

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



Cronos: control de asistencia laboral.
Aplicación web y dispositivo RFID
para el registro de fichajes.

TRABAJO FIN DE GRADO

Septiembre - 2023

AUTOR: Sandra Santiago Berna

DIRECTOR: Miguel Onofre Martínez Rach

AGRADECIMIENTOS

Me gustaría agradecer a mi familia por su apoyo a lo largo de los años. muy especialmente a mi hermana, por su constante motivación que me ha permitido seguir adelante.

Agradecer a mi pareja por su incondicional apoyo y comprensión sin el cual no habría llegado hasta aquí. También su aportación a este TFG al diseñar el modelo 3D y realizar la impresión de la caja para el prototipo.

Por último, pero no menos importante, quiero agradecer a mis profesores en la carrera por su dedicación todos estos años. Sus enseñanzas han sido la base de mi formación académica y me han permitido acceder al mundo laboral con éxito. Muy especialmente quiero agradecer a mi tutor Miguel Onofre. Su orientación y apoyo han sido imprescindibles para llevarlo a cabo.

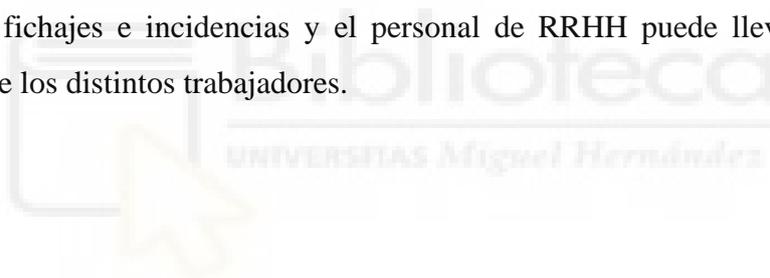


RESUMEN

Con la entrada en vigor del Real Decreto-ley 8/2019 las empresas se ven en la necesidad de llevar un control detallado sobre las jornadas realizadas por sus trabajadores. Con el fin de garantizar el cumplimiento de las normativas sobre limitación en las jornadas de trabajo, las empresas deberán ser capaces de registrar y llevar un control de los fichajes de cada trabajador. Además, llegado el caso, estos datos deben estar disponibles para su consulta por parte de una Inspección de Trabajo y Seguridad Social.

Este proyecto pretende ofrecer un producto totalmente funcional que cubra esa necesidad. Está orientado principalmente a Pymes, en especial a aquellas donde los empleados no disponen de un puesto con ordenador, como puede ser el caso de fábricas o tiendas.

Consta de un dispositivo físico desarrollado en Arduino. Emplea tecnología RFID y conecta con una API Rest para la recogida de fichajes. Por otra parte, incluye una web desarrollada en Laravel con perfiles diferenciados donde los trabajadores pueden consultar sus fichajes e incidencias y el personal de RRHH puede llevar un control detallado sobre los distintos trabajadores.



ÍNDICE

1	INTRODUCCIÓN	9
1.1	INTRODUCCIÓN E IMPORTANCIA	9
1.2	ELECCIÓN DEL PROYECTO	9
1.3	QUÉ SE PRETENDE CONSEGUIR	10
1.4	QUE NO SE PRETENDE CONSEGUIR	10
2	ESTADO DE LA CUESTIÓN	11
2.1	REAL DECRETO-LEY 8/2019	11
3	OBJETIVOS DEL PROYECTO	13
3.1	OBJETIVO PRINCIPAL	13
3.2	OBJETIVOS SECUNDARIOS	14
3.3	CONOCIMIENTOS PREVIOS	14
4	HIPÓTESIS DE TRABAJO	15
4.1	SOFTWARE	15
4.1.1	PROGRAMAS EMPLEADOS	15
4.1.1.1	XAMPP	15
4.1.1.2	SUBLIME	15
4.1.1.3	INSOMNIA	16
4.1.1.4	ARDUINO IDE	16
4.1.1.5	NAVEGADOR	17
4.1.2	LENGUAJES DE PROGRAMACIÓN	17
4.1.2.1	LARAVEL	17
4.1.2.2	PHP	18
4.1.2.3	HTML	18
4.1.2.4	JAVASCRIPT	18
4.1.2.5	CSS (TAILWIND)	19
4.1.2.6	SQL	19
4.1.2.7	ARDUINO	20
4.2	HARDWARE	21
4.2.1	PLACA ARDUINO	21
4.2.2	PLACA RFID	21
4.2.3	OTROS COMPONENTES	22
5	METODOLOGÍA	25
5.1	PLANIFICACIÓN	25

5.1.1	CICLO DE VIDA	25
5.2	CAPTURA DE REQUISITOS	27
5.2.1	ACTORES	27
5.2.1.1	USUARIO – TRABAJADOR	27
5.2.1.2	ADMINISTRADOR – RRHH	29
5.2.2	CASOS DE USO	31
5.3	API REST	48
5.4	DISPOSITIVO	50
5.4.1	CONEXIONES	52
5.4.2	LIBRERIAS	52
5.4.3	FUNCIONAMIENTO	53
5.5	BASE DE DATOS	57
5.5.1	ANÁLISIS	58
5.5.2	FUNCIONES	60
5.5.3	PROCEDIMIENTOS ALMACENADOS Y FUNCIONES	61
5.5.3.1	PROCESADO DE FICHAJES	61
5.5.3.2	GENERACIÓN DE INCIDENCIAS	66
5.6	INTERFAZ WEB	70
6	CONCLUSIONES Y PROPUESTAS	77
6.1	CONCLUSIONES	77
6.2	PROPUESTAS	77
7	BIBLIOGRAFÍA	79



ÍNDICE DE IMÁGENES

<i>Imagen 1 - Logo Xampp</i>	15
<i>Imagen 2 - Logo Sublime Text</i>	15
<i>Imagen 3 - Logo Insomnia</i>	16
<i>Imagen 4 - Logo Arduino IDE</i>	16
<i>Imagen 5 - Logos: Chrome, Edge, Firefox</i>	17
<i>Imagen 6 - Logo Laravel</i>	17
<i>Imagen 7 - Logo PHP</i>	18
<i>Imagen 8- Logo HTML</i>	18
<i>Imagen 9 - Logo JavaScript</i>	18
<i>Imagen 10 - Logo Tailwind Css</i>	19
<i>Imagen 11 - Logo SQL</i>	19
<i>Imagen 12 - Logo Arduino</i>	20
<i>Imagen 13 - Ejemplo estructura código Arduino</i>	20
<i>Imagen 14 - Placa Arduino MKR100</i>	21
<i>Imagen 15 - Tarjeta RFID RC522</i>	21
<i>Imagen 16 - Leds: Blanco, rojo y verde</i>	22
<i>Imagen 17 -Zumbador</i>	22
<i>Imagen 18 - Resistencias 220 Ohm</i>	22
<i>Imagen 19 - Transformador de voltaje</i>	23
<i>Imagen 20 - Conexión hembra tipo C</i>	23
<i>Imagen 21 - Placa de soldado</i>	24
<i>Imagen 22 - Cables</i>	24
<i>Imagen 23 - Diagrama ciclo de vida incremental</i>	26
<i>Imagen 24- Actores</i>	27
<i>Imagen 25 - Diagrama casos de uso del usuario - trabajador</i>	27
<i>Imagen 26 - Diagrama casos de uso del administrador -RRHH</i>	29
<i>Imagen 27 - Insomnia: parámetros POST para login</i>	48
<i>Imagen 28 - Insomnia: cabeceras POST para login</i>	48
<i>Imagen 29 - Insomnia: Respuesta login correcto</i>	49
<i>Imagen 30 - Insomnia: Respuesta login erróneo</i>	49
<i>Imagen 31 - Insomnia: parámetros POST para fichaje</i>	49
<i>Imagen 32 - Insomnia: cabeceras PORT para fichaje</i>	50
<i>Imagen 33- Insomnia: Respuesta al fichaje</i>	50
<i>Imagen 34 - Prototipo 1º versión</i>	50
<i>Imagen 35 - Prototipo final (exterior)</i>	51
<i>Imagen 36 - Prototipo final (interior)</i>	51
<i>Imagen 37 - Esquema de conexiones del prototipo</i>	52
<i>Imagen 38- Diagrama de flujo Setup() arduino</i>	55
<i>Imagen 39 - Diagrama de flujoproceso fichaje</i>	57
<i>Imagen 40 - Diagrama de clases</i>	58
<i>Imagen 41 - Ejemplo tipos de jornadas</i>	61
<i>Imagen 42 - Ejemplo fichajes duplicados (entrada)</i>	62
<i>Imagen 43 - Ejemplo fichajes duplicados (salida)</i>	62
<i>Imagen 44 - Ejemplo cálculo de jornada (correcto)</i>	63
<i>Imagen 45 - Ejemplo cálculo de jornada (incorrecto)</i>	63
<i>Imagen 46 - Ejemplo impuntualidad</i>	67
<i>Imagen 47 - Tipos de incidencias</i>	68
<i>Imagen 48 - Estructura web</i>	71

<i>Imagen 49 - Pantalla de login.....</i>	<i>71</i>
<i>Imagen 50 – Pantalla consulta de fichajes propios</i>	<i>72</i>
<i>Imagen 51 - Pantalla consulta incidencias propias</i>	<i>73</i>
<i>Imagen 52 - Pantalla resumen administrador</i>	<i>73</i>
<i>Imagen 53 - Pantalla de consulta de usuarios.....</i>	<i>74</i>
<i>Imagen 54- Pantalla alta nuevo usuario</i>	<i>74</i>
<i>Imagen 55 - Pantalla consulta tipos de horarios.....</i>	<i>75</i>
<i>Imagen 56 - Pantalla consulta el detalle de un horario.....</i>	<i>75</i>
<i>Imagen 57 - Pantalla consulta incidencias</i>	<i>76</i>



ÍNDICE DE TABLAS

<i>Tabla 1 - Casos de uso del usuario - trabajador</i>	28
<i>Tabla 2 - Casos de uso del administrador – RRHH</i>	30
<i>Tabla 3 - C.U. 1: Inicio de sesión</i>	31
<i>Tabla 4 - C.U. 2: Consultar resumen</i>	32
<i>Tabla 5 - C.U. 3: Consultar datos personales</i>	32
<i>Tabla 6 - C.U. 4: Editar datos personales</i>	33
<i>Tabla 7 - C.U.5: Consultar fichajes propios</i>	33
<i>Tabla 8 - C.U. 6: Consultar incidencias propias</i>	34
<i>Tabla 9 - C.U.7 - Consultar horario asignado</i>	34
<i>Tabla 10 - C.U. 8: Consultar festivos</i>	35
<i>Tabla 11 - C.U. 9: Cerrar sesión</i>	35
<i>Tabla 12 - C.U. 10: Fichar</i>	36
<i>Tabla 13 - C.U. 11: Consultar resumen administrador</i>	37
<i>Tabla 14 - C.U. 12 - Consultar listado de usuarios</i>	37
<i>Tabla 15 - C.U. 13: Alta de un nuevo usuario</i>	38
<i>Tabla 16 - C.U. 14: Editar usuario</i>	39
<i>Tabla 17 - C.U. 15: Consultar tarjetas</i>	39
<i>Tabla 18 - C.U. 16: Alta de nueva tarjeta</i>	40
<i>Tabla 19 - C.U. 17: Editar tarjeta</i>	41
<i>Tabla 20 - C.U. 18: Consultar dispositivos</i>	41
<i>Tabla 21 - C.U. 19: Alta de nuevo usuario</i>	42
<i>Tabla 22 - C.U. 20: Editar dispositivo</i>	42
<i>Tabla 23 - C.U. 21: Consultar horarios</i>	43
<i>Tabla 24 - C.U. 22: Alta de un horario</i>	44
<i>Tabla 25 - C.U. 23: Edición de un horario</i>	45
<i>Tabla 26 - C.U. 24 - Consultar festivos administración</i>	45
<i>Tabla 27 - C.U. 25: Alta de nuevo festivo</i>	46
<i>Tabla 28 - C.U. 26: Editar festivo</i>	46
<i>Tabla 29 - C.U. 27 - Consultar incidencias</i>	47
<i>Tabla 30 - C.U. 28: Resolución de incidencias</i>	48

1 INTRODUCCIÓN

1.1 INTRODUCCIÓN E IMPORTANCIA

Actualmente la sociedad es cada vez más consciente de la importancia que tiene para la salud, tanto física como mental, el disponer de unas condiciones de trabajo dignas y que prioricen el bienestar del trabajador y la conciliación personal y familiar.

Conceptos como ‘el salario emocional’ toman importancia, pero las empresas no siempre están abiertas a un cambio cultural tan grande. Por esta razón las leyes deben garantizar que se cumplen unas condiciones dignas de bienestar mediante legislaciones nuevas como es el caso del Decreto-Ley 8/2019.

El producto presentado en este TFG pretende ser una herramienta más en la lucha contra la precariedad laboral y las condiciones abusivas, en concreto solo la regulación de las jornadas laborales. Procedimientos como el registro de la jornada son un primer paso hacia ese cambio social y cultural.



1.2 ELECCIÓN DEL PROYECTO

En el momento de implantación del Real Decreto-ley 8/2019 yo me encontraba trabajando en la empresa y fui la encargada de implementar un sistema para el control de los fichajes. Si bien la necesidad a cubrir en esa ocasión es la misma que se pretende suplir con este proyecto, el enfoque sobre cómo abordar el problema es completamente distinto.

En esa ocasión para el hardware se optó por un producto comercial bastante costoso que permitía la recogida de los fichajes. En cuanto a la parte de procesamiento de los datos, mi empresa cuenta con un ERP totalmente personalizado y programado en C++, que sigue en desarrollo continuo por parte del equipo informático interno. Mi trabajo consistió en ampliar el módulo correspondiente a ‘laboral’ del ERP para incluir las funcionalidades necesarias para el procesado de los fichajes.

Trabajar en ese proyecto me llevo a pensar en cómo abordarían la problemática de implantar la nueva ley aquellas empresas más pequeñas que no disponen de tanto presupuesto para invertir en dispositivos o que no cuentan con un departamento de

informática propio que haga desarrollos personalizado. Ese pensamiento me inspiro para la creación de este producto.

1.3 QUÉ SE PRETENDE CONSEGUIR

El objetivo principal de este proyecto es poner en práctica los conocimientos adquiridos en el grado para desarrollar un producto totalmente funcional que permita cubrir la necesidad de las empresas de llevar un registro y control de las jornadas laborales acorde a lo establecido en el Decreto-Ley8/2019.

Para ello se han empleado los conocimientos adquiridos específicamente en el grado como puede ser la programación en SQL y HTML. Además de aprender sobre nuevas tecnologías como Arduino y Laravel que pueden ser de utilidad para el futuro.

El proyecto consta de 3 partes diferenciadas, el dispositivo hardware para la recogida de datos (en este caso fichajes), una API Rest para la comunicación con el dispositivo y una web con distintos perfiles para la consulta y gestión de los datos.

1.4 QUE NO SE PRETENDE CONSEGUIR

Por cuestión de tiempo no se ha podido desarrollar todas las funcionalidades pensadas inicialmente, se ha priorizado el presentar un producto simplificado pero funcional que incluya tanto el desarrollo hardware (el prototipo), como el software (la API y la web).

Algunas de esas funcionalidades de la parte web que no se han podido desarrollar son: permitir horarios con turnos rotativos, la subida de documentos (como justificantes de ausencia), resúmenes estadísticos de puntualidad o incidencias. En cuando al dispositivo, no se ha podido prepara una web implementada en el propio Arduino para facilitar el cambio de red wifi o incluir una pantalla.

Estas funcionalidades están detalladas más adelante en el apartado sobre mejoras futuras '6.2 – Propuestas'.

No se pretende crear una aplicación multiplataforma, aunque al tratarse de una web con diseño responsive si se podrá visualizar en distintos dispositivos.

Para finalizar debido a los problemas detallados en el apartado 4.1.1.1 queda fuera de los objetivos de este proyecto la subida de la web al servidor definitivo.

2 ESTADO DE LA CUESTIÓN

2.1 REAL DECRETO-LEY 8/2019

En agosto del 2019 se publicó en el BOE el Real Decreto-Ley 8/2019, con él se pretende garantizar que los trabajadores tengan jornadas de trabajo dignas y que permitan la conciliación con la vida personal. Según datos citados en el Decreto-Ley, en el 2018 las encuestas ponen de manifiesto que más del 50% de los trabajadores a tiempo completo realizaba jornadas superiores a las 40 horas semanales. En el caso de los trabajadores a tiempo parcial los datos indican que las horas trabajadas superan ampliamente las horas declaradas. En cuanto a las horas extraordinarias un 48% de las personas entrevistadas declara que no cobra por las horas extraordinarias realizadas, esto también supone que no se cotiza por ellas a la Seguridad Social.

«la realización de horas extraordinarias ilegales por superar la cifra máxima permitida de 80 al año, la realización de horas extraordinarias que no son abonadas ni compensadas con descanso, sean o no legales, y las situaciones relacionadas con la organización del trabajo y el establecimiento de altos ritmos, para actuar tanto sobre los aspectos puramente laborales como los relacionados con la incidencia que tales factores tienen en la prevención de riesgos laborales».

Real Decreto-Ley 8/2019 – Capítulo V

Hasta el momento tanto inspectores de trabajo como los propios trabajadores tenían pocas herramientas para demostrar si sufrían condiciones abusivas en lo referente a la duración de la jornada. Durante una inspección, al no disponer de un registro detallado de las horas trabajadas, resulta muy difícil demostrar si la prolongación de la jornada laboral se trata de un caso continuado o de un pico de trabajo puntual por el que era necesario realizar horas extra.

“La realización de un tiempo de trabajo superior a la jornada laboral legal o convencionalmente establecida incide de manera sustancial en la precarización del mercado de trabajo, al afectar a dos elementos esenciales de la relación laboral, el tiempo de trabajo, con relevante influencia en la vida personal de la persona trabajadora al dificultar la conciliación familiar, y el salario. Y también incide en las cotizaciones de Seguridad Social, mermadas al no cotizarse por el salario que correspondería a la jornada realizada.”

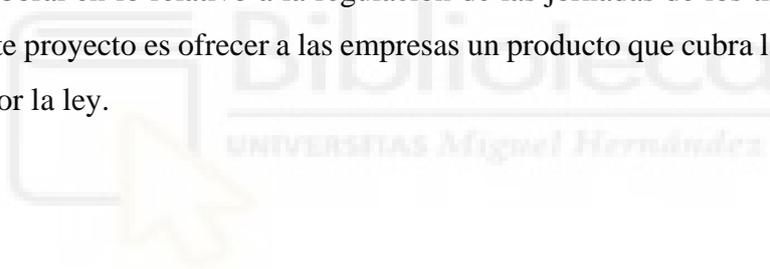
Real Decreto-Ley 8/2019 – Capítulo V

A sí mismo, con la implantación de este Decreto-Ley, la legislación se actualiza para cumplir lo establecido por la Directiva 2003/88/CE del Parlamento Europeo y del Consejo, de 4 de noviembre de 2003 que especifica que las empresas deben disponer de un sistema para registrar el cómputo total de la jornada laboral efectiva de sus trabajadores.

«La normativa europea impone a las empresas la obligación de implantar un sistema de cómputo de la jornada laboral efectiva de los trabajadores a tiempo completo que no se hayan comprometido de forma expresa, individual o colectivamente, a realizar horas extraordinarias y que no ostenten la condición de trabajadores móviles, de la marina mercante o ferroviarios, y se oponen a una normativa nacional de la que no resulta la existencia de esa obligación».

Real Decreto-Ley 8/2019 – Capítulo V

En resumen, ese Decreto-Ley pretende ayudar a corregir y mejorar la situación de precariedad laboral en lo relativo a la regulación de las jornadas de los trabajadores. El objetivo de este proyecto es ofrecer a las empresas un producto que cubra las necesidades establecidas por la ley.



3 OBJETIVOS DEL PROYECTO

3.1 OBJETIVO PRINCIPAL

El objetivo principal del proyecto es la creación de un producto asequible y útil que cubra la necesidad de las empresas de controlar la asistencia de sus trabajadores.

Consta de un dispositivo desarrollado en Arduino equipado con una placa RFID. El dispositivo es de fácil instalación, solo necesita un enchufe y acceso a una red wifi para funcionar. La empresa podrá instalar tantos dispositivos como deseé en los distintos puntos de acceso al recinto. Cada empleado dispondrá de una tarjeta personal para fichar en cualquiera de los dispositivos disponibles.

Para la consulta y el procesado de los datos se emplea las siguientes tecnologías:

- Una API Rest que permite la comunicación del dispositivo con la base de datos. Cuando se realiza un fichaje se envía una petición POST que la API procesa y guarda un registro en la base de datos.
- Una web para los trabajadores que cuenta con 2 perfiles diferenciados, uno para trabajadores (usuario) y otro para personal de RRHH (administrador). Las funcionalidades de cada perfil son las siguiente:
 - o Los trabajadores podrán acceder para consultar los fichajes, horarios e incidencias, así como sus datos personales.
 - o El personal de RRHH, además de las funcionalidades de los trabajadores, también dispondrán de habilidades propias de su rol. Podrán consultar, modificar y crear nuevos usuarios, dispositivos, horarios y tarjetas. Por otro lado, podrán de resolver incidencias y asignar horarios entre otras habilidades.
- Por último, el sistema posee una serie de tareas programadas que ejecutan procedimientos almacenados para procesar los fichajes. Estas tareas se encargan de agrupar los fichajes por jornadas y compáralos con el horario asignado al trabajador para identificar discrepancias, de ser así, el sistema generará incidencias que posteriormente estarán disponibles para ser supervisadas por el departamento de RRHH.

3.2 OBJETIVOS SECUNDARIOS

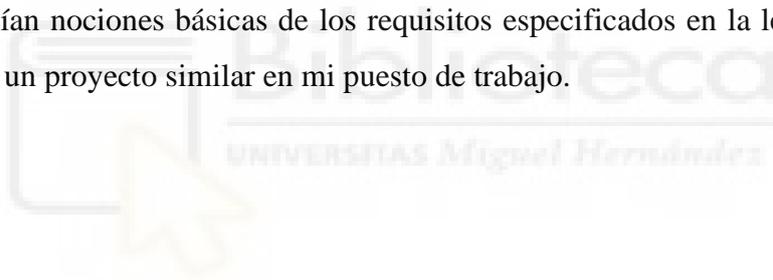
Con este proyecto se pretende afianzar y ampliar los conocimientos adquiridos en la carrera. Así como aprender sobre nuevos lenguajes de programación web como es el caso de Laravel o Tailwind y profundizar en los conocimientos sobre SQL.

Además, tenía interés en el desarrollo de un dispositivo hardware desde cero y aprender sobre su programación. Ha sido muy interesante el enfrentarme al reto que supone pensar en todos los aspectos necesarios para su creación: la búsqueda de componentes, diseñar y montar el circuito y por supuesto aprender un nuevo lenguaje como es Arduino.

3.3 CONOCIMIENTOS PREVIOS

Algunas de las competencias adquiridas en la carrera han sido especialmente útiles para el desarrollo de este proyecto. Tener unas bases sólidas sobre la programación web: html, php, css (aunque no tailwind), uso de xampp, sql y conocimientos sobre la gestión de proyecto ha facilitado poder llevarlo a término.

También poseían nociones básicas de los requisitos especificados en la ley debido a la realización de un proyecto similar en mi puesto de trabajo.



4 HIPÓTESIS DE TRABAJO

4.1 SOFTWARE

4.1.1 PROGRAMAS EMPLEADOS

A continuación, se hace un repaso por los programas empleados para la realización del proyecto, explicando brevemente las funcionalidades principales de cada uno y el motivo de su elección.

4.1.1.1 XAMPP

XAMPP es un paquete de software libre que actúa como un servidor local. Incluye un servidor web Apache, una base de datos MariaDB, un servidor de correo Mercury y un servidor FTP FileZilla entre otras. Posee una interfaz sencilla e intuitiva que permite habilitar solamente aquellos servicios que se vayan a necesitar, en este caso solo ha sido necesario el servidor Apache y la base de datos MariaDB.



Imagen 1 - Logo Xampp

Se ha seleccionado esta opción como servidor porque tiene la ventaja de poder usarse sin necesidad de conexión a internet. Además, al estar instalado en local, toda la administración se puede llevar a cabo sin asistencia de asistencia por parte de terceros. La versión utilizada ha sido la v3.3.0.

En un primer momento se valoró la opción de implementar una máquina en los servidores de la UMH, pero tras algunos retrasos por dificultades técnicas y tras haber iniciado el desarrollo en XAMPP se decidió mantenerlo, dejando como ampliación futura el subirlo a la nube.

4.1.1.2 SUBLIME

Se trata de un editor de texto y código fuente que destaca por la simplicidad de su interfaz. Pese a ello, dispone de muchas funcionalidades útiles para ayudar en la programación como pueden ser el minimapa y la organización en pestañas. Soporta un gran número de lenguajes, tiene función de marcado de llave, paréntesis y autocompletado para el código



Imagen 2 - Logo Sublime Text

lo cual agilizar la escritura. Es capaz de revisar la sintaxis en tiempo real y la resalta en función del lenguaje escogido.

Una de las grandes ventajas de este editor es la cantidad de plugins de los que dispone, para ayudarme en la realización del proyecto he utilizado los siguientes:

- **SublimeCodeIntel:** Habilita la función de autocompletado.
- **SideBarEnhancements:** Aumenta las opciones disponibles a la barra lateral de archivo incluyendo funcionalidades como mover archivos entre carpetas.
- **SublimeLinter:** Resaltador de sintaxis.
- **SublimeLinter PHP:** Resaltador de sintaxis específico para PHP
- **PHP Companion:** Permite configurar atajos de teclado personalizados para optimizar tareas como por ejemplo añadir las cabeceras o explorar la definición de una clase entre otras.

La versión de sublime empleada en el desarrollo es la 3.

4.1.1.3 INSOMNIA

Insomnia permite testear APIs para Frontend y Backend, simula peticiones GET, POST, PUT, PATCH y DELETE. Además, permite validar las respuestas recibidas mediante mensajes y códigos de error.



Imagen 3 - Logo Insomnia

La versión empleada ha sido la 2023.5.7 y ha sido especialmente útil para validar el funcionamiento de la parte API que se emplea para la conexión con el dispositivo Arduino.

4.1.1.4 ARDUINO IDE

Arduino IDE es una aplicación de código abierto desarrollada específicamente para el desarrollo de programas compatibles con las placas Arduino. Incluye un paquete de ejemplos muy útiles para iniciarse en el lenguaje y dispone de un amplio catálogo de librerías.



Imagen 4 - Logo Arduino IDE

Su interfaz se divide en 3 partes principales, la parte superior donde encontramos los menús y las opciones para interactuar con el código como: compilar, cargar el sketch

(código fuente) en la placa, etc. La parte central está destinada al editor de código. Por último, la parte inferior cuenta con un área de mensajes donde el compilador informa de estado de la ejecución y de posibles errores en el código.

4.1.1.5 NAVEGADOR

La web desarrollada se ha probado en distintos navegadores web para comprobar su funcionamiento. Salvo pequeñas variaciones estéticas que resultan irrelevantes, la web funciona correctamente en Chrome, Edge y Firefox.



Imagen 5 - Logos: Chrome, Edge, Firefox

4.1.2 LENGUAJES DE PROGRAMACIÓN

En este apartado se exponen los distintos lenguajes de programación necesarios para el desarrollo de este proyecto.

4.1.2.1 LARAVEL

Laravel es un framework de PHP que emplea la tecnología MVC (Modelo-Vista-Controlador) para el desarrollo de páginas web. Esta arquitectura garantiza una estructura de código muy clara y estructurada.



Imagen 6 - Logo Laravel

Otra de las características que hacen de Laravel una buena elección para el desarrollo son:

- El uso de Eloquent, el ORM (Object-Relational Mapping) para la conversión de objetos y comunicación con la base de datos facilita en gran medida el manipulado de los mismo. Elimina por completo el uso de sentencias SQL explícitas en el código, quedando este mucho más legible.
- Para el control de las plantillas se ha empleado el motor de Blade, este motor permite modularizar las vistas (layouts) para evitar la redundancia de código. Otra característica que resultan muy útil es el permitir el uso de PHP plano en las vistas y la herencia entre plantillas.

- El control de autenticación se ha implementado usando Laravel Breeze, este paquete proporciona una estructura básica con la que implementar el inicio de sesión y el control de sesiones.

4.1.2.2 PHP

PHP, acrónimo recursivo de Hypertext Preprocessor, es un lenguaje de código abierto destinado a la creación de aplicaciones y páginas web. Su principal ventaja es que se ejecuta en el propio servidor web, esto proporciona mayor agilidad y velocidad para la conexión con las bases de datos.



Imagen 7 - Logo PHP

Es compatible con cualquier tipo de servidor y SO y puede combinarse con otros lenguajes de programación como HTML, CSS o Java.

Para este proyecto se ha empleado la versión 8.1.2.

4.1.2.3 HTML

HTML (HyperText Markup Language) es uno de los 3 pilares fundamentales para cualquier página web, junto con el CSS y el JS. Se trata de un lenguaje de marcado que define la estructura que tendrá la web. Está compuesto por una serie de etiquetas que permiten definir el contenido de la web como los títulos, párrafos, imágenes, enlaces, listas y demás. La estructura de las plantillas de Blade se componen de código HTML 5.



Imagen 8 - Logo HTML

4.1.2.4 JAVASCRIPT

JavaScript es un lenguaje de programación que aporta dinamismo a la web ya que permite que está reaccione ante la interacción del usuario, lo cual mejora la experiencia de navegación.



Imagen 9 - Logo JavaScript

Se emplea para control multimedia, incluir animaciones (como en el caso de algunos tipos de menús), actualizaciones dinámicas del contenido y animación de imágenes entre otras funcionalidades.

Funciona tanto en el lado del cliente como en el del servidor, si se emplea en la parte del cliente aporta el dinamismo y las animaciones comentadas anteriormente, mientras que empleado en la parte del servidor es capaz de generar dinámicamente nuevo contenido empleando la lógica de la aplicación e incluso modificando la base de datos.

Para este proyecto se ha usado para la parte del cliente y la versión empleada es la 10.13.5

4.1.2.5 CSS (TAILWIND)

El CSS (Cascading Style Sheets) es un lenguaje imprescindible en la creación de páginas web, mediante él se crean reglas que definen la apariencia que tendrá nuestra web.

Actualmente existen muchas herramientas, frameworks y librerías que facilitan la creación de diseños personalizados.

Para este proyecto se ha seleccionado Tailwind, es el recomendado por la documentación oficial de Laravel y dispone de una comunidad muy amplia donde encontrar ejemplos y modelos.

Tailwind es un framework de CSS basado en la arquitectura Atomic CSS, esta tipo de arquitectura favorece el uso de clases implementadas para un solo propósito y con nombres basados en su función. Tailwind provee de una librería con un amplio catálogo de clases que se pueden combinar entre si para conseguir un diseño altamente personalizado.



Tailwind CSS

Imagen 10 - Logo Tailwind Css

4.1.2.6 SQL

SQL (Structured Query Language), es un lenguaje de consulta estructurado que permite administrar y recuperar datos almacenados en un sistema de gestión de bases de datos relacionales. Para este proyecto, al usar el ORM de Eloquent,



Imagen 11 - Logo SQL

no ha sido necesario redactar las consultas más básicas para la obtención de datos, sin embargo, si se ha empleado este lenguaje para desarrollar procedimientos almacenados. Esto procedimiento se ejecuta de forma programada por el servidor y permiten manipular los datos: agrupan los fichajes en jornadas y los procesan para generar incidencias entre otras funciones.

4.1.2.7 ARDUINO

El lenguaje Arduino es una versión reducida de C/C++, pero incluye muchas de sus funcionalidades como el uso de librerías, funciones, tipos de datos, variable. La característica principal es la capacidad de interactuar con los puertos de entrada y salida de la placa, existen multitud de librerías específicas para estas funciones que permiten controlar tanto las entradas propias de la placa, como los periféricos conectados a esta ya sean leds, altavoces o sensores de cualquier tipo.



Imagen 12 - Logo Arduino

Todos los programas creados para Arduino tienen una estructura similar a la indicada en la imagen 13, es imperativo respetarla para el correcto funcionamiento de este tipo de placas. En primer lugar, hay una sección destinada a la declaración de las cabeceras de las librerías que se va a emplear, así como de variables globales. En la siguiente sección 'Setup' se incluirán las acciones que la placa debe realizar nada más arrancar. La parte principal del código es la función loop, todo el código aquí incluido se ejecuta en bucle continuamente mientras la placa esté encendida, esta es una particularidad de este tipo de placas que se debe tener muy presente a la hora de diseñar el código. Para finalizar, a continuación de esta estructura se pueden declarar tantas funciones adicionales como se necesiten.

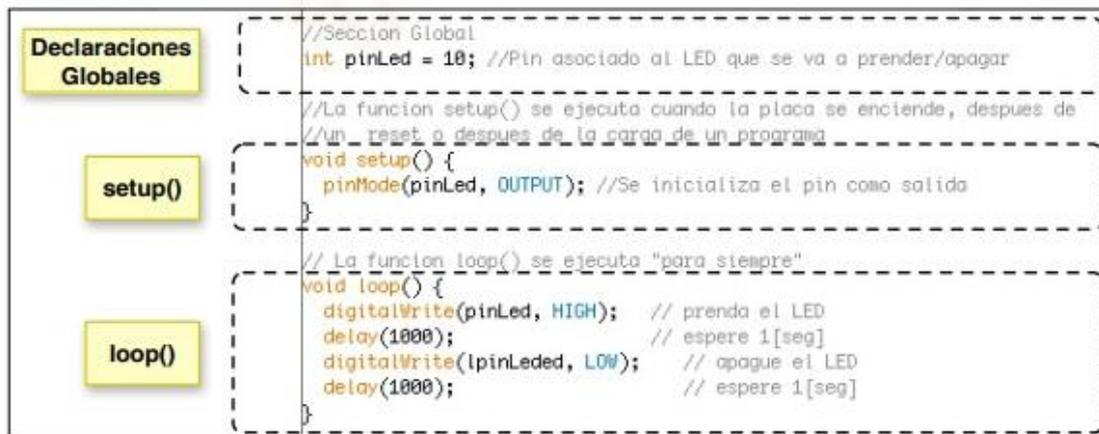


Imagen 13 - Ejemplo estructura código Arduino

4.2 HARDWARE

Dado que el proyecto incluye el desarrollo de un dispositivo y la fabricación de un prototipo, se ha incluido este apartado para explicar las características del hardware empleado para su fabricación.

4.2.1 PLACA ARDUINO

Arduino dispone de una gran variedad de placas con distintas funcionalidades, además de infinidad de módulos que se pueden combinar entre sí. La placa seleccionada es el **modelo MKR1000**, está especialmente diseñada para proyectos del internet de las cosas (IoT). Dispone de un microcontrolador ARM SAMD21 de 32-bits con 256Kb de memoria FLASH y 32Kb de SRAM. Tiene integrado un potente chip Wifi WINC1500 de bajo consumo con el stack TCP/IP, así como un chip de encriptación por hardware ECC508 que permite realizar comunicaciones seguras.



Imagen 14 - Placa Arduino MKR1000

Destaca por su bajo consumo y su tamaño reducido, esas características junto con el hecho de contar un el chip Wifi integrado y de tener un precio muy asequible (en torno a 35€) la convierte en una elección perfecta para cubrir las necesidades del proyecto.

4.2.2 PLACA RFID

Se ha utilizado una placa modelo RFID – RC522 que emplea 3.3V de alimentación. Permite conexiones SPI y UART para comunicarse con la placa principal e incluye un llavero y una tarjeta regrabable para realizar pruebas. El precio de este componente es muy asequible alrededor de 1 o 2€ aproximadamente, es muy buena elección para este proyecto en el que se intentan minimizar los gastos.



Imagen 15 - Tarjeta RFID RC522

Para más detalles técnicos sobre este componente se puede consultar la hoja técnica incluida en la bibliografía.

4.2.3 OTROS COMPONENTES

A continuación, se incluye una lista de otros componentes electrónicos empleados para el desarrollo.

- 1) **Leds:** Se han incluido leds de 3 colores (Blanco, verde, rojo) para indicar al usuario el estado del dispositivo y ofrecer un feedback de las interacciones y el estado del mismo.



Imagen 16 - Leds: Blanco, rojo y verde

- 2) **Zumbador:** Al igual que los leds el zumbador proporciona un feedback para indicar si el proceso de fichaje ha tenido éxito.



Imagen 17 -Zumbador

- 3) **Resistencias:** Se han añadido resistencias de 220 Ohms para regular la alimentación de los leds.



Imagen 18 - Resistencias 220 Ohm

- 4) **Transformador de voltaje:** Ha sido necesario añadir 1 transformador para solucionar un problema de voltaje que ha surgido al alimentar la placa RFID. La placa MKR1000 dispone únicamente de salidas de 5v, siendo necesaria una de 3,3v para el RFID. Mediante un divisor de voltaje con resistencias se obtuvo los 3,3v deseados, sin embargo las salidas de la placa tienen una salida de amperaje muy limitada, insuficiente para alimentar la placa. Por esta razón se optó por incorporar un transformador regulable y conectar cada placa directamente a la alimentación que necesita.

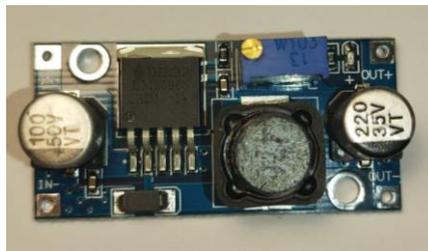


Imagen 19 - Transformador de voltaje

- 5) **Conexión hembra tipo C:** Al tener que añadir al diseño el transformador de voltaje es necesario contar con una conexión adicional donde conectar el cargador.



Imagen 20 - Conexión hembra tipo C

- 6) **Protoboard / placa de soldado:** Para el desarrollo del prototipo se empleó una protoboard para implementar las conexiones. Una vez definidos el circuito se ha replicado en una placa definitiva con las conexiones soldadas para mayor seguridad.

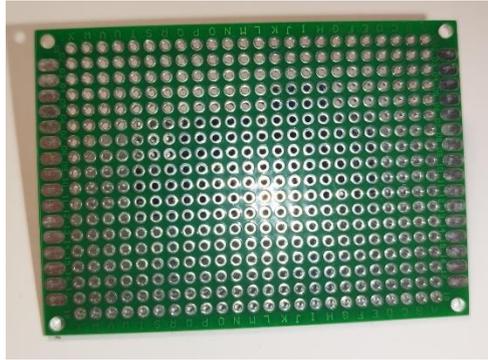


Imagen 21 - Placa de soldado

7) **Cables:** Necesarios para conectar los distintos componentes entre sí.

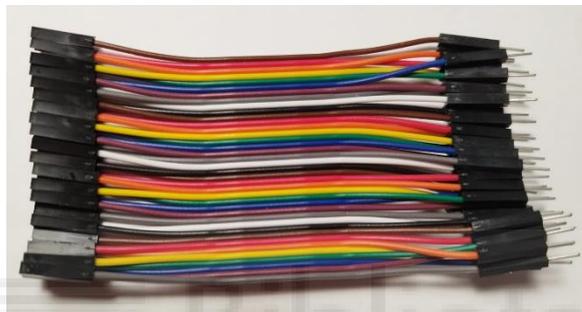


Imagen 22 - Cables

La mayoría de los componentes (placa Arduino, cables, zumbador, resistencias, leds) han sido proporcionados por la universidad como parte del kit (MKR IoT BUNDLE) para el desarrollo del proyecto. La placa RFID, el transformador, la placa de soldado y la conexión tipo C son los únicos componentes empleados que no estaban incluidos en el kit.

5 METODOLOGÍA

5.1 PLANIFICACIÓN

5.1.1 CICLO DE VIDA

Una vez analizado el proyecto en conjunto y dada la complejidad y la variedad de tecnologías empleadas se ha decidido separar el desarrollo en 3 etapas diferenciadas. Cada una de estas etapas cuenta con su propio ciclo de desarrollo donde se siguen los pasos a continuación descritos.

- **Programación de la API Rest:** Llevada a cabo en Laravel, gestiona la futura interacción con el dispositivo. Supuso la primera toma de contacto con el nuevo lenguaje y permitió familiarizarse con el uso de rutas y controladores. Las pruebas de funcionamiento de esta etapa se llevaron a cabo con el programa Insomnia simulando las llamadas realizadas por el dispositivo, tanto para el login como para el fichaje.
- **Dispositivo en Arduino:** se optó por darle prioridad frente al desarrollo web por ser la tecnología menos conocida y la más compleja. Esta etapa incluye el diseño del dispositivo, la implementación del hardware y el desarrollo software que conecta con la API. Depende por completo de la etapa 1 ya que no es posible probar la conexión sin tener implementada la API.
- **La web en laravel:** esta etapa relativamente independiente de las demás, aunque comparten el uso de alguno de los modelos de datos o clases. La mayoría de los datos: usuarios web, horarios, dispositivos, tarjetas... no dependen del hardware, solo comparten los usuarios y la parte del fichaje. Los datos sobre fichajes se han introducido mediante el uso de seeders.

La metodología de trabajo empleada es el ciclo de vida incremental o iterativo, esta metodología está basada en la repetición de las distintas fases que componen un proyecto, explicadas a continuación. En cada una de las iteraciones se analiza, implementa y presenta una parte del producto final, este planteamiento resulta especialmente útil para un proyecto como este que tienen 3 etapas muy diferenciadas como se ha visto anteriormente.

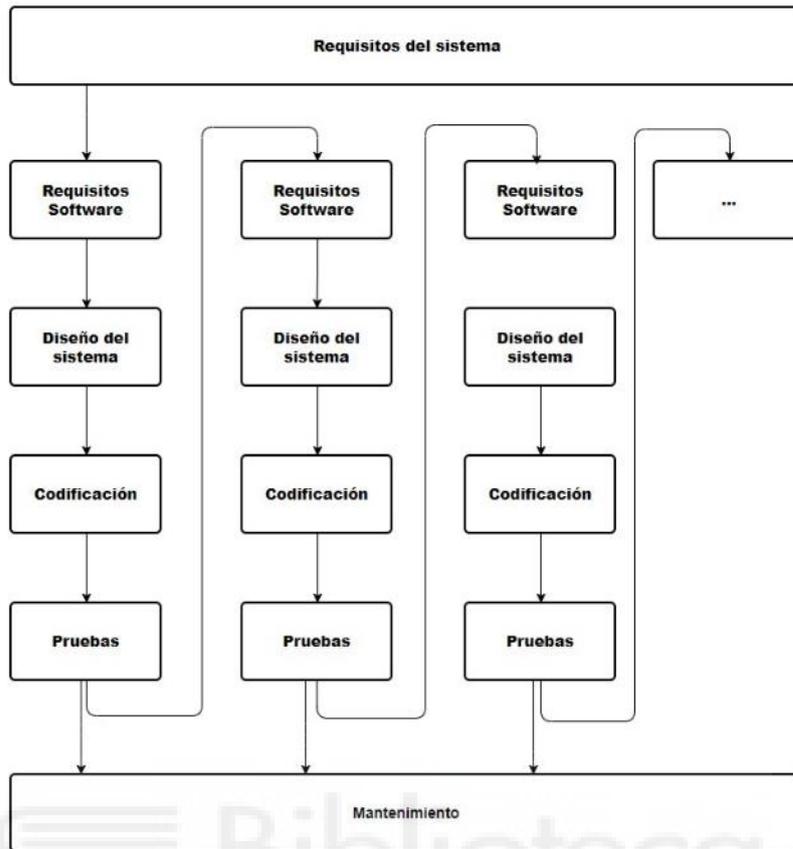


Imagen 23 - Diagrama ciclo de vida incremental

Este modelo se fundamenta en 4 fases:

- 1) **Análisis de requisitos:** en esta fase se definen todas las características que debe tener el producto final, es importante tener en cuenta todos los objetivos a cubrir y detallar todas las funcionalidades.
- 2) **Diseño del sistema:** se descomponen las distintas partes que forman el sistema y se detallan las especificaciones técnicas, también se detalla cómo se van a relacionar entre si las distintas partes. Incluye el análisis de las herramientas necesarias para la codificación.
- 3) **Codificación:** creación de algoritmos e implementación del código fuente.
En el caso del hardware también se incluye la fabricación del prototipo.
- 4) **Pruebas:** Comprobación del funcionamiento para garantizar que el producto cumple con las especificaciones indicadas en la parte de análisis de requisitos.

5.2 CAPTURA DE REQUISITOS

5.2.1 ACTORES

En este proyecto se han detectado 2 tipos de actores diferenciados principalmente por el rol que desempeñan en la empresa. Se dispone de un perfil básico que permite consultas y de otro perfil de administrador que será una extensión del anterior con habilidades propias como la gestión de los datos.

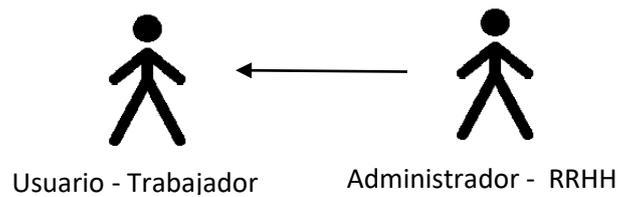


Imagen 24- Actores

5.2.1.1 USUARIO – TRABAJADOR

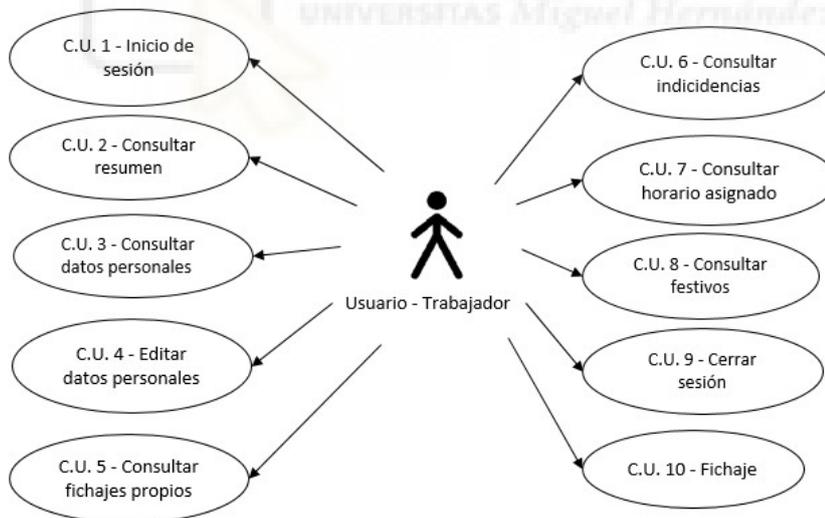


Imagen 25 - Diagrama casos de uso del usuario - trabajador

Usuario	Trabajador
Descripción	Este rol engloba a todos los trabajadores. Básicamente está limitado a la consulta de los datos propios del usuario.
Casos de uso	<p>C.U. 1 – Inicio de sesión</p> <p>C.U. 2 – Consultar resumen</p> <p>C.U. 3 – Consultar datos personales</p> <p>C.U. 4 – Editar datos personales</p> <p>C.U. 5 – Consultar fichajes propios</p> <p>C.U. 6 – Consultar incidencias propias</p> <p>C.U. 7 – Consultar horario asignado</p> <p>C.U. 8 – Consultar festivos</p> <p>C.U. 9 – Cerrar sesión</p> <p>C.U. 10 - Fichaje</p>

Tabla 1 - Casos de uso del usuario - trabajador



5.2.1.2 ADMINISTRADOR – RRHH

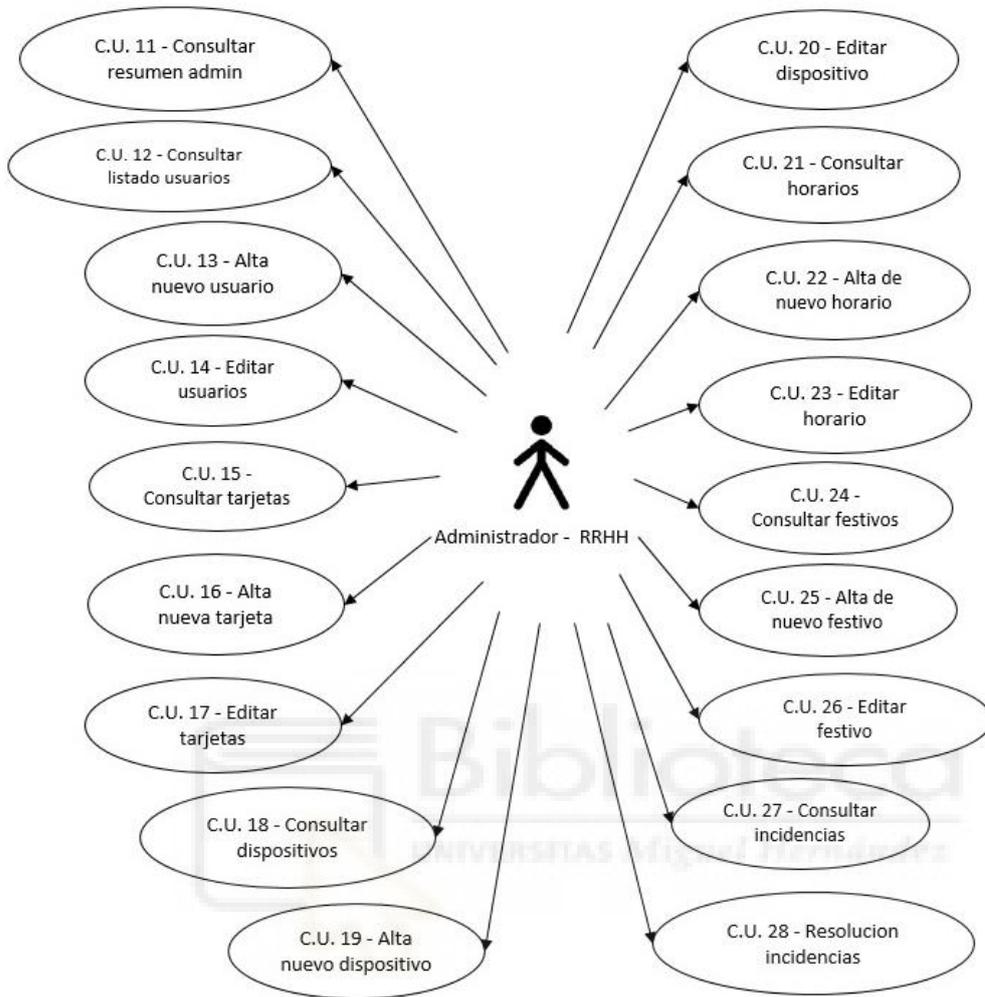


Imagen 26 - Diagrama casos de uso del administrador -RRHH

Administrador**Personal RRHH**

Descripción	<p>Este rol hereda las habilidades del Usuario – Trabajador y además incluye habilidades para la consulta y gestión de los datos de otros usuarios.</p> <p>Tiene habilidades para consultar la información (fichajes, horarios, datos personales...) de todos los usuarios registrados. Además, en quien se encarga de gestionar toda esta información.</p> <p>Es un perfil pensado para el personal de RRHH, que es el responsable de llevar el control de asistencia de todos los trabajadores.</p>
Casos de uso	<p>Todos los del actor Usuario – Trabajado</p> <p>C.U. 11 – Consultar resumen administrador</p> <p>C.U. 12 – Consultar listado de usuarios</p> <p>C.U. 13 – Alta de un nuevo usuario</p> <p>C.U. 14 – Editar usuario</p> <p>C.U. 15 – Consultar tarjetas</p> <p>C.U. 16 – Alta de nueva tarjeta</p> <p>C.U. 17 – Editar tarjeta</p> <p>C.U. 18 – Consultar dispositivos</p> <p>C.U. 19 – Alta de nuevo dispositivo</p> <p>C.U. 20 – Editar dispositivo</p> <p>C.U. 21 – Consultar horarios</p> <p>C.U. 22 – Alta de nuevo horario</p> <p>C.U. 23 – Editar horario</p> <p>C.U. 24 – Consultar festivos</p> <p>C.U. 25 – Alta de nuevo festivo</p> <p>C.U. 26 – Editar festivo</p> <p>C.U. 27 – Consultar incidencias</p> <p>C.U. 28 – Resolución de incidencias</p>

Tabla 2 - Casos de uso del administrador – RRHH

5.2.2 CASOS DE USO

Una vez identificados los actores en esta sección se detallan los distintos casos de usos en los que pueden intervenir. Esto dependerá del Rol asignado a cada tipo de usuario.

C.U. 1 Inicio de sesión

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Inicio de sesión en la web para poder acceder a los distintos servicios.
Precondición	Tener un usuario creado y activo.
Secuencia normal	P1: Acceder a la web P2: Completar los campos email y contraseña P3: Pulsar el botón 'Iniciar sesión' P4: La web envía los datos al servidor para validarlos.
Postcondición	La web muestra la página principal, la información mostrada será la correspondiente al usuario identificado. En el caso de los administradores además dispondrán de un menú de opciones adicionales.
Excepciones	1) Usuario o contraseña incorrecto: se recarga la pantalla de inicio de sesión informando del error. 2) Usuario no registrado o inactivo: se recarga la pantalla de inicio de sesión informado de que el usuario no se ha encontrado.
Comentarios	

Tabla 3 - C.U. 1: Inicio de sesión

C.U. 2 Consultar resumen

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Página que incluye información resumida para el trabajador sobre el horario que le corresponde para ese día, sus últimos fichajes y sus últimas incidencias.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Seleccionar en el menú principal la opción, 'Mi resumen'
Postcondición	La web mostrará la pantalla con los datos mencionados del usuario que ha iniciado sesión.
Excepciones	
Comentarios	Es la página que aparece por defecto para los usuarios - trabajadores

Tabla 4 - C.U. 2: Consultar resumen

C.U. 3 Consultar datos personales

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Consultar datos personales asociados al usuario tales como nombre, DNI...
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Pulsar en el icono de una persona situado en la barra superior a la derecha. P2: Seleccionar la opción, 'Perfil'
Postcondición	La web mostrará la pantalla con los datos personales del usuario que ha iniciado sesión.
Excepciones	
Comentarios	

Tabla 5 - C.U. 3: Consultar datos personales

C.U. 4 Editar datos personales

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Editar datos personales de la ficha del usuario
Precondición	<ul style="list-style-type: none">- Tener sesión iniciada con un usuario activo.- Acceder a la pantalla de consulta de datos personales (C.U. 3).
Secuencia normal	P1: Cambiar el dato que corresponda. P2: Pulsar el botón guardar.
Postcondición	Los datos modificados serán almacenados en la base de datos.
Excepciones	
Comentarios	No todos los datos del perfil son modificables por el usuario, en la pantalla de consulta de los datos personales aparecerán en formato de formulario aquellos que permitan edición.

Tabla 6 - C.U. 4: Editar datos personales

C.U. 5 Consultar fichajes propios

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Consultar los fichajes realizados por el usuario.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Pulsar en el menú superior la opción 'Mis fichajes'
Postcondición	La web mostrará una página con el listado de los fichajes realizados por el usuario identificado.
Excepciones	Si el usuario aún no ha realizado ningún fichaje la tabla estará vacía.
Comentarios	

Tabla 7 - C.U.5: Consultar fichajes propios

C.U. 6 Consultar incidencias propias

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Consultar incidencias del usuario identificado.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Pulsar en el menú superior la opción ‘Mis incidencias’, P2: Seleccionar en la tabla la incidencia que se quiere consultar.
Postcondición	La web mostrará una página con el detalle de la incidencia seleccionada: horario esperado, fichajes realizados, discrepancias detectadas y resolución aplicada.
Excepciones	Si el usuario no tiene ninguna incidencia, la tabla del paso 2 se mostrará vacía.
Comentarios	

Tabla 8 - C.U. 6: Consultar incidencias propias

C.U. 7 Consultar horario asignado

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Consultar el horario asignado al usuario identificado.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Pulsar en el menú superior la opción ‘Mi horario’,
Postcondición	La web mostrará una página con el detalle del horario asignado, que días se deben trabajar y las horas de entrada y salida.
Excepciones	Si el usuario no tiene ninguna incidencia, la tabla se mostrará vacía.
Comentarios	

Tabla 9 - C.U.7 - Consultar horario asignado

C.U. 8 Consultar festivos

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Consultar los días establecidos como festivo.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: Pulsar en el menú superior la opción 'Festivos',
Postcondición	La web mostrará una página que contiene una tabla con los días considerados festivos.
Excepciones	
Comentarios	

Tabla 10 - C.U. 8: Consultar festivos

C.U. 9 Cerrar sesión

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	Finaliza la sesión en curso del usuario.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: En el menú principal a la derecha aparece el nombre del usuario. Al pulsarlo se despliega un menú P2: Seleccionar la opción, 'Salir' del menú.
Postcondición	Se cerrará la sesión de usuario actual y se mostrará la pantalla de Inicio de sesión.
Excepciones	-
Comentarios	

Tabla 11 - C.U. 9: Cerrar sesión

C.U. 10**Fichar**

Actores principales	Trabajador / Administrador
Actores secundarios	-
Descripción	El usuario deberá emplear su tarjeta identificativa en las entradas o salidas de su puesto laboral, para ello hará uso de uno de los dispositivos habilitados en las instalaciones.
Precondición	<ul style="list-style-type: none"> - El dispositivo debe tener acceso wifi. - El dispositivo tiene que haber iniciado sesión correctamente y disponer del token de validación. - El usuario asociado al trabajador debe estar activo. - La tarjeta empleada debe estar activa y asociada al trabajador.
Secuencia normal	P1: Acercar la tarjeta identificativa al dispositivo.
Postcondición	<ul style="list-style-type: none"> - El dispositivo emitirá un sonido y encenderá un led para comunicar al usuario el resultado de la operación. - Si el fichaje es correcto se registrará en la base de datos la información de la tarjeta empleada, el dispositivo que se ha usado y la hora de la operación.
Excepciones	Si no se puede conectar con el servidor o si el usuario o la tarjeta no están activos el dispositivo emitirá un zumbido y se encenderá un led rojo indicando que ha fallado el fichaje.
Comentarios	Este caso de uso no se da en la web. Es una acción del usuario sobre el dispositivo físico de fichaje.

Tabla 12 - C.U. 10: Fichar

C.U. 11**Consultar resumen administrador**

Actores principales	Administrador
Actores secundarios	-

Descripción	Página que incluye información resumida sobre los últimos datos de todos los usuarios: últimos fichajes y sus últimas incidencias.
Precondición	Tener sesión iniciada con un usuario activo.
Secuencia normal	P1: En el menú de administración seleccionar la opción, ‘Resumen’
Postcondición	La web mostrará la pantalla con los datos mencionados del usuario que ha iniciado sesión.
Excepciones	
Comentarios	

Tabla 13 - C.U. 11: Consultar resumen administrador

C.U. 12 Consultar listado de usuarios

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar la lista total de usuarios.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción ‘Usuarios’
Postcondición	La web se actualiza para mostrar un listado con todos los usuarios registrados.
Excepciones	
Comentarios	Solo esta incluidos los usuarios tipo 1 (administrado - RRHH) y tipo 2 (usuarios – trabajador), es decir, aquellos que pertenecen a la clase UsuariosWeb (los de tipo 0 son los que pertenecen a la clase Device).

Tabla 14 - C.U. 12 - Consultar listado de usuarios

C.U. 13**Alta de un nuevo usuario**

Actores principales	Administrador
Actores secundarios	-
Descripción	Dar de alta un nuevo usuario en la web.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción ‘Usuarios’ P2: Pulsar el botón ‘Nuevo usuario’ de la parte superior. P3: Completar el formulario que aparece en pantalla con los datos del nuevo usuario. P4: Pulsar el botón ‘Guardar’
Postcondición	<ul style="list-style-type: none">- Si los datos son correctos el usuario será añadido a la base de datos.- La web vuelve a mostrar el listado de todos los usuarios, incluido el nuevo.
Excepciones	<ul style="list-style-type: none">- El correo electrónico debe ser único.
Comentarios	Este tipo de usuario (administrador) es el único que puede dar de alta otros usuarios. A diferencia de una web convencional los usuarios no pueden darse de alta por sí mismo. Una vez admitido en la empresa el administrador dará de alta el perfil y facilitará los datos de inicio de sesión al usuario - trabajador.

Tabla 15 - C.U. 13: Alta de un nuevo usuario

C.U. 14 **Editar usuario**

Actores principales	Administrador
Actores secundarios	-
Descripción	Editar un usuario existente.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo. Estar en la pantalla de consulta de usuarios (C.U. 12)
Secuencia normal	P1: Localizar el usuario que se desea modificar en la tabla. P2: Pulsar el botón ‘Editar’ de la parte derecha de la fila correspondiente al usuario. P3: Modificar los datos de la ficha del usuario seleccionado. P4: Pulsar el botón ‘Guardar’
Postcondición	- Los datos del usuario son actualizados en la base de datos.
Excepciones	Si los datos requeridos no están completos saldrá un aviso para completarlos.
Comentarios	

Tabla 16 - C.U. 14: Editar usuario

C.U. 15 **Consultar tarjetas**

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar todas las tarjetas registradas en la web y su estado actual.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción ‘Tarjetas’
Postcondición	La web se actualiza para mostrar un listado con todas las tarjetas registradas.
Excepciones	Si no hay tarjetas registradas la tabla se mostrará vacía.
Comentarios	

Tabla 17 - C.U. 15: Consultar tarjetas

C.U. 16 Alta de nueva tarjeta

Actores principales	Administrador
Actores secundarios	-
Descripción	Registrar una nueva tarjeta.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo. Estar en la pantalla de consulta de tarjetas (C.U. 15)
Secuencia normal	P1: Pulsar el botón ‘Nueva tarjeta’ de la parte superior. P2: Completar el formulario que aparece en pantalla con los datos de la nueva tarjeta. P3: Pulsar el botón ‘Guardar’
Postcondición	<ul style="list-style-type: none">- Si los datos son correctos la nueva tarjeta será añadido a la base de datos.- La web vuelve a mostrar el listado de todas las tarjetas, incluida el nueva.
Excepciones	
Comentarios	

Tabla 18 - C.U. 16: Alta de nueva tarjeta

C.U. 17 Editar tarjeta

Actores principales	Administrador
Actores secundarios	-
Descripción	Editar los datos de una tarjeta existente.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo. Estar en la pantalla de consulta de tarjetas (C.U. 15)
Secuencia normal	P1: Localizar la tarjeta que se desea modificar en la tabla. P2: Pulsar el botón ‘Editar’ de la parte derecha de la fila correspondiente la tarjeta. P3: Modificar los datos de la ficha de la tarjeta seleccionada. P4: Pulsar el botón ‘Guardar’

Postcondición	<ul style="list-style-type: none"> - Los datos de la tarjeta son actualizados en la base de datos - La web vuelve a mostrar el listado de todas las tarjetas, incluida el nueva.
Excepciones	
Comentarios	

Tabla 19 - C.U. 17: Editar tarjeta

C.U. 18 Consultar dispositivos

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar la lista de dispositivos creados y su estado.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción ‘Dispositivos’
Postcondición	La web se actualiza para mostrar un listado con todos los dispositivos registrados.
Excepciones	
Comentarios	Los dispositivos heredan de la clase usuarios, su identificador es tipo 0.

Tabla 20 - C.U. 18: Consultar dispositivos

C.U. 19 Alta de nuevo dispositivo

Actores principales	Administrador
Actores secundarios	-
Descripción	Dar de alta un nuevo dispositivo.
Precondición	<p>Tener sesión iniciada con un usuario de tipo administrador que este activo.</p> <p>Estar en la pantalla de consulta de dispositivos (C.U. 18)</p>

Secuencia normal	<p>P1: Pulsar el botón ‘Nuevo dispositivo’ de la parte superior.</p> <p>P2: Completar el formulario que aparece en pantalla con los datos del nuevo dispositivo.</p> <p>P3: Pulsar el botón ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Si los datos son correctos el dispositivo será añadido a la base de datos. - La web vuelve a mostrar el listado de todos los dispositivos, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 21 - C.U. 19: Alta de nuevo usuario

C.U. 20 **Editar dispositivos**

Actores principales	Administrador
Actores secundarios	-
Descripción	Editar el nombre o la ubicación de un dispositivo existente.
Precondición	<p>Tener sesión iniciada con un usuario de tipo administrador que este activo.</p> <p>Estar en la pantalla de consulta de dispositivos.</p>
Secuencia normal	<p>P1: Localizar el dispositivo que se desea modificar en la tabla.</p> <p>P2: Pulsar el botón ‘Editar’ de la parte derecha de la fila correspondiente al dispositivo.</p> <p>P3: Modificar los datos de la ficha del dispositivo seleccionado.</p> <p>P4: Pulsar el botón ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Los datos del dispositivo son actualizados en la base de datos - La web vuelve a mostrar el listado de todos los dispositivos, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 22 - C.U. 20: Editar dispositivo

C.U. 21 Consultar horarios

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar los tipos de horarios creados y ver el detalle de las franjas horarias que lo componen.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción 'Horarios'. La web se actualiza para mostrar un listado con todos los horarios creados. P2: Seleccionar el horario que se desea consultar para ver el detalle de las horas
Postcondición	La página se actualiza para mostrar el detalla de las horas que componen el horario.
Excepciones	
Comentarios	

Tabla 23 - C.U. 21: Consultar horarios

C.U. 22 Alta de un horario

Actores principales	Administrador
Actores secundarios	-
Descripción	Dar de alta un nuevo tipo de horario indicando las franjas horarias que lo componen.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo. Estar en la pantalla de consulta de horarios (C.U. 21)
Secuencia normal	P1: Pulsar el botón 'Nuevo tipo de horario' P2: Indicar el nombre del horario P3: Pulsar 'Guardar' P4: Localizar en la lista de horarios el que se acaba de crear

	<p>P5: Pulsar ‘Consultar detalle’</p> <p>P6: Pulsar ‘Nueva hora’</p> <p>P7: Introducir los datos del día y la hora y pulsar ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Si los datos son correctos el horario será añadido a la base de datos. - La web vuelve a mostrar el listado de todos los horarios, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 24 - C.U. 22: Alta de un horario

C.U. 23 Edición de un horario

Actores principales	Administrador
Actores secundarios	-
Descripción	<p>Editar un tipo de horario existente, ya sea su nombre o el detalle de las horas de entrada y salida para cada franja horario.</p>
Precondición	<p>Tener sesión iniciada con un usuario de tipo administrador que este activo.</p> <p>Estar en la pantalla de consulta de horarios (C.U. 21)</p>
Secuencia normal	<p><u>Editar el nombre del horario</u></p> <p>P1: Localizar el horario que se desea modificar en la tabla.</p> <p>P2: Pulsar el botón ‘Editar tipo’ de la parte derecha de la fila correspondiente al horario.</p> <p>P3: Modificar el nombre del horario seleccionado.</p> <p>P4: Pulsar el botón ‘Guardar’</p> <p><u>Editar las horas del horario</u></p> <p>P1: Localizar el horario que se desea modificar en la tabla.</p> <p>P2: Pulsar ‘Consultar detalle’</p> <p>P3: Localizar la hora que se quiere editar y pulsar ‘Editar’</p> <p>P4: Modificar el día o la hora y pulsar ‘Guardar’</p>

Postcondición	- Si los datos son correctos el horario será añadido a la base de datos.
Excepciones	
Comentarios	

Tabla 25 - C.U. 23: Edición de un horario

C.U. 24 Consultar festivos administración

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar la lista de festivos creados y su estado.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción 'Festivo'
Postcondición	La web se actualiza para mostrar un listado con todos los dispositivos registrados.
Excepciones	
Comentarios	Si es solo para consulta C.U.8 y C.U.24 muestran los mismos datos.

Tabla 26 - C.U. 24 - Consultar festivos administración

C.U. 25 Alta de nuevo festivo

Actores principales	Administrador
Actores secundarios	-
Descripción	Dar de alta un nuevo festivo.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo. Estar en la pantalla de consulta de festivos desde el menú de administración (C.U. 24)
Secuencia normal	P1: Pulsar el botón 'Nuevo festivo' de la parte superior.

	<p>P2: Completar el formulario que aparece en pantalla con los datos del nuevo festivo.</p> <p>P3: Pulsar el botón ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Si los datos son correctos el festivo será añadido a la base de datos. - La web vuelve a mostrar el listado de todos los festivos, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 27 - C.U. 25: Alta de nuevo festivo

C.U. 26 **Editar festivo**

Actores principales	Administrador
Actores secundarios	-
Descripción	Editar un festivo existente.
Precondición	<p>Tener sesión iniciada con un usuario de tipo administrador que este activo.</p> <p>Estar en la pantalla de consulta de festivos (C.U 25).</p>
Secuencia normal	<p>P1: Localizar el festivo que se desea modificar en la tabla.</p> <p>P2: Pulsar el botón ‘Editar’ de la parte derecha de la fila correspondiente al festivo.</p> <p>P3: Modificar los datos del festivo seleccionado.</p> <p>P4: Pulsar el botón ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Los datos del festivo son actualizados en la base de datos - La web vuelve a mostrar el listado de todos los festivos, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 28 - C.U. 26: Editar festivo

C.U. 27 Consultar incidencias

Actores principales	Administrador
Actores secundarios	-
Descripción	Consultar la lista de incidencias generadas y su estado.
Precondición	Tener sesión iniciada con un usuario de tipo administrador que este activo.
Secuencia normal	P1: En el menú de administración pulsar la opción 'Incidencias'. La web se actualiza para mostrar un listado con todas las incidencias generadas P2: Localizar la incidencia que se quiere consultar y pulsar 'Consultar' para ver los detalles de la misma.
Postcondición	La pantalla se actualiza para mostrar los detalles de la incidencia seleccionada.
Excepciones	

Tabla 29 - C.U. 27 - Consultar incidencias

C.U. 28 Resolución de incidencias

Actores principales	Administrador
Actores secundarios	-
Descripción	Resolver una incidencia asignándole una resolución de las ofrecidas por el sistema.
Precondición	<ul style="list-style-type: none">- Tener sesión iniciada con un usuario de tipo administrador que este activo.- Debe haber una incidencia registrada creada por una anomalía en los fichajes de un trabajador.- Estar en la pantalla de consulta de incidencias (C.U. 27)
Secuencia normal	P1: Localizar la incidencia que se desea modificar en la tabla. P2: Pulsar el botón 'Editar' de la parte derecha de la fila correspondiente.

	<p>P3: Modificar los datos de la incidencia seleccionado.</p> <p>P4: Pulsar el botón ‘Guardar’</p>
Postcondición	<ul style="list-style-type: none"> - Los datos de la incidencia son actualizados en la base de datos - La web vuelve a mostrar el listado de todas las incidencias, incluido el nuevo.
Excepciones	
Comentarios	

Tabla 30 - C.U. 28: Resolución de incidencias

5.3 API REST

La API Rest es la que permite la comunicación del dispositivo Arduino con la base de datos. Para comprobar su funcionamiento durante el desarrollo se empleó el programa Insomnia. A continuación, se pueden ver algunas capturas del programa que muestran los datos enviados, las cabeceras necesarias para la petición y la respuesta en cada caso.

Realiza 2 funciones:

- **Login**

El dispositivo tiene que autenticarse enviado en formato json el usuario (email) y contraseña tal y como se puede ver en la imagen 28.

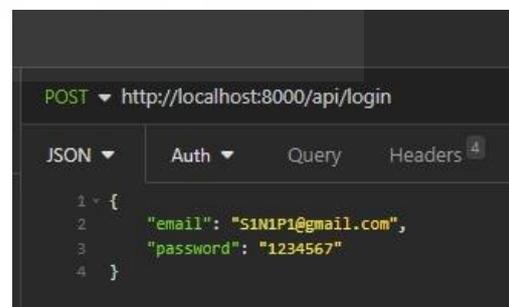


Imagen 27 - Insomnia: parámetros POST para login

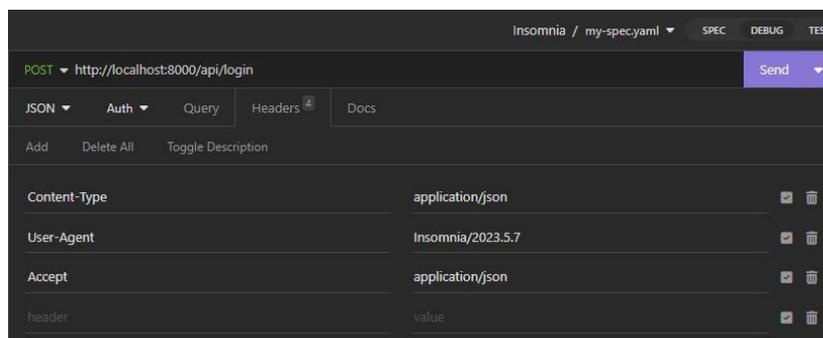


Imagen 28 - Insomnia: cabeceras POST para login

Si todo está correcto recibirá un fichero json con el token de sesión necesario para acceder a rutas protegidas y el código de finalización 200 (Imagen 29). Si la autenticación no es correcta, recibe el mensaje de error “Unauthorized” y el código 401 (Imagen 30).

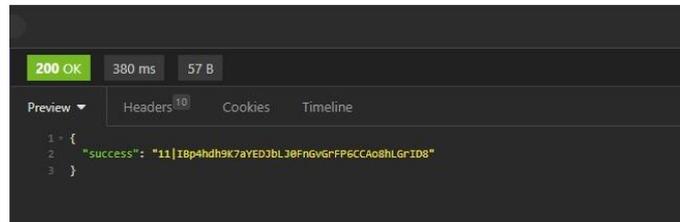


Imagen 29 - Insomnia: Respuesta login correcto

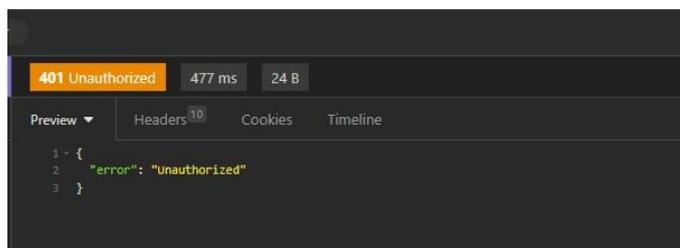


Imagen 30 - Insomnia: Respuesta login erróneo

- Fichaje

Cuando un usuario acerca una tarjeta al sensor RFID el dispositivo envía una petición POST a la API para registra el fichaje. Para poder realizar el fichaje el dispositivo tiene que estar autenticado como usuario e incluirlo en el envío. En este caso los datos de la tarjeta se envían en formato JSON (Imagen 32) y el token de sesión necesario para acceder a la ruta protegida se incluye como parte de la cabecera (Imagen 33).



Imagen 31 - Insomnia: parámetros POST para fichaje

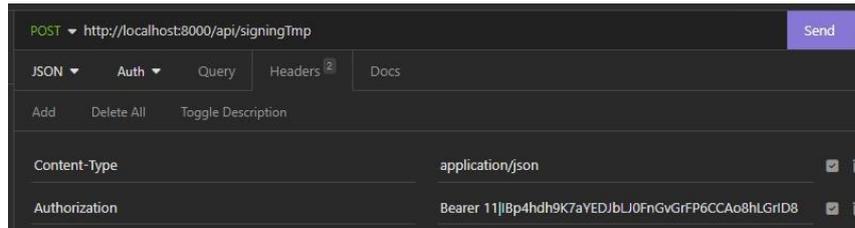


Imagen 32 - Insomnia: cabeceras PORT para fichaje

Si el fichaje tiene éxito la API devuelve 201 indicando que el fichaje ha sido registrado (Imagen 34), cualquier otro código de respuesta se considera un error al registrar el fichaje.

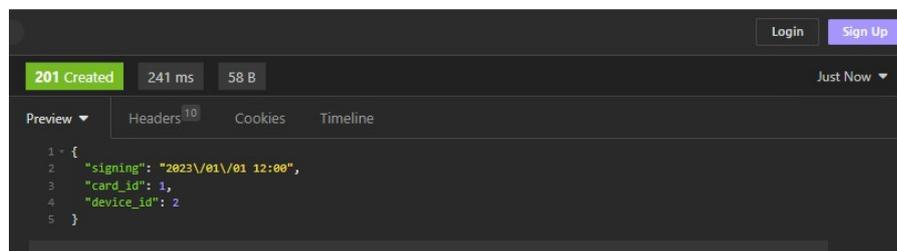


Imagen 33- Insomnia: Respuesta al fichaje

5.4 DISPOSITIVO

En lo referente al prototipo, por el momento se han detallado los componentes empleados (4.2 - Hardware), el entorno de desarrollo de la parte software (4.1.1.4 – Arduino IDE) y el lenguaje empleado (4.1.2.7 - Arduino). Este apartado se centra en el desarrollo del dispositivo, conexiones, librerías necesarias y la lógica del código.

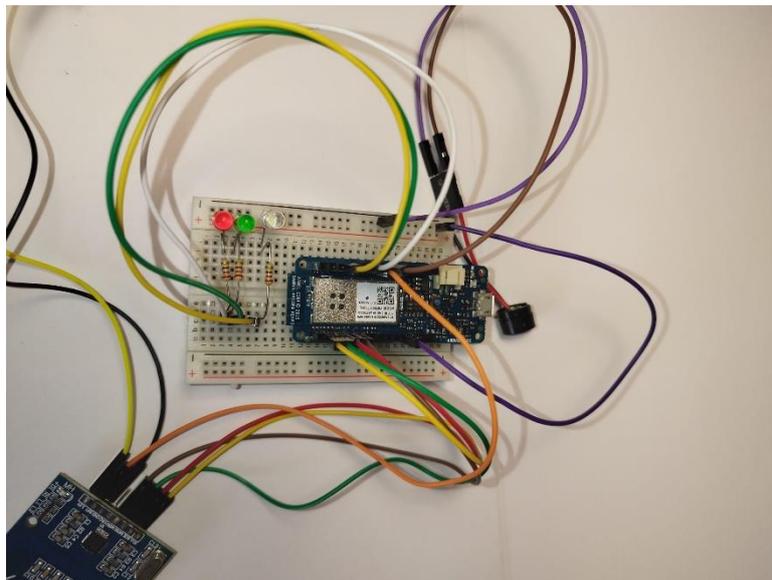


Imagen 34 - Prototipo 1ª versión

Para la presentación final de prototipo se ha optado por fabricar una caja con cierre imantado y a medida con una impresora 3D



Imagen 35 - Prototipo final (exterior)

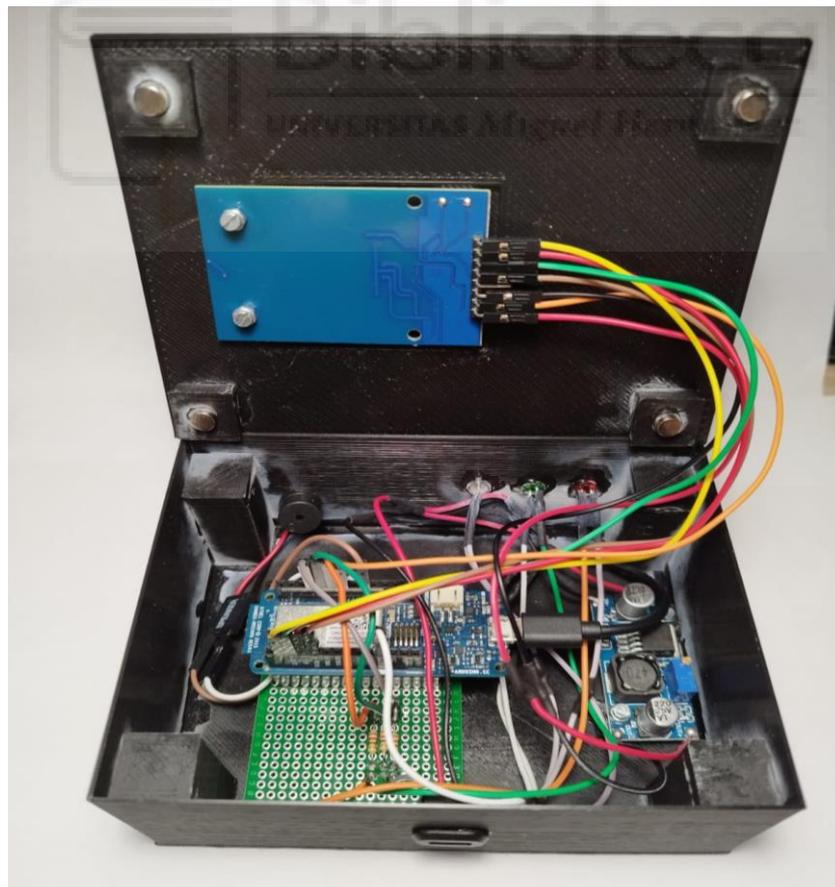


Imagen 36 - Prototipo final (interior)

5.4.1 CONEXIONES

Esquema de conexión del Arduino con la placa RFID y el resto de componentes del dispositivo.

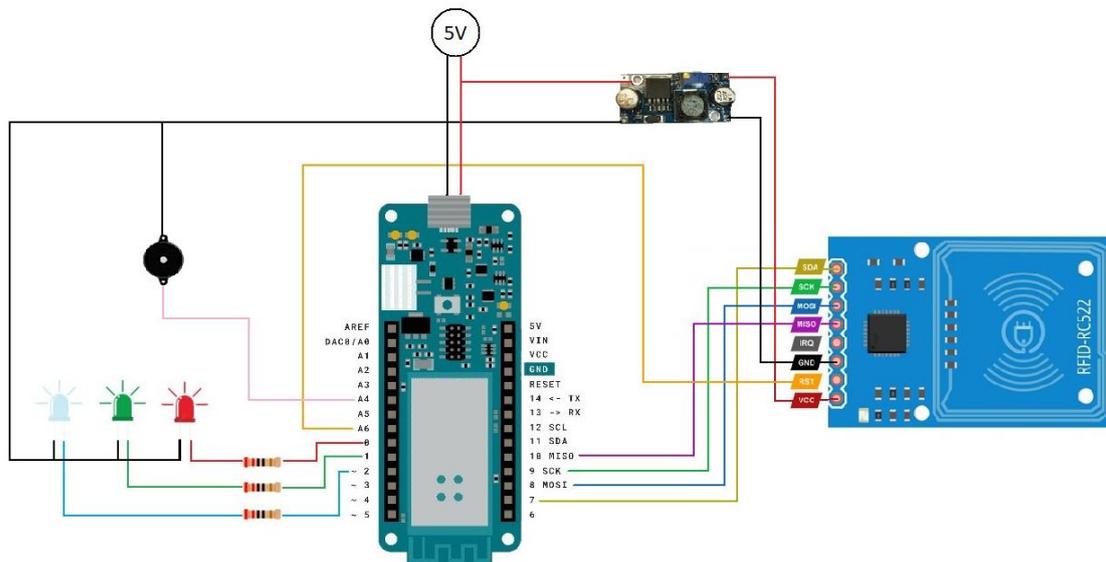


Imagen 37 - Esquema de conexiones del prototipo

5.4.2 LIBRERIAS

En este apartado se explican brevemente las librerías de Arduino empleadas para el proyecto.

- SPI.h

Incluye las funciones para el funcionamiento del bus SPI (Serial Peripheral Interface), este bus implementa una arquitectura maestro-esclavo entre la placa principal y el resto de dispositivos conectados. Se trata de una comunicación Full Duplex donde se pueden enviar y recibir datos simultáneamente.

- Wifi101.h

Librería de software libre que implementa un controlador web para dispositivos basados en el módulo WiFi ATMEL WINC1500. Permite conectar la placa directamente a una red wifi especificando el SSID y la contraseña. Es compatible con el cifrado WEP y WPA2.

- **MFRC522.h**

Proporciona las funciones necesarias la comunicación del Arduino principal con el módulo RFID conectado por los puertos SPI. Incluye funciones de lectura y escritura por radiofrecuencia de distintos tipos de tarjetas.

- **EasyBuzzer.h**

Librería compatible con todas las placas Arduino, permite configurar distintos tipos de pitidos con un zumbador.

5.4.3 FUNCIONAMIENTO

La función del dispositivo es establecer la conexión con la wifi asignada, conectar con el servidor para solicitar el token de sesión y permanecer a la espera hasta que detecte una tarjeta en el lector RFID. Al detectar la tarjeta avisará de la lectura al usuario mediante un zumbido y realizará una petición post a la API para registrar el fichaje, enviando: el ID de la tarjeta, el ID del dispositivo y el token de sesión. Después quedará a la espera de la respuesta sobre el resultado de la transacción y al recibirla, proporcionará feedback al usuario mediante la activación de un led y emitiendo un sonido. Tras esto, volverá a quedar a la espera de la siguiente lectura.

A continuación, explica el código empleado más en detalle siguiendo la estructura propia de Arduino explicada en el apartado 4.1.2.7.

- **Cabeceras y declaración de constantes**

En esta primera sección del código se incluyen las cabeceras de las librerías especificadas en el apartado anterior. También se incluye un fichero “arduino_secrets.h” donde se especifican las constantes que almacenan el nombre de la red wifi y la clave de acceso, se incluye en un fichero aparte para mantener el código limpio, además facilita el cambio de estos parámetros sin tener que modificar el fuente principal.

En esta sección también se declaran las constantes para especificar:

- La IP del servidor donde se aloja al API.
- La configuración para los pins RST_PIN y AA_PIN de la placa RFID.
- También creamos los objetos:

- MRDC522 mfrc522(SS_PIN, RST_PIN): para la conexión con la placa RFID
- WiFiCliente client: para la conexión a la red wifi.
- Se declara la variable global “Token” para almacenar el identificado de sesión segura con la API.
- Se declaran las constantes necesarias para los 3 led incluidos:
 - LEDR = 0; Led rojo, indica un error en el registro del fichaje.
 - LEDG = 1; Led verde, indica que el fichaje se ha registrado con éxito.
 - LEDW = 2; Led blanco que indica si la conexión con el servidor esta activa.

- **Función Setup()**

Esta función se ejecuta al conectar el Arduino a la corriente, las funciones de configuración que realiza son las siguientes:

- Inicializa el bus SPI y la placa NFC
- Configura los pins asignados a los leds como salidas.
- Inicializa el puerto serial
- Conecta con la wifi especificada, si no detecta la red o la contraseña es incorrecta queda en bucle hasta poder conectarse.
- Llama a la función LoginArduino en bucle hasta que almacena el token con éxito.
 - Conecta al servidor
 - Realiza una petición post enviando los datos del dispositivo para iniciar sesión
 - Lee y evalúa la respuesta para verificar el resultado y almacena el token devuelto:
 - Si el código es 200 (login correcto), enciende el led verde y emite un pitido.
 - En caso contrario, la variable Token se vacía al considerar la respuesta errónea, se enciende el led rojo y emite un pitido con una tonalidad distinta.
 - Para finalizar se desconecta del servidor.

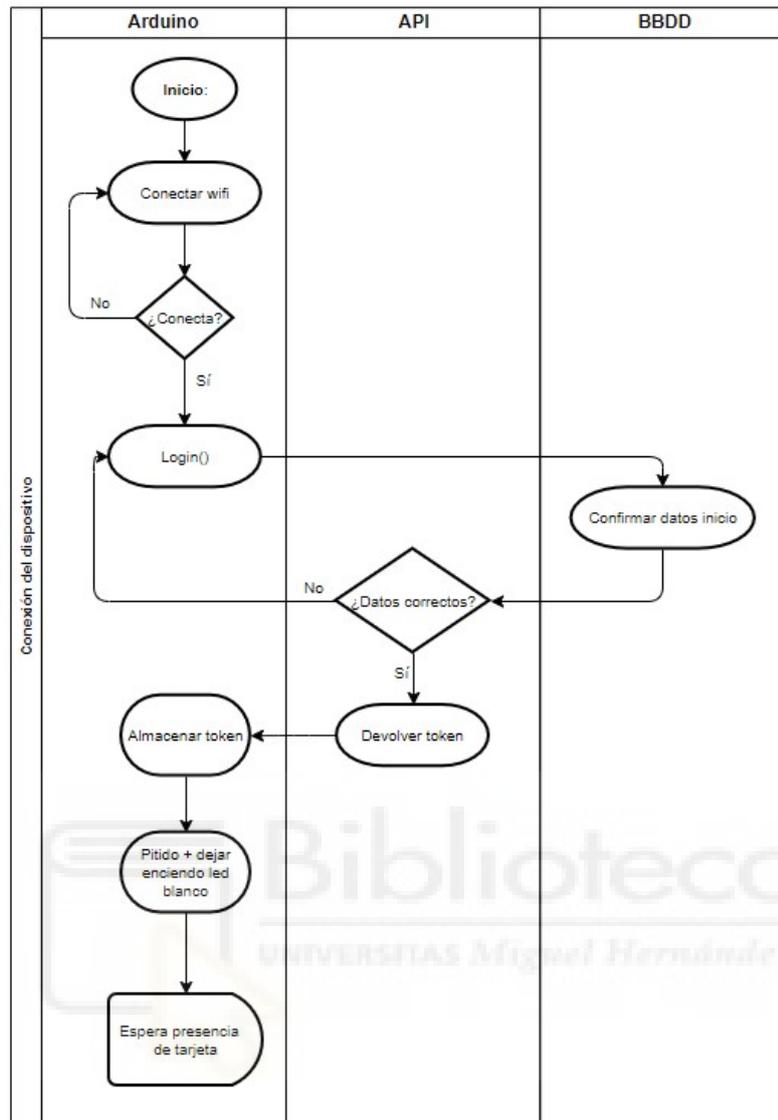


Imagen 38- Diagrama de flujo Setup() arduino

- Función loop()

Esta es la sección de código principal en todo programa Arduino, está diseñada para ejecutarse en un bucle continuo. En este caso, cada iteración servirá comprobar si se ha acercado alguna tarjeta RFID al dispositivo usando las funciones de la librería RFID.

- Comprobar mediante la función `PICC_IsNewCardPresent()` si alguna tarjeta se ha aproximado al lector, de no ser así, no se realizará ninguna acción y seguirá iterando.

- Si se detecta una tarjeta, el siguiente paso es emplear la función `PICC_ReadCardSerial()` para comprobar que la tarjeta detectada está en uno de los formatos aceptados y es posible leerla.
- Si el formato es correcto, leeremos tantos bytes como tamaño tiene la UDI (identificador de la tarjeta) y los almacenamos en una variable.
- Una vez terminada la lectura ejecutamos al función `Fichar()`, esta función que realiza los siguientes pasos:
 - Conectar con el servidor
 - Realizar una petición POST a la API enviando el token y la información obtenida de la tarjeta para insertar la información del fichaje en la base de datos.
 - Esperar hasta obtener una respuesta por parte del servidor.
 - Leer y procesar la respuesta.
 - Si devuelve el código 201, el fichaje ha sido realizado con éxito, enciende el led verde durante un segundo y lo apaga.
 - En caso contrario se realiza una comprobación adicional para comprobar si el código de la respuesta es 401, de ser así, significa que el dispositivo no está identificado correctamente por lo que se llama a la función `LoginArduino()` explicada anteriormente para volver a iniciar sesión.

Tras la comprobación e independientemente de si el error ha sido por un fallo en la conexión o no, se comunica al usuario que no se ha podido realizar el fichaje correctamente encendiendo durante un segundo el led rojo del dispositivo.
 - Para finalizar se desconecta del servidor.

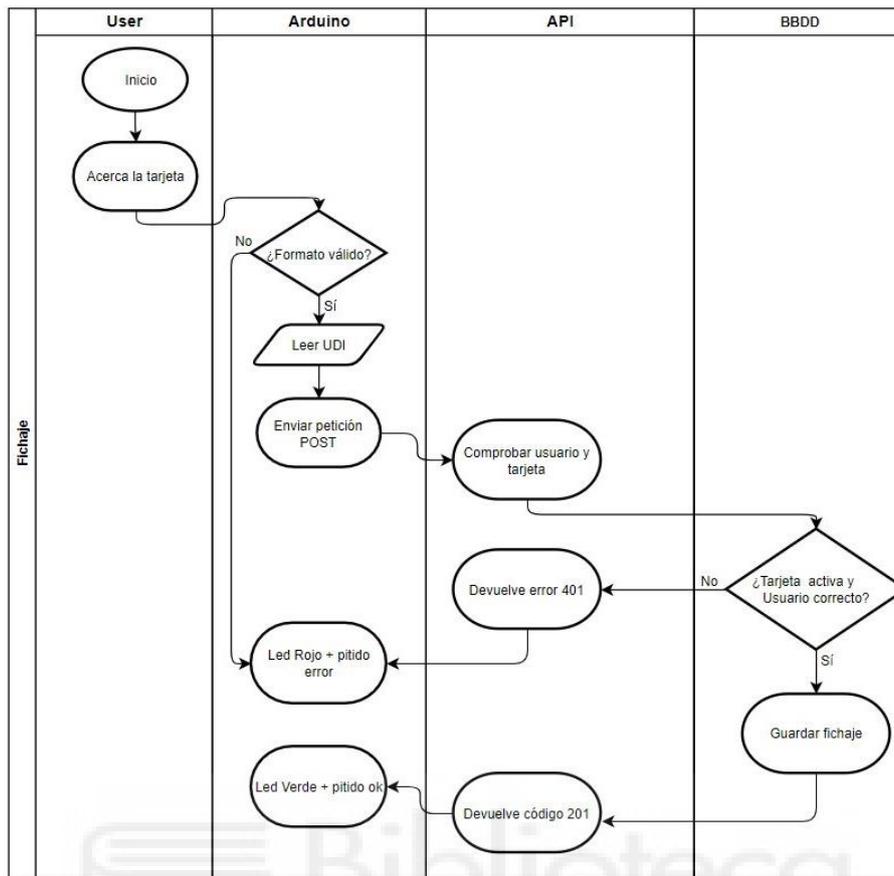


Imagen 39 - Diagrama de flujo proceso fichaje

5.5 BASE DE DATOS

La base de datos consta de 12 tablas propias que almacenan los datos referentes a usuarios, fichajes, horarios, festivos, vacaciones y tarjetas, además de 4 adicionales que se crean automáticamente al habilitar la autenticación con breezer. También ha sido necesaria la creación de 4 funciones dentro de la base de datos y 9 procedimientos almacenados que se encargan del procesado de los datos y generación de incidencias.

Toda la composición de la base de datos se ha implementado mediante migraciones en laravel y los datos de muestra necesarios para las pruebas y la demostración de uso se completan mediante seeders.

5.5.1 ANÁLISIS

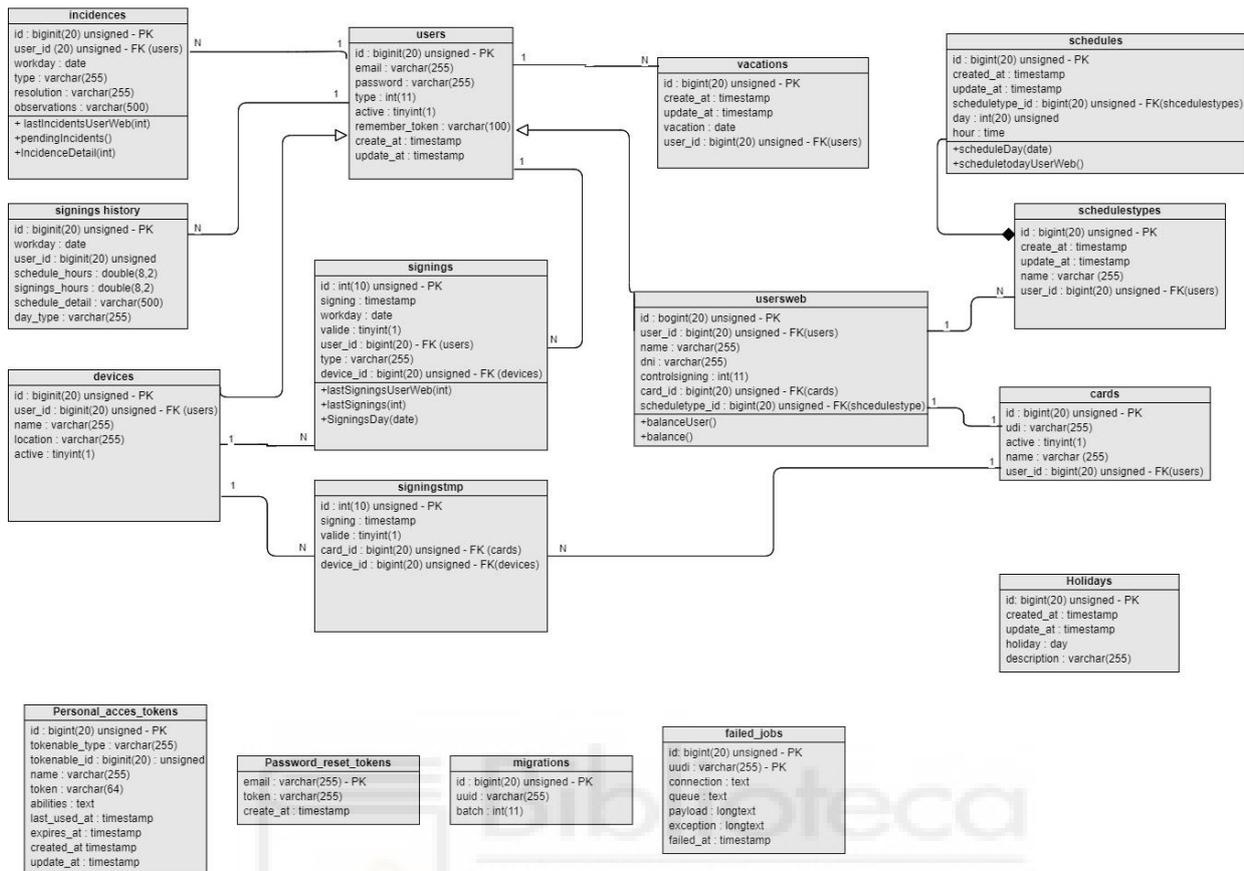


Imagen 40 - Diagrama de clases

En este primer apartado los campos que componen cada modelo, proporcionando una breve descripción de su funcionalidad, posibles valores y otras características.

Users

- **Type:** indica cuál de los dos tipos de usuario se trata, en función de su valor se clasificará como device (valor = 0) o como userWeb (administrador/RRHH = 1, usuario/ trabajador = 2).

Signings

- **Type:** existen 2 tipos posibles. Por una parte, aquellos fichajes que se registran de forma automática cuando el trabajador ficha en el dispositivo (valor = A), estos fichajes no se pueden borrar ni editar, solo se pueden alterar para considerarlos válidos o no válidos. Por otro lado, están los fichajes de creación manual (valor = M) que son creados por los administradores con el fin de regularizar una

incidencia. Estos fichajes se pueden borrar y editar. Esta utilidad no se ha podido implementar por falta de tiempo, pero el modelo ha quedado preparado para llevarlo a cabo en una mejora futura.

- **lastSigningsUserWeb(N)**: Obtiene un listado de los último N fichajes realizados por el usuario autenticado.
- **lastSignings(N)**: Devuelve una lista con los últimos N fichajes realizados.
- **SigningsDay()**: Recupera los fichajes del día indicado por parámetro para el usuario autenticado.

SigningsTmp

- **Valide**: indica si se trata un fichaje válido (valor = 1) o no válido (valor = 0). Al calcular el cómputo total de horas y realizar comprobaciones de puntualidad para las incidencias los fichajes no válidos son ignorados.

UsersWeb

- **ControlSigning**: es necesario para contemplar la casuística que se especifica en la ley que indica que algunos usuarios están exentos del control horario debido a las funciones que desempeñan (comerciales, gerencia...). Con este campo se indica si es necesario llevar un control sobre sus fichajes y si la ausencia de esto generara incidencias (valor = 1) o está exento (valor = 0).
- **balanceUser()**: Obtiene el balance total de horas acumuladas por el usuario autenticado. Cuando una incidencia se resuelve como 'Balance' la diferencia entre horas fichadas y horas del horario componen el balance. Si el balance es negativo el trabajador debe horas a la empresa, si es positivo la empresa le debe horas libres.
- **balance**: Obtiene el balance total de horas para el usuario indicado por parámetro.

Schedule

- **ScheduletodayUserWeb()**: obtiene el horario del día para el usuario autenticado.
- **ScheduleDay(date)**: obtiene el horario asignado al usuario autenticado para el día indicado por parámetro.

Incidences

- **Type**: indica de que tipo es la incidencia: Puntualidad, horas totales... Los tipos posibles se explican con más detalle en el apartado 5.5.3.2

- **lastIncidentsUserweb(N)**: Rescata las últimas N incidencias del usuario autenticado.
- **pendingIncidents()**: Obtiene un listado de todas las incidencias sin resolución, es decir, aquellas pendientes de procesar.
- **incidenceDetail(N)**: Recupera la información de la incidencia especificada por parámetro.

SigningHistory

- **Schedule_hours**: guarda el total de horas que debe realizar el trabajador en la jornada indicada.
- **Signings_hours**: guarda el número real de horas trabajadas calculado en base a los fichajes.
- **ScheduleDetail**: indica el horario de la hornada para guardarlo a modo de histórico en formato “Lunes: 9:00- 14:00 -15:00 – 18:00”.
- **Day_type**: almacena el tipo de día que es para el trabajador esa jornada: Labora, no labora, festivo, vacaciones.

5.5.2 FUNCIONES

Las 3 funciones que se han implementado se emplean para realizar pequeñas consultas recurrente de forma ágil y permiten una mayor limpieza en el código ya que evitan duplicidades. Son las siguiente:

- **DayNameES**: recibe una fecha y devuelve el nombre del día al que corresponde (Lunes, Martes, Miércoles...), es un equivalente a la función estándar SQL DayName pero esta devuelve el nombre del día en castellano.
- **DatWeek**: similar a la anterior, también devuelve el nombre del día, pero en este caso recibe un entero. En SQL se dispone de la función DayOfWeek que al pasarle una fecha te devuelve un entero que corresponde al día de la semana, siendo Domingo = 0, Lunes = 1 y así sucesivamente.
- **TypeOfDay**: esta función es más específica para este proyecto, recibe como parámetro un usuario y una fecha. La función realiza una consulta a distintas tablas: vacaciones, festivos, horario del trabajador (para saber si la fecha pertenece a un día laboral o no labora) y devuelve que tipo de día que es la fecha para ese usuario en concreto.

- **FormatBalance:** en la tabla de histórico los datos sobre las horas totales trabajadas en una jornada y las horas totales indicadas en el horario se guardan en formato decimal para facilitar su manipulación. Pero cuando se quiere mostrar al usuario es poco intuitivo decir que ha trabajado 1,9 horas que en realidad es 1 hora 54 min. Esta función realiza esa conversión de decimal a un string con un formato más ameno. También se emplea para mostrar las horas totales del balance de cada usuario.

5.5.3 PROCEDIMIENTOS ALMACENADOS Y FUNCIONES

Para ciertas tareas en el procesamiento de los datos se ha optado por hacer uso de procedimientos almacenados en la base de datos.

Las 2 tareas que requieren el uso de procedimientos son la agrupación en jornadas de los fichajes y la generación de incidencias, a continuación, se detallan cada una de ellas, su cometido, como funciona y de que procedimientos y funciones hace uso.

5.5.3.1 PROCESADO DE FICHAJES

Esta tarea se centra en la agrupación de los fichajes acumulados en jornadas laborales. Los fichajes recogidos del dispositivo tienen una tarjeta asociada y un día/hora, habitualmente los fichajes asociados a una jornada coincidirán con la fecha de la jornada. La excepción a esta casuística son los turnos de trabajo nocturnos que empiezan un día y terminan en el siguiente.

Para poder calcular correctamente la duración de la jornada y poder realizar las comprobaciones pertinentes de puntualidad y horas totales trabajadas los fichajes se procesan y agrupan en jornadas tal y como se indica en el esquema incluido a continuación.

Para esta tarea hay 2 variables configurables que serán necesarias:

Tarjeta	Fichaje	Jornada
1	01/01/23 08:57	01/01/2023 Turno normal: día del fichaje y jornada coinciden
1	01/01/23 14:03	
1	01/01/23 15:01	
1	01/01/23 18:05	
2	01/01/23 19:02	01/01/2023 Turno nocturno: un fichaje del día 2 pertenece a la jornada del día 1
2	01/01/23 22:30	
2	01/01/23 23:01	
2	02/01/23 03:32	

Imagen 41 - Ejemplo tipos de jornadas

- **Fichajes duplicados** (preconfigurado a 5 min): Esta variable indica el tiempo máximo que puede pasar entre 2 fichajes del mismo trabajador para considerarlo fichajes duplicado, es decir el trabajador ha pasado la tarjeta por el sensor 2 veces. Cuando esto sucede aquellos fichajes que se consideran duplicados se marcan como ‘no validos’ y no se tendrán en cuenta para el cálculo de las horas por rangos. Un rango de trabajo se calcula restando el fichaje de salida menos el de la entrada anterior para saber el tiempo transcurrido.

Para diferenciar fichajes de entrada de los de salida se tiene en cuenta la siguiente lógica, si una jornada tiene 4 fichajes (ordenados por hora) se considera que aquellos en una posición impar son entradas (el fichaje 1 y 3) y las posiciones pares son salidas (2 y 4).

En el caso de detectar fichajes duplicados el razonamiento que se aplica para decidir cuál de los 2 es válido es la siguiente:

- Si se trata de fichajes de entrada el primer fichaje realizado es el válido.

Datos originales			Comprobación				Conclusión	
Nº	Fichaje	Tiempo	Fichaje	Nº	Tipo fichaje (supuesto)	Tiempo	Fichaje	Válido
1	01/01/23 08:57	3 min	01/01/23 08:57	1	Entrada	3min	01/01/23 08:57	Sí
2	01/01/23 09:00		01/01/23 09:00	2	¿ Salida ?		01/01/23 09:00	No
3	01/01/23 14:03	5h 3 min					01/01/23 14:03	Sí
4	01/01/23 15:01	1h 4 min					01/01/23 15:01	Sí
5	01/01/23 18:05	3h 6 min					01/01/23 18:05	Sí

Como el tiempo entre estos 2 fichajes esta dentro de los parámetros para considerarlos duplicados, como se trata de una **ENTRADA** el válido es el **PRIMERO** de ellos

Imagen 42 - Ejemplo fichajes duplicados (entrada)

- Si se trata de fichajes de salida el último fichaje realizado es el válido.

Datos originales			Comprobación				Conclusión	
Nº	Fichaje	Tiempo	Fichaje	Nº	Tipo fichaje (supuesto)	Tiempo	Fichaje	Válido
1	01/01/23 08:57	5h 6 min	01/01/23 18:02	4	Salida	3min	01/01/23 08:57	Sí
2	01/01/23 14:03		01/01/23 18:05	5	¿ Entrada ?		01/01/23 14:03	Sí
3	01/01/23 15:01	1h 4 min					01/01/23 15:01	Sí
4	01/01/23 18:02	3h 6min					01/01/23 18:02	No
5	01/01/23 18:05	3 min					01/01/23 18:05	Sí

Como el tiempo entre estos 2 fichajes esta dentro de los parámetros para considerarlos duplicados, como se trata de una **SALIDA** el válido es el **ÚLTIMO** de ellos

Imagen 43 - Ejemplo fichajes duplicados (salida)

Esta clasificación de los fichajes se lleva a cabo en el procedimiento DuplicateSignings que se explica más adelante.

- **Jornada máxima** (preconfigurado a 13h): se emplea para detectar posibles errores en los fichajes, si el cálculo de la jornada por rangos de fichaje entrada-

salida (solo de aquellos fichajes considerados válidos, es decir, no duplicados) es superior a 13 horas lo más probable es que el trabajador se haya olvidado de realizar uno de los fichajes. Al seleccionar los fichajes por pares para calcular los rangos se ha seleccionado un fichaje que pertenece a la jornada siguiente, de ahí el exceso de horas calculadas.

- Ejemplo: fichajes correctos

Tarjeta	Fichaje	Nº fichaje	Tipo fichaje (supuesto)	Horas trabajadas	Total jornada
2	01/01/23 19:02	1	Entrada	3h 28min	7h 59min
2	01/01/23 22:30	2	Salida		
2	01/01/23 23:01	3	Entrada	4h 31min	
2	02/01/23 03:32	4	Salida		

Imagen 44 - Ejemplo cálculo de jornada (correcto)

Para los datos de la imagen 45 el tiempo total de la jornada está dentro de lo aceptado, por lo tanto, todos los fichajes pertenecen a la jornada del 01/01/23, aunque el último se haya realizado el día 2.

- Ejemplo: fichajes erróneos

Tarjeta	Fichaje	Nº fichaje	Tipo fichaje (supuesto)	Horas trabajadas	Total jornada
2	01/01/23 19:02	1	Entrada	3h 28min	24h 27min
2	01/01/23 22:30	2	Salida		
2	01/01/23 23:01	3	Entrada	20h 59min	
2	02/01/23 20:00	4	Salida		

Imagen 45 - Ejemplo cálculo de jornada (incorrecto)

En este segundo ejemplo correspondiente a la imagen 46, el total de horas trabajadas supera el máximo fijado para una jornada plausible lo que indica que hay algún error en el fichaje. El fichaje nº 4 que se ha asumido una salida en realidad es una entrada, el trabajador se olvidó de fichar a la salida. En conclusión, Los fichajes 1,2,3 son de la jornada 01/01/23 y el fichaje 4 pertenece a la jornada del 02/02/23.

Una vez aclarados estos conceptos básicos a continuación se incluye una traza del proceso completo.

- En primer lugar, se identifican los trabajadores que tienen indicado en su ficha el control de jornada (la ley contempla excepciones como gerencia o comerciales que están exentos).

- Para cada uno de esos trabajadores se comprueban los fichajes pendientes de procesar (tabla SIGNINGTMP), se comprueba si existen duplicados y se marcan como válidos o no.
- Los fichajes agrupan en la jornada indicada en el primero de ellos, si se detecta un cambio de día se comprueba si:
 - Si el total de los fichajes (válidos) procesados hasta el momento son pares se considera que la jornada está completa y el fichaje con otra fecha pertenece a otra jornada.
 - Si el total de los fichajes (válidos) procesados hasta el momento son impares, se evalúa si el fichaje con fecha distinta puede formar parte de la jornada. Esta comprobación se realiza calculando cuantas horas tendría la jornada si se incluye ese fichaje, si el resultado no supera las horas máximas consideradas para una jornada el fichaje se incluirá en la jornada. Si el total es superior a la jornada máxima establecida se asume que ha habido un error por parte del trabajador (se ha olvidado de fichar a la entrada o salida) y el número de los fichajes de esa jornada es impar. Esto ocasiona que no se puedan calcular las horas trabajadas, lo cual generará una incidencia, este proceso se detalla en el apartado siguiente.
- Una vez agrupados los fichajes se copian a la tabla definitiva SIGNING indicando la jornada asignada y se borra de la temporal (SIGNINGTMP).
- Se comprueba que tipo de día es la jornada (haciendo uso de la función TypeOfDay explicada anteriormente)
 - **Laboral, no laboral:** se considera laboral si el día está incluido en el horario del trabajador y no laboral en caso contrario.
 - **Festivo:** si el día está incluido en la tabla donde se registran los festivos anuales (HOLIDAYS).
 - **Vacaciones:** si el día de la jornada está incluido en la tabla que guarda las vacaciones asociadas a ese trabajador (VACATIONS).
- Tras eso se guarda un histórico en la tabla SIGNINGSHISTORY donde se almacenan las horas totales trabajadas, el detalle del horario asignado al trabajador ese día y el tipo de día.
- Se comprueba si la jornada tiene incidencias, es decir, si hay discrepancias entre el horario asignado y los fichajes realizados y si las horas totales trabajadas se corresponden con las que se esperaban.

- Esta secuencia se repite hasta procesar todos los fichajes ubicados en la tabla temporal. El único caso en el que permanecerá un fichaje sin procesar será si es el único de su grupo y no hay posteriores.

Por ejemplo: si los fichajes se procesan a las 12 p.m. y un trabajador ha iniciado su turno nocturno a las 10 p.m. el fichaje de las 10 p.m. es la entrada, pero no tiene un fichaje de salida porque esta se producirá a las 6 a.m. del día siguiente. En el momento de procesar los fichajes aún no se ha realizado por lo que el fichaje de las 10 p.m. quedará pendiente en la tabla temporal y será procesado al día siguiente.

Para finalizar, el proceso realiza una comprobación basándose en los días registrado en la tabla de histórico para comprobar si no se ha fichado en algún día laboral. De ser así, crea un registro en el histórico con total de horas en la jornada 0 y se genera la incidencia correspondiente. Esta casuística se da cuando un trabajador no se presenta a trabajar en un día laboral y por lo tanto no hay fichajes de ese día en la tabla temporal.

Los procedimientos empleados para realizar todo este proceso son los siguientes:

- **SigninsTmpToSignings**

Este es el procedimiento principal, se ejecutará de forma programada todas las noches y llevará a cabo todo el proceso mediante llamadas a distintos procedimientos adicionales. La tarea programada para la ejecución de este procedimiento dependerá del entorno en el que se suba la web, ya sea una tarea programada en Windows o el uso de un Cron en Linux.

Este procedimiento en primer lugar itera sobre los usuarios (USERWEB) que estén activos y tengan una tarjeta de fichaje asociada, para cada uno de ellos realiza 2 acciones:

- Llamada al procedimiento duplicateSignings, indicando la tarjeta.
- Llamada al procedimiento makeWorkday

- **DuplicateSignings**

Este procedimiento se encarga de comprobar la duplicidad de los fichajes y clasificarlos en válidos o no válidos. Recibe como parámetro de entrada el id de una tarjeta y procesa todos los fichajes de la tabla SIGNINGSTMP tal y como se ha explicado anteriormente en la traza del proceso.

- **makeWorkday**

Recibe como parámetro de entrada el id del usuario y el id de la tarjeta asociada. Su cometido principal es iterar sobre cada fichaje de la tabla SIGNINGSTMP y en base a la fecha y el cálculo de horas totales asociar los fichajes a la jornada correspondiente. Una vez identificados estos fichajes que compone una jornada realiza los siguientes pasos:

- Llamada al procedimiento transferSignings
- Llamada al procedimiento updateSigningsHistory
- Llamada al procedimiento checkIncidences

Este proceso se repetirá hasta que no queden fichajes en la tabla o hasta que se detecte que los últimos pertenecen a la última jornada y esta está incompleta.

- **transferSignings**

Una vez se han identificado los fichajes que se asocian a una jornada, este procedimiento se encarga de insertarlos en la tabla definitiva SIGNINGS y de borrarlos de la tabla temporal SIGNINGSTMP.

- **updateSigningsHistory**

Este procedimiento edita o crea (en el caso de que no exista), el resumen del registro de la jornada en la tabla SIGNINGSHISTORY. Este resumen permite para cada jornada tener un histórico precalculado de las horas fichadas y las indicadas en el horario.

- **checkIncidences**

La funcionalidad principal de este procedimiento es detectar irregularidades en los fichajes. Que se considera una irregularidad y como se detectan esta explicado más en detalle en el apartado siguiente.

5.5.3.2 GENERACIÓN DE INCIDENCIAS

Se considera incidencia toda aquella discrepancia entre los fichajes y el horario asignado, puede ser por falta de puntualidad, horas de menos u horas de más. Solo es posible que se genere una incidencia por día y usuario, esta incluirá todas las irregularidades

detectadas. Este procedimiento emplea una variable que aporta cierto margen a los cálculos:

- **Margen en puntualidad** (preconfigurado en 240 segundos / 4 min): no se considera impuntualidad si la diferencia entre el fichaje y la hora indicada por el horario es inferior a 4 minutos.
 - En el caso de los fichajes de entrada este margen se aplica a los 4 minutos posteriores al horario, si el trabajador llega antes no cuenta como incidencia en la puntualidad, solo cuenta si entra con retraso.
 - En el caso de los fichajes de salida el margen se aplica a los minutos previos a la salida indicada en el horario, se el trabajador sale más tarde tampoco se consideraría una incidencia por puntualidad.

Tipo	Horario	Fichaje	Valoración
Entrada	8:00	7:55	Puntual: Llega antes de tiempo (el margen no aplica)
Entrada	8:00	8:05	Impuntual: Llega despues de su hora, supera lo 4 min de magen
Salida	18:00	18:07	Puntual: Sale despues de su hora (el margen no aplica)
Salida	18:00	17:50	Impuntual: Sale antes de tiempo, supera lo 4 min de magen

Imagen 46 - Ejemplo impuntualidad

- **Horas trabajadas** (preconfigurado a 0 segundos): No se ha aplicado margen para el total de horas mínimas que debe trabajar el usuario, si el horario indica 8h totales y el trabajador realiza 7:59 saltará una incidencia.
- **Horas extra** (preconfigurado a 1500 segundos / 25 min): esta variable indica a partir de que exceso en las horas totales de la jornada se considera que pueden empezar a contar como horas extra. Es decir, si el horario indica 8h y el trabajador realiza 8h 15min no saltará incidencia porque solo hay 15 min. de más, pero si el exceso supera los 25 min indicados en el margen se generará una incidencia para informar del hecho y considerar si se asigna ese exceso como hora extra.

Se han contemplado 3 tipos de incidencias:

- **Error en el nº de fichajes:** si el nº de fichajes de la jornada es impar significa que el usuario se ha olvidado de fichar en una de las entrada o salidas. También es posible que un trabajador tenga menos fichajes de los indicados en el horario (si

no ha parado a comer, por ejemplo) o que tenga más (si ha tenido que hacer una parada adicional en su jornada).

- **Por puntualidad:** si el fichaje se diferencia del horario tal y como se ha indicado al explicar el margen de puntualidad.
- **Por horas totales:** esta incidencia se produce si las horas totales fichadas discrepan con las horas totales del horario. Engloba las incidencias provocadas por horas de más y horas de menos.
- **Por puntualidad y horas:** esta categoría incluye las incidencias que han detectado discrepancias en los dos puntos anteriores.

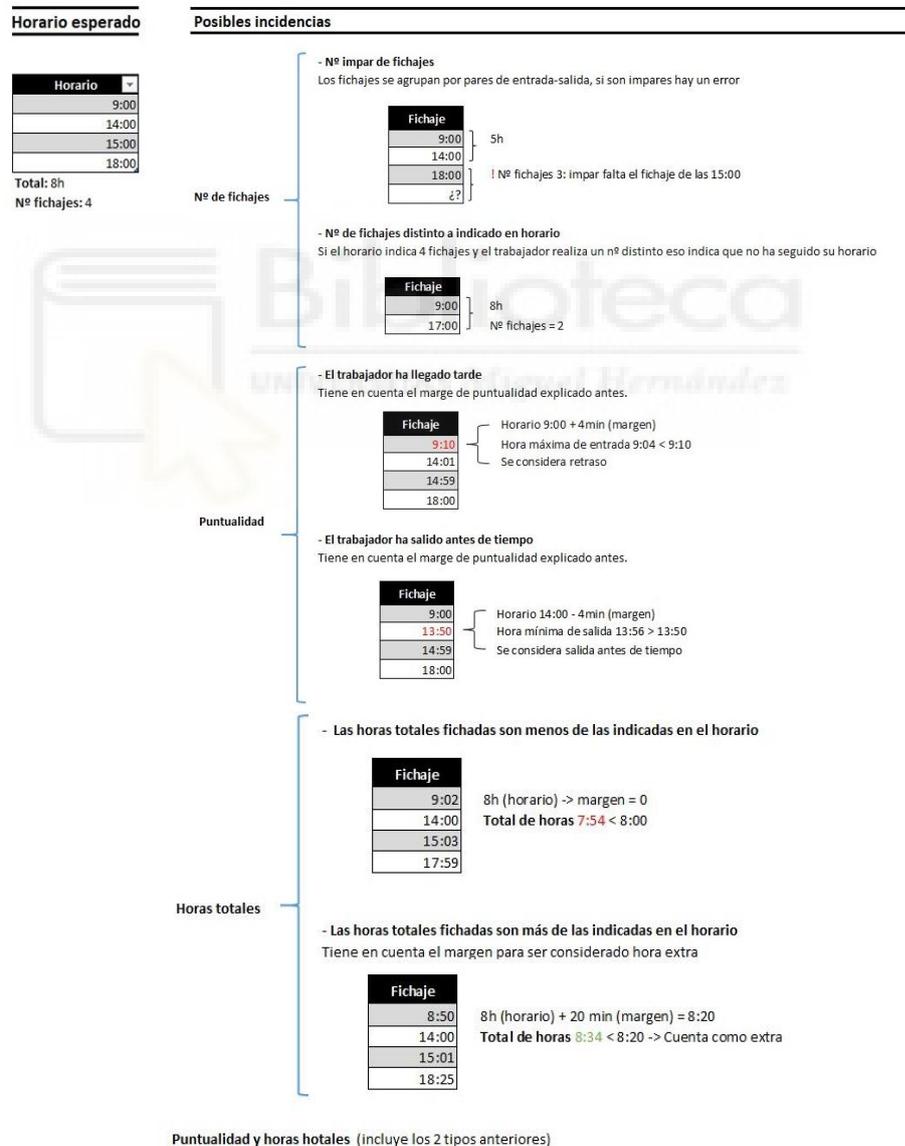


Imagen 47 - Tipos de incidencias

El procedimiento sigue los siguientes pasos para detectar las incidencias:

- Comprobar que el trabajador tenga un horario asignado, de no ser así se incluye en las observaciones de la incidencia: *'El usuario no dispone de horario asignado'*.
- Si dispone horario se recopilan datos del mismo: total de horas y nº de fichajes esperados. Esto se hace con el procedimiento almacenado ScheduleResume que se explicara más adelante.
- Se emplea la función TypeOfDay para obtener el tipo de día al que pertenece la jornada (Labora, No labora, Festivo, Vacaciones) para ese usuario en esa jornada.
- Comprobar que el nº de fichajes no es impar, de serlo la incidencia incluiría entre las causas de la incidencia: *'Error: N° impar de fichajes'*. Los fichajes se componen por parejas entrada-salida para poder comprobar el tiempo entre ellos, si son impares es porque se ha producido un error (posiblemente el usuario a olvidado fichar una de las veces).
- Comprobar que el nº de fichajes coincide con el indicado en el horario, de no ser así se indica como causa de la incidencia: *'N° incorrecto de fichajes: se esperaban X y se han registrado Y'*
- El siguiente paso es comprobar uno a uno los fichajes para controlar la puntualidad (la reglas sobre que se consideran puntualidad se han expuesto al principio de la sección). Si se detecta alguna irregularidad se incluirá en el detalla de la incidencia: *'Puntualidad: horario (XX:XX), fichaje (YY:YY)'*. Este proceso lo lleva a cabo el procedimiento checkPuntuality explicado más adelante.
- Una vez evaluados los fichajes se comprueban las horas totales, si son inferiores a las indicadas en el horario o las superan (teniendo en cuenta los márgenes explicados al principio del apartado) se incluirá en la causa de la incidencia: *'Horas totales: se han realizado [más/menos] horas (XX:XX) de las indicadas en el horario (YY:YY)'*
- La última verificación es para comprobar si hay fichajes en un día no laboral, en ese caso se incluirá *'Horas totales: En día de tipo [no laboral/festivo/vacaciones] se han detectado horas trabajadas'*
- El último paso es actualizar la tabla de histórico por si se han producido modificaciones en el total de horas trabajadas mediante el procedimiento almacenado updateSigningsHistory (explicado en el apartado anterior).

A continuación, se da una breve explicación sobre algunos de los procedimientos adicionales empleados.

- **ScheduleResume**

Recibe por parámetro el id de un horario y un entero que indica el día de la semana, en base a eso calcula y devuelve el total que se deben trabajar ese día y un resumen de las horas indicadas en el horario.

Ejemplo del formato del resumen: “Lunes: 9:00 – 14:00 – 15:00 – 18:00”

- **checkPuntuality**

Este procedimiento itera sobre los fichajes de la jornada para el usuario indicado y lo va comprando con el horario establecido con la ayuda del procedimiento auxiliar `locateSchedule`. Si detecta una irregularidad, teniendo en cuenta el margen de puntualidad explicado anteriormente, añadirá a las observaciones el siguiente mensaje: “Horario(XX:XX) - fichaje(YY:YY)”, además indicará que se ha detectado una incidencia por puntualidad.

- **locateSchedule**

Se trata de un procedimiento auxiliar, necesario porque MariaDB no permite incluir un bucle dentro de otro. Recibe por parámetro un tipo de horario, el nº de hora que se está buscando, el día de la semana en formato numérico y una variable ‘observaciones’ que debe completar si detecta alguna irregularidad y devolver.

El procedimiento itera sobre las horas del horario hasta llegar a la indicada que se está buscando, es decir, si el horario tiene 4 horas establecidas: 9:00 – 14:00 – 15:00 – 18:00 y se le pide la 3ª hora, el procedimiento devolver 15:00.

5.6 INTERFAZ WEB

Para el diseño de la interfaz se ha optado por un diseño sencillo y limpio que dé prioridad a los datos mostrados y facilite su lectura. Tal y como se indica en el esquema de distribución (imagen 49), en la parte superior se ha ubicado un banner con el nombre de la aplicación. Justo debajo se encuentra el menú básico de consulta. Este menú es común a ambos perfiles y permite consultar datos relativos al usuario identificado: fichajes, incidencias, vacaciones... En el caso de los administrados justo debajo de menú aparecerá

otro con opciones similares a las del menú superior, pero que permiten crear, editar y consultar los datos de todos los usuarios.



Imagen 48 - Estructura web

A continuación, se hace un repaso por las pantallas más relevantes de la aplicación:

- **Pantallas compartidas por ambos perfiles**
 - **Login**



Imagen 49 - Pantalla de login

- **Resumen principal:** en esta pantalla se puede ver un resumen del balance de horas del usuario, sus últimos fichajes y su horario para el día actual.

Cronos: Control de asistencia laboral.

Mi resumen
Mis fichajes
Mis incidencias
Mis vacaciones
Mi horario
Festivos
Jorge ▾

Balance de horas: -2 horas y 7 min

Total de horas que el trabajador debe a la empresa (balance < 0) o tiene acumuladas debido a horas extra (balance > 0).
El balance esta calculado en base a todas las incidencias con resolución 'Balance', el tiempo aportado por cada incidencia es la diferencia entre las horas fichadas menos las horas indicadas en el horario.

Mis últimos fichajes

Los fichajes sin jornada aún no han sido procesados.

Jornada	Fichaje	Válido	Dispositivo
Sin asignar	09:06	-	NIP1
Sin asignar	14:01	-	NIP1
Sin asignar	15:02	-	NIP1
Sin asignar	17:02	-	NIP1
28/07/2023	09:00	Sí	NIP1
28/07/2023	14:00	Sí	NIP1
28/07/2023	14:59	Sí	NIP1
28/07/2023	16:59	Sí	NIP1

Mi horario para hoy

Tipo de horario asignado: Jornada partida

Día	Hora
Martes	09:00
Martes	14:00
Martes	15:00
Martes	18:00

- **Consulta fichajes propios:** lista de últimos fichajes sin procesar (sin asignar a jornada) y una lista de los últimos fichajes procesados.

Cronos: Control de asistencia laboral.

Mi resumen
Mis fichajes
Mis incidencias
Mis vacaciones
Mi horario
Festivos
Sandra ▾

Resumen
Fichajes
Incidencias
Usuarios
Vacaciones
Tarjetas
Dispositivos
Horarios
Festivos

Mis fichajes sin procesar

Los fichajes sin procesar son aquellos que aún no han sido asignados a una jornada y no se han verificado con el horario.

Fichaje	Dispositivo
28/07/2023 08:59	NIP1

Mis fichajes procesados

Los fichajes procesados ya han sido asignados a una jornada y comprobados. En el caso de encontrar discrepancias relevantes con el horario se habrá generado una incidencia que se puede consultar en la sección correspondiente.

Jornada	Fichaje	Válido	Tipo	Dispositivo
24/07/2023	24/07/2023 09:00	Sí	Auto	NIP1
24/07/2023	24/07/2023 09:02	Sí	Auto	NIP1
24/07/2023	24/07/2023 13:58	Sí	Auto	NIP1
24/07/2023	24/07/2023 15:00	Sí	Auto	NIP1
24/07/2023	24/07/2023 17:59	Sí	Auto	NIP1
21/07/2023	21/07/2023 08:57	Sí	Auto	NIP1
21/07/2023	21/07/2023 13:56	Sí	Auto	NIP1
21/07/2023	21/07/2023 14:56	Sí	Auto	NIP1
21/07/2023	21/07/2023 16:57	Sí	Auto	NIP1

Imagen 50 – Pantalla consulta de fichajes propios

- **Consulta detalle incidencias:** en el detalle se muestran los datos básicos de la incidencia como fecha, tipo, resolución y las observaciones. En estas últimas vienen especificadas todas las irregularidades detectadas. Además, se incluye un listado con los fichajes de esa jornada y otro con el horario.

Cronos: Control de asistencia laboral.

[Mi resumen](#)
[Mis fichajes](#)
[Mis incidencias](#)
[Mis vacaciones](#)
[Mi horario](#)
[Festivos](#)
Jorge ▾

VOLVER

Incidencia:
 Las incidencias son generadas por discrepancias entre el horario asignado y los fichajes realizados. Pueden deberse a falta de puntualidad, realización de horas extra, ausencias...

Nº	Fecha	Tipo	Resolución
3	04/09/2023	Puntualidad y Horas totales	Balance

Discrepancias detectadas:
 Puntualidad: Horario(09:00:00) - fichaje(09:06:38). Horario(18:00:00) - fichaje(17:00:49). Se han hecho menos horas (6 horas y 54 min) de la indicadas en el horario(8 horas y 0 min)

Fichajes de esa jornada

Jornada	Fichaje	Válido	Tipo	Dispositivo
26/07/2023	26/07/2023 09:06	Sí	Auto	N1P1
26/07/2023	26/07/2023 13:59	Sí	Auto	N1P1
26/07/2023	26/07/2023 15:00	Sí	Auto	N1P1
26/07/2023	26/07/2023 17:00	Sí	Auto	N1P1

**Horario de esa jornada:
Jornada partida**

Día	Hora
Viernes	09:00
Viernes	14:00
Viernes	15:00

Imagen 51 - Pantalla consulta incidencias propias

- **Resumen administrador,** incluye los últimos fichajes realizados (procesados y sin procesar) y una lista de las incidencias pendientes de resolver.

Cronos: Control de asistencia laboral.

[Mi resumen](#)
[Mis fichajes](#)
[Mis incidencias](#)
[Mis vacaciones](#)
[Mi horario](#)
[Festivos](#)
Sandra ▾

Resumen
[Fichajes](#)
[Incidencias](#)
[Usuarios](#)
[Vacaciones](#)
[Tarjetas](#)
[Dispositivos](#)
[Horarios](#)
[Festivos](#)

Mis últimos fichajes
 Los fichajes sin jornada aún no han sido procesados.

Usuario	Jornada	Fichaje	Válido	Dispositivo
sandra	Sin asignar	08:59	-	N1P1
jorge	Sin asignar	09:01	-	N1P1
jorge	24/07/2023	08:59	Sí	N1P1
jorge	24/07/2023	09:00	Sí	N1P1
sandra	24/07/2023	09:00	Sí	N1P1
sandra	24/07/2023	09:02	Sí	N1P1
sandra	24/07/2023	13:58	Sí	N1P1
jorge	24/07/2023	13:59	Sí	N1P1
sandra	24/07/2023	15:00	Sí	N1P1
jorge	24/07/2023	15:01	Sí	N1P1

Incidencias pendientes
 Incidencias pendientes de resolución.

Nº	Fecha	Usuario	Tipo	
2	02/09/2023	jorge	Puntualidad y Horas totales	Resolver
1	02/09/2023	sandra	Puntualidad y Horas totales	Resolver

Imagen 52 - Pantalla resumen administrador

- **Consulta usuarios**, listado de los usuarios de la web



Imagen 53 - Pantalla de consulta de usuarios

- **Nuevo usuario**: Pantalla con los datos básicos de alta para un nuevo usuario.

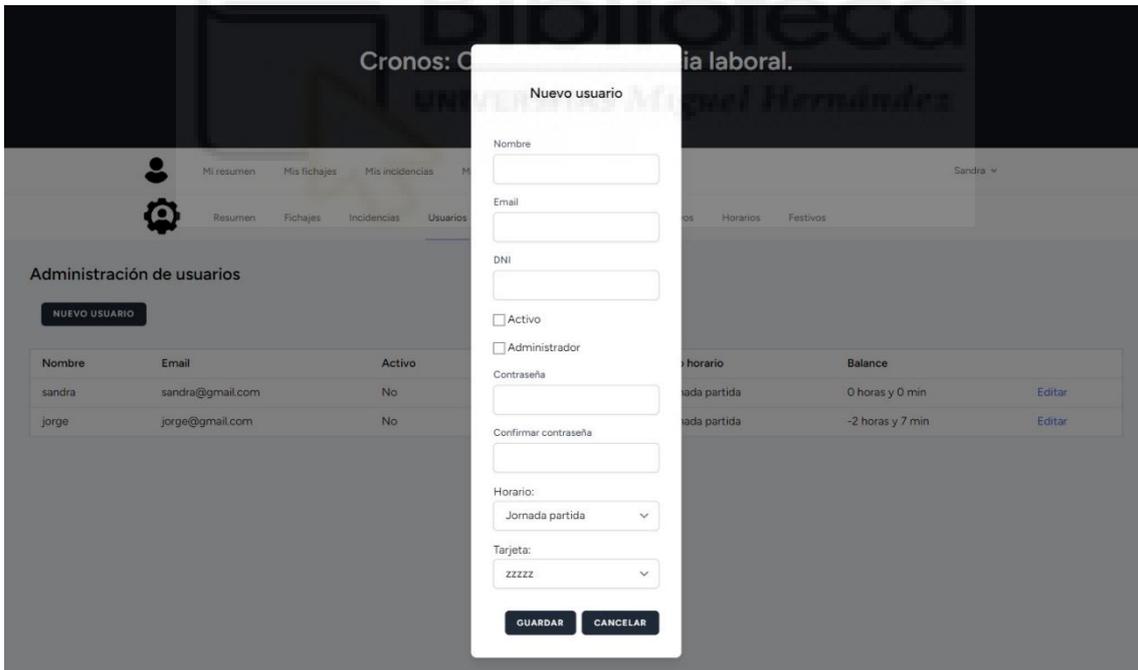


Imagen 54- Pantalla alta nuevo usuario

- **Consulta de horarios:** Listado de los horarios creados

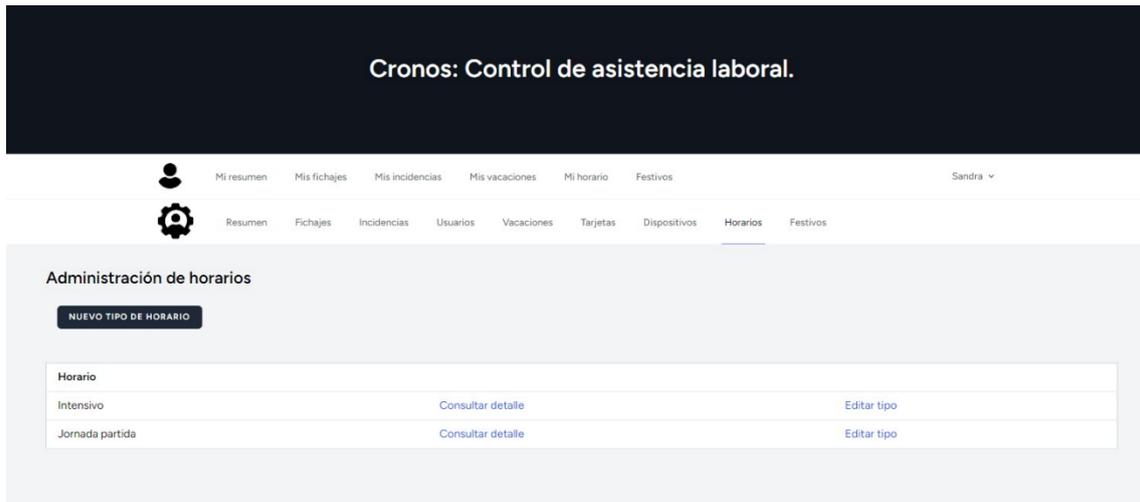


Imagen 55 - Pantalla consulta tipos de horarios

- **Consultar detalle de un horario:** Listado de las horas que componen un horario, se puede editar y borrar desde aquí.

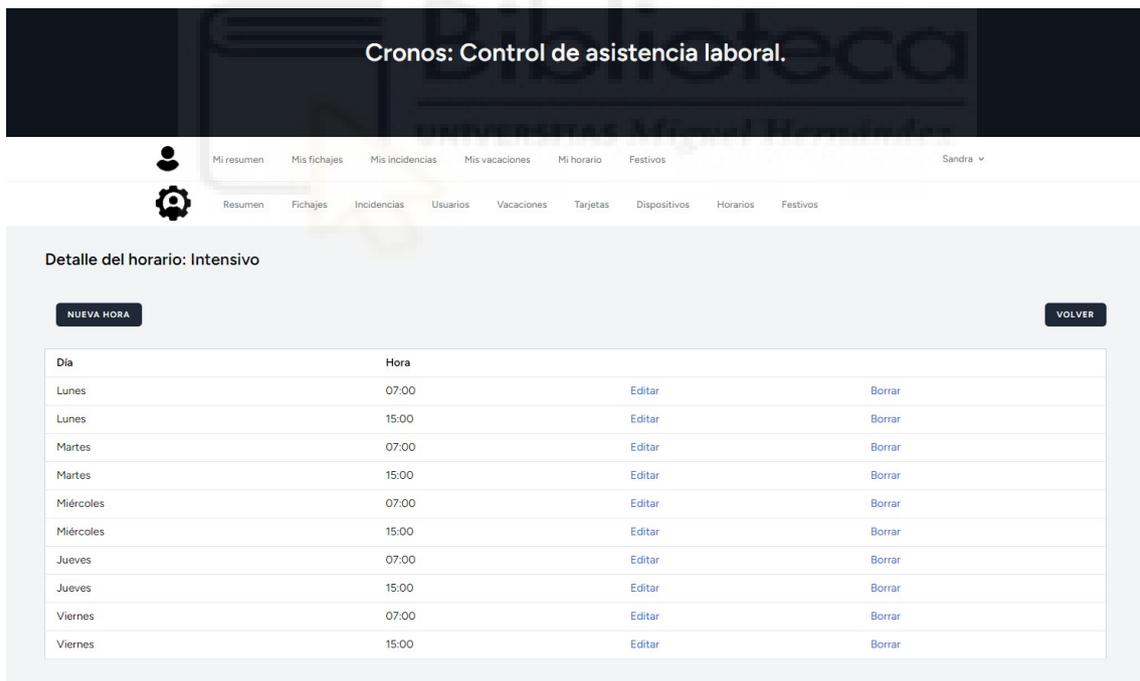


Imagen 56 - Pantalla consulta el detalle de un horario

- **Consulta de incidencias:** En la parte superior para que se más accesible se listan las incidencias pendientes de resolver. En parte inferior se encuentra la lista completa de las incidencias procesadas.

Cronos: Control de asistencia laboral.

Mi resumen

Mis fichajes
Mis incidencias
Mis vacaciones
Mi horario
Festivos

Sandra ▾

Resumen

Fichajes
Incidencias
Usuarios
Vacaciones
Tarjetas
Dispositivos
Horarios
Festivos

Incidencias pendientes

ID	Fecha	Usuario	Tipo	
1	03/09/2023	sandra	Puntualidad y Horas totales	Resolver

Incidencias resueltas

ID	Fecha	Usuario	Tipo	Resolucion	
2	03/09/2023	jorge	Puntualidad y Horas totales	Balance	Consultar

Imagen 57 - Pantalla consulta incidencias



6 CONCLUSIONES Y PROPUESTAS

6.1 CONCLUSIONES

En términos generales se puede afirmar que los objetivos fijados al inicio de este proyecto se han cumplido, el resultado final es un producto que ofrece a pequeñas y medianas empresas una solución asequible y funcional para afrontar la necesidad legal del control de asistencia. Se compone de 3 partes diferenciadas, que emplean lenguajes y tecnologías distintas y que trabajan juntas:

- Una web con una interfaz intuitiva que permite la gestión y consulta de los datos para los distintos perfiles de trabajadores tanto en formato web como móvil gracias a su diseño responsive.
- Un dispositivo asequible, de fácil instalación y uso para la recogida de datos mediante tecnología RFID.
- Una API Rest que gestiona las comunicaciones del dispositivo con el servidor web y se encarga del procesado de los datos recogido.

Para su realización se han puesto en práctica los conocimientos adquiridos en las distintas asignaturas cursadas a lo largo de la carrera como Diseño de aplicaciones en internet, Física, Bases de datos, Gestión de proyecto o Interfaces de usuario entre otras. Además, el seleccionar para su realización tecnologías no conocidas hasta el momento como Arduino o Laravel me ha forzado a ser autodidacta y a adquirir habilidades y conocimientos que sin duda me serán útiles en el futuro.

6.2 PROPUESTAS

Mejoras para la web:

- Mejorar la resolución de incidencias: permitir editar como válido o no un fichaje desde la interfaz, añadir fichajes manuales si falta alguno...
- Incluir **estadísticas detalladas** sobre la puntualidad de los trabajadores en general o individualmente.
- Añadir más variedad de horarios, actualmente solo se permite un tipo de horario por trabajador. Una posible mejora es añadir un sistema de asignación de horarios

variables o un sistema de horarios programados por turnos para cubrir las necesidades de aquellas empresas donde los empleados trabajan con turnos rotativos.

- Chat interno
- Subida de documento como justificantes de ausencia

Mejoras para el dispositivo:

- **Actualización red y contraseña wifi:** ahora mismo el nombre de la red y contraseña wifi que emplea el dispositivo esta especificada directamente en el código. Es poco práctico que se deba recompilar el código para cambiar la red o la contraseña. La placa elegida cuenta con una antena wifi integrada, es posible añadir un botón para que, al iniciar el sistema, entre en modo servidor web. Esto permitiría incluir en el propio Arduino una web sencilla para actualizar los parámetros para el nombre de la red y la contraseña.
- **Cambiar el método de fichajes por algún sensor de tipo biométrico** como el reconocimiento facial o la huella dactilar. Esto evitaría que los trabajadores pudieran fichar por los compañeros simplemente prestando la tarjeta identificativa.

7 BIBLIOGRAFÍA

Leves y decretos

Real Decreto-ley 8/2019, de 8 de marzo, de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo. (BOE núm. 61, 12 de marzo de 2019)

<https://www.boe.es/buscar/doc.php?id=BOE-A-2019-3481>

Páginas web

Documentación Laravel (enero, 2023) – Version 10.x

<https://laravel.com/docs/10.x>

Documentación Laravel (enero, 2023) – Bootcamp: Learn the PHP Framework for Web Artisans (Version Blade)

<https://bootcamp.laravel.com/>

Documentación PHP (enero, 2023) – Manual PHP

<https://www.php.net/manual/es/>

Documentación MariaDB (febrero, 2023) – Programmatic & Compound Statements

<https://mariadb.com/kb/en/programmatic-compound-statements/>

Documentación MySQL (mayo, 2023) – Bucles y cursores en MySQL con ejemplos

<https://poesiabinaria.net/2013/12/bucles-y-cursos-en-mysql-con-ejemplos/>

Documentación MySQL (mayo, 2023) – MySQL Stored Procedure Variables

<https://www.mysqltutorial.org/variables-in-stored-procedures.aspx>

Documentación Taildwind (febrero, 2023) – Taildwind CSS Modal – Flowbite

<https://flowbite.com/docs/components/modal/>

Diagramas UML (2023, agosto) - Diagrama de clases

<https://diagramasuml.com/diagrama-de-clases/#Clases>

Recursos – Descarga de iconos

<https://www.flaticon.es/>

Fichas técnicas

Placa Arduino MK1000

<https://tienda.bricogeek.com/arduino/1028-arduino-mkr1000-wifi.html>

Módulo RFID – MFRC522

<https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

