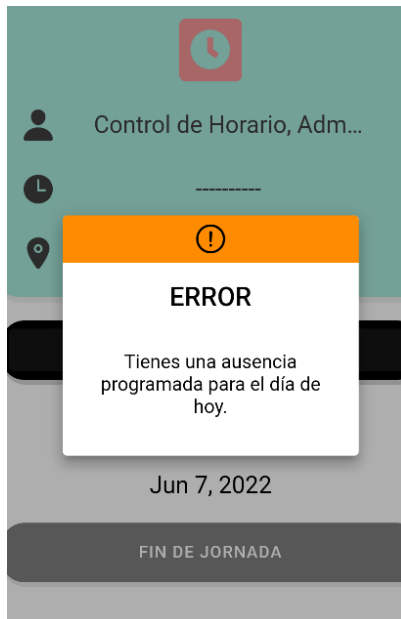
Como podemos apreciar en la imagen, los dos textos dentro del contenedor naranja que antes no se mostraban, han aparecido, eso es debido a que ambos textos hacen referencia a la fecha y la hora en la que se inició la jornada y la localización de la misma (en este caso vemos Estados Unidos en la localización porque es la geolocalización del emulador que estoy usando para mostrar la aplicación) si es que haya aceptado otorgar a la aplicación los permisos para acceder a su localización, en caso contrario aparecerá un texto que dice “Localización no disponible”.

También ha cambiado el texto del botón que anteriormente era el de “Iniciar Jornada” por la cantidad de tiempo que ha pasado desde que el usuario interactuó con el mismo, para así que sea sencillo llevar constancia del tiempo empleado trabajando. Todo esto se ha realizado gracias a una petición HTTP de tipo POST al servidor para crear una sesión, que debe recibir como cuerpo de la petición una sesión, que se trata de un objeto que tiene muchas propiedades como el nombre del usuario, su empresa, la fecha en la que comenzó, o la localización en la que se inició, entre otros.

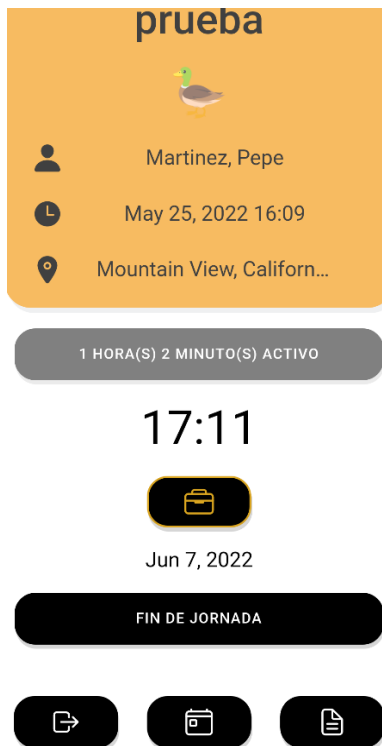
Una pregunta que podemos hacernos es; ¿Por qué aparece en el contador que llevamos 33 minutos en la sesión si acabamos de darle al botón de Iniciar Jornada?. Esto ocurre porque una vez interactuamos con el botón, el primer proceso que este realiza es llamar a una función que hace una petición HTTP de tipo GET al servidor para ver si ya hay una sesión activa de nuestro usuario, y en caso de que así sea, esta sesión continuará con el tiempo transcurrido en dicha sesión de trabajo en lugar de empezar una nueva. Esto puede ser de gran utilidad porque lo normal no es tener la aplicación activa durante toda la jornada laboral, si no que lo normal sería entrar dos o tres veces al día como mucho, que sería al entrar a trabajar, momento en el que iniciaríamos nuestra jornada y al terminar de trabajar, momento en el que finalizaremos nuestra sesión. Durante el tiempo transcurrido la aplicación puede perfectamente cerrarse o minimizarse y no pasaría nada ya que continuaría con la sesión que empezamos al principio del día.



También cabe detallar que si el usuario quiere iniciar sesión un día que tiene una ausencia programada y aprobada por su supervisor, no podrá iniciar sesión y le saltará un PopOver que le informe de que tiene una ausencia programada para ese mismo día.





También podemos ver que ha aparecido un nuevo botón en el centro, que se corresponde con la opción para informar a la aplicación de que vamos a tomarnos un descanso, ya sea para comer o cualquier otra actividad. Una vez hagamos click en él, este cambiará de apariencia para informarnos que estamos en un descanso.. Una vez esto haya ocurrido se iniciará una nueva sesión de la misma manera que la anterior pero de otro tipo, llamada “breakSession”, que funciona de la misma manera que una sesión normal solo que contabilizará el tiempo que nos hemos tomado de descanso. Para finalizar la sesión de descanso habrá que volver a interactuar con el botón.



El siguiente botón que vemos es el de “Fin de jornada” que terminará nuestra jornada, guardando esta en la base de datos mediante una petición HTTP de tipo POST pasando como cuerpo de la petición la sesión que hemos realizado con todas sus propiedades y devolviendo la pantalla principal a su estado anterior, como la vimos nada más iniciar sesión en la aplicación.

## **Pantalla de ausencias**

La pantalla de ausencias es la cual accedemos al interactuar con el botón que tiene el icono del calendario en la pantalla principal y nos sirve para poder ver las ausencias del usuario que inició sesión en la aplicación, ya estén aprobadas o no por su supervisor, y añadir ausencias nuevas y el motivo de ellas. La pantalla está dividida en dos partes, la parte superior donde podremos añadir nuevas ausencias mediante el uso de un formulario, y la parte inferior donde podremos ver todas las ausencias que hemos ido acumulando durante el último año.

 Ausencias 

---


**Fecha de la ausencia**

Fecha de inicio

Fecha de fin

**Motivo**

-Seleccione motivo de la ausencia-- ▾

 GUARDAR

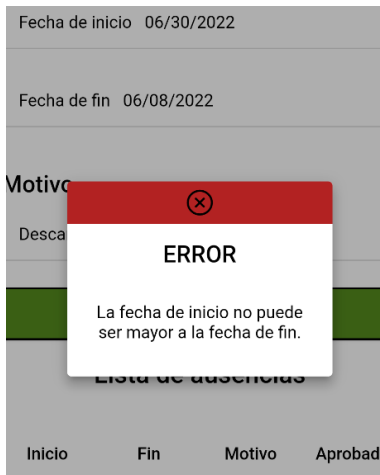
---

**Lista de ausencias**

Inicio	Fin	Motivo	Aprobada
01/07/2022	02/07/2022	Descanso	<input type="checkbox"/>
23/06/2022	24/06/2022	Descanso	<input checked="" type="checkbox"/>
29/07/2022	30/07/2022	Vacaciones	<input type="checkbox"/>

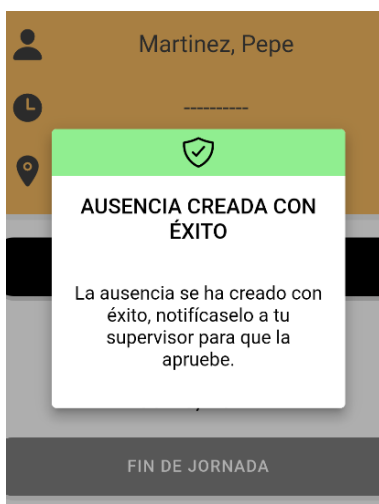
Como podemos ver arriba del todo contamos con una cabecera que cuenta con el logo de la empresa alineado a la izquierda, un título en el centro para saber que estamos en la pantalla de ausencias y un botón de cerrar a la derecha mediante el cual podremos volver a navegar a la pantalla principal de la aplicación.

Lo siguiente que vemos es el formulario para crear una nueva ausencia, el cual cuenta con tres campos; la fecha de inicio, la fecha de fin y el motivo de la ausencia. Debajo de estos se encuentra el botón para guardar la ausencia y enviar el formulario, que hasta que no hayamos rellenado todos los campos del formulario, se encontrará deshabilitado y no podremos pulsarlo. Los dos primeros campos corresponden a la fecha en la que tendrá lugar la ausencia y al interactuar con ellos podremos ver un calendario para elegir la fecha que deseemos. Cabe destacar que la fecha de inicio no puede ser mayor a la fecha de fin y si intentamos hacer esto, nos saldrá un PopOver indicándonos nuestro error.



El siguiente campo que nos encontramos es el del motivo de la ausencia, el cual se trata de un menú con los distintos motivos posibles de una ausencia, que pueden ser por descanso, vacaciones o incapacidad. Una vez hayamos rellenado correctamente todos los campos del formulario y pulsemos en el botón de guardar, se nos devolverá a la pantalla principal de la aplicación y nos saldrá un PopOver informándonos de que la ausencia ha sido creada con éxito, aunque deberá ser aprobada por el supervisor correspondiente al usuario que ha creado la ausencia.

Este proceso se realiza mediante una petición HTTP de tipo POST al servidor que recibe como cuerpo de la petición un objeto de tipo Ausencia, que tiene como propiedades el nombre del usuario que ha creado dicha ausencia, la empresa en la que trabaja ese usuario, la fecha de inicio y la fecha de fin de la ausencia, el motivo de la ausencia y si ha sido aprobada o no.






Lo siguiente que podemos ver en la pantalla de ausencias es un contenedor que tiene todas las ausencias que ha tenido el usuario en el último año, estén aprobadas o no, junto con su fecha y motivos correspondientes. Este contenedor se trata de un contenedor sobre el que podremos hacer scroll si hubiera suficientes ausencias como para que no cupieran dentro del contenedor.

Las ausencias son obtenidas mediante una petición HTTP al servidor de tipo GET que recibe como cuerpo de la petición el nombre del usuario y de su empresa, la fecha desde la que empezará a buscar y la fecha hasta la que buscará, que por defecto es el día de hoy. Como resultado de esta petición recibiremos un listado de objetos de tipo Ausencia con todas sus propiedades que cumplan con los requisitos indicados.

## **Pantalla de sesiones**

La pantalla de sesiones es la cual accedemos al pulsar el botón que tiene un icono de una hoja de texto en la pantalla principal y nos sirve para poder ver todas las sesiones de trabajo que ha tenido el usuario entre hoy y la fecha que este desee (por defecto la fecha se encuentra en seis meses hacia atrás desde el día de hoy).

 Sesiones 

Desde: 12/08/2021 

**Apr 22, 2022**  
Tiempo de jornada: 0 horas 0 minutos  
Horas de diferencia: 0 horas  
Minutos de diferencia: 0 minutos

**May 11, 2022**  
Tiempo de jornada: 5 horas 29 minutos  
Horas de diferencia: 5 horas  
Minutos de diferencia: 29 minutos

**May 17, 2022**  
Tiempo de jornada: 0 horas 1 minutos  
Horas de diferencia: 0 horas  
Minutos de diferencia: 1 minutos

En esta pantalla podemos observar la misma cabecera que en la pantalla de ausencias, que cuenta con el logo de la empresa del usuario en la parte izquierda (esta vez no es el pato de antes porque he cambiado de usuario a uno que tuviera más sesiones para poder ver mejor la pantalla), un título indicándonos en qué pantalla estamos en el centro y a la derecha un botón para volver a la pantalla principal de la aplicación.

Lo siguiente que vemos es un input que está relacionado con la fecha desde la que queremos ver nuestras sesiones (por defecto seis meses hacia atrás desde el día de hoy). Cuando pulsemos en él nos aparecerá un calendario para poder elegir fácilmente la fecha que deseemos.

Más abajo podemos ver distintas cards, cada una correspondiente a una sesión del usuario. Como título de la card tenemos la fecha de la sesión y en el cuerpo de la card podemos encontrarnos con tres textos. El primero de ellos se corresponde con la duración de la sesión en horas y minutos. Para explicar el segundo y tercer texto, antes debemos saber que cada usuario tiene asignado un tiempo de jornada que debe cumplir, este usuario al ser de prueba tiene asignada que su jornada debe durar 0 horas y 0 minutos. El segundo y tercer texto informan al usuario sobre el tiempo que ha durado su sesión de trabajo, ya sea de menos o de más, según el tiempo de jornada laboral que tenga asignado.

## **CONCLUSIONES**

La realización del proyecto descrito nos lleva a la obtención de una aplicación que permite al usuario guardar un registro de su jornada laboral, anotando ausencias, sesiones de trabajo, horas de la jornada realizada, y otros tipos de registros como pueden ser los descansos.

Hemos visto como en el desarrollo y metodología seguida, se utilizaban distintos lenguajes y procesos para alcanzar los objetivos deseados. Como recapitulación en el apartado previo a entrar en la aplicación, nos encontramos con el “Splash Screen” que para generarla hay que

cambiar ciertos archivos en la carpeta .android ( e .ios si también se requiere en dispositivos IOS). Seguidamente nos encontramos con la pantalla de *login* que cuenta con un formulario de dos campos, que una vez introducidos correctamente se avanzará al llamado “Home”, la pantalla principal de la aplicación y donde se realizan la mayoría de los procesos contados en las primeras secciones del documento. Esta parte cuenta con diversas funcionalidades antes descritas en su correspondiente parte.

Como conclusiones personales he aprendido el desarrollo y los ciclos de vida de un proyecto real, realizar un código más limpio y organizado además de seguro. Saber trabajar a partir de un código proporcionado por otra persona, aprender nuevas tecnologías como Angular, Ionic o .NET y por último aprender el desarrollo de una API y una mejor utilización de la misma.

## **BIBLIOGRAFÍA**

- Documentación de AngularJS
  - <https://angular.io/docs>
- Documentación de Ionic
  - <https://ionic.io/docs>
- Documentación de Capacitor
  - <https://capacitorjs.com/docs>
- Plugins oficiales de Capacitor (Usados para la geolocalización y almacenamiento temporal del dispositivo)
  - <https://capacitorjs.com/docs/apis>
- Plugins de Capacitor hechos por la comunidad (Usado el plugin http para poder realizar peticiones HTTP)
  - <https://github.com/capacitor-community>
  - <https://github.com/capacitor-community/http>

