

# **TimeCheck:**

## **Software para el control horario en empresas de construcción**

Trabajo de fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



# **VNiVERSiDAD D SALAMANCA**

Mayo de 2023

**Autor**

Rodrigo de la Calle Alonso

**Tutor/a**

Álvaro Lozano Murciego

## **Certificado del tutor**

D. Álvaro Lozano Murciego, profesor del Departamento de Informática y Automática de la Universidad de Salamanca.

HACE CONSTAR

Que el trabajo titulado “TimeCheck: Software para el control horario en empresas de construcción”, que se presenta, ha sido realizado por D. Rodrigo de la Calle Alonso con DNI 49839013Z y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado de Ingeniería Informática en esta Universidad.

Salamanca, a 5 de julio de 2023

Fdo.: \_\_\_\_\_

## Resumen

Desde 2019, el registro de la jornada laboral en España ha sido obligatorio y regido por el Real Decreto-ley 8/2019 para todo tipo de empresas. Cuando este problema se traslada al sector de la construcción nos encontramos que gran parte de los sistemas actuales que intentan resolver este problema no se adaptan a las particularidades del sector, debido a la gran movilidad a la que están expuestos los obreros y el entorno en el que trabajan.

Hasta ahora, este control se llevaba de forma manual mediante hojas de firmas. Posteriormente, nuevas alternativas han surgido como sistemas de identificación mediante identificaciones RFID, sistemas biométricos o, lo más reciente, geolocalización mediante aplicaciones móviles. Ninguna de ellas se adapta completamente a los requisitos que presenta un sector como el de la construcción.

Por ello, el presente Trabajo Fin de Grado persigue la creación de un sistema de control horario cuyo fin es responder a esta necesidad mediante un software que pueda facilitar este proceso, que sea flexible con un entorno tan variable y una ley nueva, susceptible a reforma y que a su vez favorezca a la digitalización de las empresas, la cual es una meta de la Unión Europea para la agenda 2030. Todo esto se llevará a cabo empleando las últimas tecnologías disponibles y además enfocándolo a que tenga un coste mínimo para que sea más atractivo para las empresas que lo necesiten.

El sistema presenta diferentes roles de usuario, cada uno con una responsabilidad diferente, siendo el rol de administrador el encargado de gestionar el software en cada empresa y el resto de los roles son los correspondientes a los empleados. Esta solución software se ha complementado con otra parte hardware para adaptarse a las diferentes formas de fichar para cada uno de los roles del sistema. Esto permite al sistema adaptarse a las necesidades del entorno de trabajo.

En este documento se van a exponer los distintos requisitos que se deben de respetar para que el resultado se ajuste al entorno real y no solo a un marco teórico. Se detallan los casos de uso, la planificación y gestión del proyecto, las tecnologías implicadas y las alternativas que se han valorado para llevarlo a cabo. Finalmente se mostrará el resultado del desarrollo, los aspectos relevantes y las conclusiones extraídas tras desarrollar este Trabajo Fin de Grado.

**Palabras clave:** PWA, sector de la construcción, control horario, IoT, Firebase.

## **Abstract**

Since 2019, the registration of working hours in Spain has been mandatory and governed by the Royal Decree-Law 8/2019 for all types of companies. When this issue is applied to the construction industry, it is found that a large part of the current systems attempting to solve this problem do not adapt well to the specificities of the sector, due to the high mobility to which the workers are exposed and the environment in which they work.

Until now, this control was carried out manually using signature sheets. Subsequently, new alternatives have emerged, such as identification systems using RFID tags, biometric systems, or most recently, geolocation through mobile applications. However, none of them fully meet the requirements presented by a sector like construction.

Therefore, the present Bachelor's Thesis aims to create a timekeeping system whose purpose is to address this need through software that can facilitate this process. It should be flexible enough to accommodate the variable environment and the ever-changing law, which is susceptible to reform, while also promoting the digitalization of companies, a goal set by the European Union for the 2030 agenda. All of this will be carried out by employing the latest available technologies and ensuring minimal costs to make it more attractive to the companies that need it.

The system includes different user roles, each with different responsibilities. The administrator role is responsible for managing the software in each company, and the rest of the roles are assigned to employees. This software solution is complemented by a hardware component to adapt to the various ways of clocking in for each role in the system. This allows the system to adapt to the specific needs of the working environment.

In this document, the different requirements that must be respected to align the result with the real environment, rather than just a theoretical framework, will be exposed. The use cases, project planning, management, the technologies involved, and the evaluated alternatives to carry it out will be detailed. Finally, the development's outcome, relevant aspects, and conclusions drawn from conducting this Bachelor's Thesis will be presented.

**Key Words:** PWA, construction industry, working attendance, IoT, Firebase.

# Tabla de contenido

<b>1 INTRODUCCIÓN .....</b>	<b>1</b>
1.1 PRESENTACIÓN DE LA TEMÁTICA DEL PROYECTO .....	1
1.2 ESTRUCTURA DE LA MEMORIA .....	3
<b>2 OBJETIVOS DEL PROYECTO.....</b>	<b>4</b>
2.1 OBJETIVOS DEL SOFTWARE .....	4
2.2 OBJETIVOS PERSONALES .....	4
<b>3 CONCEPTOS TEÓRICOS.....</b>	<b>6</b>
3.1 CONTROL HORARIO .....	6
3.2 SISTEMA DE CONTROL HORARIO .....	7
3.3 LEY DE MEDIDAS URGENTES DE PROTECCIÓN SOCIAL Y DE LUCHA CONTRA LA PRECARIEDAD LABORAL EN LA JORNADA DE TRABAJO. ....	9
3.4 CUADRILLAS .....	10
3.5 PROGRESSIVE WEB APP (PWA).....	11
3.6 ARQUITECTURA <i>SERVERLESS</i> .....	12
<b>4 TÉCNICAS Y HERRAMIENTAS.....</b>	<b>13</b>
4.1 VUEJS.....	13
4.2 FRAMEWORK UI PRIMEVUE.....	13
4.3 NODEJS.....	13
4.4 BIBLIOTECAS.....	14
4.5 FIREBASE .....	16
4.6 ARDUINO.....	16
4.7 HERRAMIENTAS DE DISEÑO.....	17
4.8 HERRAMIENTAS CASE .....	17
<b>5 ASPECTOS RELEVANTES DEL DESARROLLO.....</b>	<b>18</b>
5.1 CICLO DE VIDA .....	18
5.2 ESPECIFICACIÓN DE REQUISITOS .....	20
5.3 ESPECIFICACIÓN DE ANÁLISIS .....	22
5.4 ESPECIFICACIÓN DE DISEÑO .....	24
5.4.1 <i>Patrones Arquitectónicos</i> .....	24
5.4.2 <i>Modelo del dominio</i> .....	27
5.4.3 <i>Base de datos</i> .....	29
5.4.4 <i>Interfaz de usuario</i> .....	30
5.4.5 <i>Pruebas</i> .....	31
5.5 IMPLEMENTACIÓN .....	32
5.5.1 <i>Despliegue</i> .....	32
5.5.2 <i>Base de datos</i> .....	33
5.6 HARDWARE .....	35
5.7 PRESENTACIÓN DE LA APLICACIÓN .....	37
5.7.1 <i>Interfaces del administrador</i> .....	40
5.7.1 <i>Interfaces del empleado</i> .....	45
<b>6 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS .....</b>	<b>47</b>
6.1 CONCLUSIONES .....	47
6.2 LÍNEAS DE TRABAJO FUTURAS .....	48
<b>REFERENCIAS.....</b>	<b>49</b>

## Tabla de Ilustraciones

ILUSTRACIÓN 1 CONTROL MEDIANTE RFID.....	7
ILUSTRACIÓN 2 EJEMPLO REGISTRO HORARIO EXCEL.....	9
ILUSTRACIÓN 3 ESTRUCTURA DE UNA OBRA.....	10
ILUSTRACIÓN 4 FULLCALENDAR [24].....	14
ILUSTRACIÓN 5 APEXCHARTS [26].....	15
ILUSTRACIÓN 6 FASES PROCESO UNIFICADO [46].....	18
ILUSTRACIÓN 7 CICLO DE VIDA MICROSOFT PROJECT.....	19
ILUSTRACIÓN 8 PAQUETE GESTIÓN DE USUARIOS.....	20
ILUSTRACIÓN 9 PAQUETE EMPRESA.....	21
ILUSTRACIÓN 10 PAQUETE EMPLEADO.....	21
ILUSTRACIÓN 11 DIAGRAMA DE CLASES.....	22
ILUSTRACIÓN 12 DIAGRAMA DE SECUENCIA.....	23
ILUSTRACIÓN 13 PATRÓN MVVM [48].....	24
ILUSTRACIÓN 14 DISEÑO, ARQUITECTURA.....	25
ILUSTRACIÓN 15 EJEMPLO DE DIAGRAMA DE SECUENCIA DE DISEÑO.....	26
ILUSTRACIÓN 16 DISEÑO, DIAGRAMA DE CLASES.....	27
ILUSTRACIÓN 17 DISEÑO, BASE DE DATOS.....	29
ILUSTRACIÓN 18 DISEÑO, WIREFRAME OBRERO.....	30
ILUSTRACIÓN 19 DISEÑO, WIREFRAME RESULTADOS.....	30
ILUSTRACIÓN 20 DIAGRAMA DE DESPLIEGUE.....	32
ILUSTRACIÓN 21 FIRESTORE DATABASE.....	33
ILUSTRACIÓN 22 REALTIME DATABASE.....	34
ILUSTRACIÓN 23 M5STICK CPLUS.....	35
ILUSTRACIÓN 24 CONFIGURACIÓN M5.....	36
ILUSTRACIÓN 25 VALIDACIÓN MEDIANTE M5.....	36
ILUSTRACIÓN 26 REGISTRADOR DE TIEMPO ANALÓGICO [50].....	37
ILUSTRACIÓN 27 LOGO OSCURO.....	37
ILUSTRACIÓN 28 LOGO CLARO.....	37
ILUSTRACIÓN 29 REGISTRARSE.....	38
ILUSTRACIÓN 30 INICIAR SESIÓN.....	38
ILUSTRACIÓN 31 VERIFICAR EMAIL.....	39
ILUSTRACIÓN 32 MENÚ CONFIGURACIÓN.....	39
ILUSTRACIÓN 33 VENTANA INICIO ADMINISTRADOR.....	40
ILUSTRACIÓN 34 EMPLEADOS ADMINISTRADOR (NOMBRES FICTICIOS).....	41
ILUSTRACIÓN 35 OBRAS ADMINISTRADOR.....	41
ILUSTRACIÓN 36 EVENTO EMPLEADO.....	42
ILUSTRACIÓN 37 EVENTO UBICACIÓN.....	42
ILUSTRACIÓN 38 CALENDARIO DE TRABAJO.....	43
ILUSTRACIÓN 39 RESULTADOS.....	44
ILUSTRACIÓN 40 AVISOS.....	44
ILUSTRACIÓN 41 ADOBE COLOR.....	45
ILUSTRACIÓN 42 OFICINA.....	46
ILUSTRACIÓN 43 CAPATAZ.....	46

# 1 Introducción

## 1.1 Presentación de la temática del proyecto

Desde 2019, el registro de la jornada laboral en España ha sido obligatorio y está regulado por el Real Decreto-ley 8/2019 [1] para todo tipo de empresas. Sin embargo, cuando este requisito se aplica al sector de la construcción, se presentan desafíos particulares debido a la gran movilidad de los obreros y al entorno en el que trabajan.

Hasta ahora, el control horario se llevaba a cabo de forma manual mediante hojas de firmas. Con el tiempo, se han implementado sistemas de identificación utilizando tecnologías como las identificaciones RFID [2], sistemas biométricos [3] e incluso geolocalización a través de aplicaciones móviles [4]. Sin embargo, ninguno de estos métodos se ajusta perfectamente al sector de la construcción necesitando, en algunos casos, soluciones ad-hoc para este sector y sus flujos de trabajo.

Por este motivo, se presenta el desarrollo de un nuevo sistema cuyo objetivo es dar respuesta a esta necesidad a través de un software que facilite el proceso de control horario. Este sistema debe de ser flexible para adaptarse a un entorno tan variable como el de la construcción y cumplir con la normativa vigente.

Además, este proyecto busca aplicar la digitalización a las empresas de este sector, lo cual se alinea con la meta establecida por la Unión Europea para la agenda 2030 [5]. Para alcanzar los Objetivos de Desarrollo Sostenible (ODS) [6], es fundamental armonizar tres elementos básicos: el crecimiento económico, la inclusión social y la protección del medio ambiente. Este proyecto colabora a la consecución del ODS 9 “Industria, innovación e Infraestructuras” y en concreto su meta 9.4:

*“...modernizar la infraestructura y **reconvertir las industrias para que sean sostenibles, utilizando los recursos con mayor eficacia y promoviendo la adopción de tecnologías y procesos industriales limpios y ambientalmente racionales...**”*

Este proyecto evidentemente es beneficioso para cualquier empresa que lo pueda llegar a aplicar ya que evita revisar miles de hojas de asistencia que hasta ahora tenían que ser creadas y leídas por un administrativo lo cual le impide atender otras tareas primordiales de su oficio donde “el tiempo es dinero”. Usar esta tecnología puede repercutir en un mayor rendimiento y condiciones laborales lo cual genera un crecimiento económico notable. La herramienta contribuye por tanto a la digitalización de la empresa.

Cabe destacar que este software es una forma de que todos los empleados sean tratados de forma igualitaria reflejando la situación real y objetiva de cada uno de ellos evitando así cualquier tipo de discriminación personal o laboral.

Por último, la adopción de este proceso supone eliminar las grandes cantidades de papel y tinta que se requieren actualmente para imprimir las hojas de asistencia y lo necesario para almacenar esta información. Esto contribuye a reducir el uso de papel, plásticos y químicos con los que están fabricados los tóner y cartuchos de tinta de las impresoras. El impacto limitado a una única empresa puede ser anecdótico, pero si la aplicación tuviera una gran adopción el impacto en reducción de estos materiales sería relevante.

## 1 Introducción

El sistema puesto presenta diferentes roles de usuario, cada uno con responsabilidades concretas. Por un lado, el rol de administrador será el encargado de gestionar el software en cada empresa, mientras que los demás roles corresponderán a los empleados. Además de la solución *software*, sistema dispone de una parte *hardware*, ya que la forma de fichar puede variar según los diferentes roles y el sistema busca adaptarse a las necesidades del entorno de trabajo de cada empleado.

Este Trabajo de Fin de Grado tiene como objetivo proporcionar una solución efectiva y adaptada a las necesidades del control horario y gestión laboral en el sector de la construcción, promoviendo el cumplimiento de la normativa legal vigente y facilitando la digitalización de las empresas del sector.

Como motivación personal, este trabajo pretende aplicar los conocimientos adquiridos durante el Grado en Ingeniería Informática, para dar como resultado un producto real que además solucione la necesidad surgida en una empresa familiar, la cual no solo confía en mí para ello si no que colabora con las pruebas de este sistema en un entorno real, haciendo de este trabajo, un producto útil y no solo un prototipo.

Para perseguir esta idea, este proyecto ha sido presentado a la “*Convocatoria De Prototipos Orientados Al Mercado En TCUE*” [7] del curso 2022/2023 siendo aceptado y por lo tanto convirtiéndome en beneficiario de esta propuesta. La formación de este curso a aportado valor y un punto de vista empresarial a este trabajo.

En este documento se exponen los distintos requisitos que debe de cumplir el sistema para ajustarse a un entorno real de explotación y no solo a un marco teórico. Se describirán los casos de uso del sistema, así como la planificación y gestión del proyecto. También se mencionarán las tecnologías involucradas y las alternativas que se han considerado para cada caso. Por último, se presentará el resultado de la investigación y el desarrollo final del sistema.



## 1.2 Estructura de la memoria

La estructura de esta memoria se ha organizado de la siguiente manera:

- **Introducción:** se presenta una breve introducción sobre el tema del proyecto y se explican las secciones que componen la memoria.
- **Objetivos del proyecto:** se definen y explican tanto los objetivos del sistema software como los objetivos personales del autor.
- **Conceptos teóricos:** se detallan de forma concisa los aspectos teóricos principales relacionados con el dominio de aplicación del sistema y que son necesarios para comprender el contexto del proyecto.
- **Técnicas y herramientas:** se describen las técnicas y herramientas utilizadas durante el desarrollo del proyecto.
- **Aspectos destacados del desarrollo del proyecto:** se explican las etapas más relevantes del proceso de desarrollo del proyecto, incluyendo el ciclo de vida, la especificación de requisitos, el análisis, el diseño, la implementación y las pruebas realizadas.
- **Conclusiones y futuras líneas de trabajo:** se analizan y discuten los resultados finales, y se presentan posibles mejoras y áreas de desarrollo futuro.

Además de la memoria, se adjuntan cinco anexos técnicos que detallan el proyecto:

- **Anexo I, Planificación temporal:** contiene el cronograma detallado que se seguirá a lo largo del tiempo para llevar a cabo el proyecto.
- **Anexo II, Especificación de requisitos del software:** incluye la definición detallada de los requisitos del proyecto.
- **Anexo III, Especificación de Análisis y Diseño:** proporciona el análisis y el diseño detallado del proyecto. También se realizan pruebas de aceptación después de la implementación de cada iteración.
- **Anexo IV, Documentación técnica para programadores:** ofrece una descripción de los archivos que componen el proyecto dirigido al equipo de desarrollo que tuviera que continuar con el desarrollo del proyecto.
- **Anexo V, Manual de usuario:** proporciona una guía de instalación y uso de la aplicación para los usuarios finales del sistema.

## 2 Objetivos del proyecto

En este apartado se indican y detallan los diferentes objetivos del proyecto.

### 2.1 Objetivos del software

El objetivo principal del proyecto es el desarrollo de una aplicación web progresiva para el control horario y de la jornada laboral de los trabajadores principalmente orientado a la industria de la construcción permitiendo devolver informes que permitan cumplir el “*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*”.

Para cumplir este objetivo general, se plantean una serie de objetivos específicos:

- Diseño y desarrollo de una aplicación web progresiva donde el administrador de la empresa podrá:
  1. Registrar a sus empleados y adjudicar sus roles para que estos fichan en las entradas y salidas a lo largo de su jornada.
  2. Registrar las obras en las que se van a trabajar.
  3. Consultar el estado de la empresa en tiempo real.
  4. Obtener resultados agrupados y filtrados por distintos parámetros como obras, cuadrillas u oficinas.
- Diseño y desarrollo de una aplicación web progresiva donde los empleados de las empresas puedan fichar con su móvil tanto para entrar como para salir en función de su rol y su lugar de trabajo.
- Elección, configuración y programación de un dispositivo que permita fichar a empleados que trabajen en sitios predefinidos como oficinas o almacenes.

### 2.2 Objetivos personales

Este proyecto tiene una gran motivación personal ya que está inicialmente destinado a apoyar a una empresa familiar. Un servicio de este tipo les sería de gran utilidad y han confiado en mí para encontrar una solución. Se busca no solo cubrir esta necesidad si no también aprovechar la oportunidad para avanzar en el incipiente proceso de digitalización que a su vez aportará ventajas como ahorro de tiempo, recursos y costes de la gestión de estos datos.

Este es un proyecto que llevo tiempo queriendo llevar a cabo, pero necesitaba alcanzar los conocimientos suficientes para ello, además de tener tiempo para dedicárselo. Por ello, realizarlo como Trabajo de Fin de Grado es una buena oportunidad ya que además contaré con el apoyo de mi tutor de TFG para las posibles dudas que me pudieran surgir.

Además de lo anterior, se presentan una serie de objetivos técnicos personales:

- Familiarizarse con el *framework* VueJS, sus lenguajes como JavaScript, y patrones de diseño que se deben de utilizar para el desarrollo de la aplicación.
- Conocer cómo funciona una base de datos no relacional en este caso Firestore.
- Aprender a utilizar la suite de servicios de Firebase.

## 2 Objetivos del proyecto

- Profundizar en microcontroladores compatibles con Arduino y con capacidades para el Internet de las Cosas (IoT, del inglés, *Internet of Things*).
- Poner en práctica conceptos adquiridos en el Grado en Ingeniería Informática como el desarrollo de interfaces gráficas, la Interacción Persona-Ordenador, la Ingeniería del Software, la Programación Orientada a Objetos, el Diseño Bases de Datos, etc.

## 3 Conceptos teóricos

### 3.1 Control horario

El concepto "control horario" se refiere a la gestión y seguimiento de los tiempos de trabajo de los empleados dentro de una organización o empresa. Consiste en llevar un registro preciso de las horas de entrada y salida de los trabajadores, así como de los períodos de descanso y pausas durante la jornada laboral.

El control horario es fundamental para asegurar el cumplimiento de las normativas laborales y la protección de los derechos de los trabajadores, presentes en las leyes sobre el tiempo de trabajo y las horas extras, y asimismo para garantizar una gestión eficiente de los recursos humanos en una empresa.

Existen varias razones por las cuales las organizaciones implementan sistemas de control horario. Algunas de ellas incluyen:

- Imperativo legal: En muchos países, existen leyes y regulaciones que establecen límites sobre el número de horas de trabajo permitidas, descansos obligatorios, pago de horas extras, entre otros. El control horario ayuda a asegurar que la empresa cumpla con estas normativas y evite posibles sanciones legales.
- Gestión de recursos: Registrar y analizar los horarios de los empleados permite a las empresas tener un control efectivo de cómo se distribuye el tiempo de trabajo. Esto facilita la asignación de tareas, la planificación de proyectos y la optimización de los recursos disponibles.
- Pago justo y preciso: El control horario proporciona información precisa sobre las horas trabajadas por cada empleado, lo que facilita el cálculo de los salarios y evita errores o discrepancias en los pagos.
- Productividad y eficiencia: Al conocer el tiempo que los empleados dedican a diferentes tareas y proyectos, las empresas pueden identificar áreas de mejora, identificar posibles cuellos de botella y optimizar los procesos para aumentar la productividad y eficiencia laboral.

Para las empresas de construcción, el control horario es especialmente relevante debido a la naturaleza de su trabajo, que a menudo implica diferentes ubicaciones, proyectos y equipos de trabajo. Un software de control horario específico para empresas de construcción puede incluir características adicionales como el seguimiento de horas por proyecto, geolocalización para validar la ubicación de los empleados y funciones de registro de asistencia en obras o sitios remotos.

Es importante destacar que el control horario debe de llevarse a cabo de manera ética y respetando la privacidad de los empleados, cumpliendo con las leyes y regulaciones locales en materia de protección de datos y derechos laborales.

### 3.2 Sistema de control horario

Un sistema de control horario es una herramienta o conjunto de métodos y tecnologías utilizados para registrar y gestionar los horarios y asistencias de los empleados en una organización. Estos sistemas automatizan el proceso de control horario, brindando mayor precisión, eficiencia y facilidad de uso en comparación con los métodos manuales tradicionales.

A continuación, se describen las cuatro categorías más comunes de sistemas de control horario, cada una utilizando diferentes enfoques y tecnologías:

- Manual: este método implica que los empleados registren sus horarios manualmente, generalmente mediante hojas de asistencia en papel o formularios electrónicos. Los empleados escriben su hora de entrada y salida, y pueden incluir información adicional como la duración de las pausas. También puede haber un responsable a cargo de este control. Aunque este enfoque es simple, puede ser propenso a errores y fraudes, ya que depende de la honestidad y precisión de los empleados al registrar sus horarios, sin contar que estos datos tendrían que ser digitalizados posteriormente en algún tipo de software lo que implica un malgasto de tiempo y recursos en ello.
- Mediante tecnología RFID (del inglés, *Radio Frequency Identification*, en español, Identificación por Radiofrecuencia): en este sistema, los empleados llevan consigo tarjetas, etiquetas o llaveros que contienen chips RFID [2]. Los lectores de RFID instalados en las ubicaciones de entrada y salida registran automáticamente la presencia del empleado al detectar su tarjeta o etiqueta como se muestra en la Ilustración 1. Estos lectores pueden implicar una barrera física o no, como por ejemplo un tornillo. Esto proporciona un registro rápido y económico, sin embargo, también puede haber problemas de suplantación si alguien utiliza la identificación de otro empleado. Por último, tener en cuenta que es muy complicado instalar este tipo de sistemas en zonas en construcción sin contar que se encuentran expuestos a la intemperie y a habituales robos de materiales de obra.



Ilustración 1 Control Mediante RFID [8]

### 3 Conceptos teóricos

- Mediante sistemas biométricos: los sistemas de control horario biométricos [9] utilizan características únicas y distintivas del cuerpo humano para identificar y registrar la asistencia de los empleados. Esto puede incluir la lectura de huellas dactilares [3], reconocimiento facial, escaneo de iris o reconocimiento de voz [10]. Estos sistemas ofrecen un alto nivel de precisión y seguridad, además, no se requiere llevar una identificación adicional. Sin embargo, con este método, el dato utilizado como identificador, por ejemplo, la huella dactilar, debe de ser registrado anteriormente por la empresa por lo que supone una gestión de datos personales por parte de la empresa. A raíz de esto pueden surgir problemas relacionados con la privacidad y la protección de datos personales y confidenciales [11].
- Mediante aplicaciones móviles: con el auge de los teléfonos inteligentes, muchas empresas están optando por soluciones de control horario basadas en aplicaciones móviles [4]. Estas aplicaciones permiten a los empleados registrar sus horarios y asistencias utilizando sus dispositivos móviles, normalmente, a través de funciones como la geolocalización. Las aplicaciones móviles proporcionan flexibilidad y conveniencia, ya que los empleados pueden registrar su tiempo desde cualquier lugar. Sin embargo, nos encontramos nuevamente con los problemas sobre datos personales confidenciales además de tener en cuenta que la mayoría de los obreros, no llevan consigo sus dispositivos móviles debido a que se pueden dañar con facilidad por la naturaleza de su trabajo.

Cada categoría de sistema de control horario tiene sus propias ventajas e inconvenientes, por lo que es necesario evaluar cual se adapta plenamente a los requisitos del sector de la construcción o si es necesario proponer una nueva solución.

### 3.3 Ley de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo.

El “*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*” [1], fue una normativa implementada en España con el objetivo de regular y garantizar el control horario de los trabajadores, así como combatir la precariedad laboral.

Este decreto estableció “*La empresa garantizará el registro diario de jornada, que deberá incluir el horario concreto de inicio y finalización de la jornada de trabajo de cada persona trabajadora*”. A través de esta medida, se busca asegurar el cumplimiento de los límites legales de tiempo de trabajo, descansos y horas extras, así como fomentar una cultura de trabajo adecuada y combatir la precariedad laboral.

Estos registros pueden ser solicitados en cualquier momento por el ministerio de Trabajo, aunque se no especifica un método en particular para llevar a cabo el registro de la jornada laboral. Esto significa que las empresas tienen flexibilidad para elegir el sistema que mejor se adapte a sus necesidades y características. Lo que si se solicita obligatoriamente son: horas de entrada y salida y horas extra realizadas [12].

Sin embargo, es importante destacar que el decreto establece que el registro de la jornada debe de ser objetivo y fiable. Además, se debe de garantizar que exista una protección adecuada de los datos personales recopilados en el proceso. El formato de entrega puede ser en Word o Excel, pero siempre generados automáticamente para evitar así el fraude. En la Ilustración 2 se puede observar un ejemplo en Excel válido.

Nombre o referencia según relación personal	Fecha	Hora entrada	Inicio descanso	Fin descanso	Hora salida	Horas normales	Horas extras	Total horas trabajadas	Total Horas T. - Descanso	Total minutos trabajados
dato1	1/1/2020	8:00	10:00	10:20	16:00	7:30	0:30	8:00	7:40	480
dato 2	1/1/2020	8:30	10:00	10:20	18:00	7:30	2:00	9:30	9:10	570
dato 3	1/1/2020	8:00	10:00	10:20	14:00	6:00	0:00	6:00	5:40	360
dato 4	2/1/2020	8:00	10:00	10:20	15:00	7:00	0:00	7:00	6:40	420
dato5	3/1/2020	8:00	10:00	10:20	18:15	7:30	2:45	10:15	9:55	615
dato6	4/1/2020	8:00	10:00	10:20	17:00	7:30	1:30	9:00	8:40	540
dato7	12/1/2020	8:00	10:00	10:20	20:17	7:30	4:47	12:17	11:57	737
dato8	8/1/2020	8:00	10:00	10:20	18:30	7:30	3:00	10:30	10:10	630
dato9	3/1/2020	8:00	10:00	10:20	20:00	7:30	4:30	12:00	11:40	720
dato10	10/1/2020	8:30	10:00	10:20	18:00	7:30	2:00	9:30	9:10	570

Ilustración 2 Ejemplo Registro Horario Excel [12]

### 3.4 Cuadrillas

En el contexto de la construcción, el término "cuadrilla" se refiere a un grupo de trabajo conformado por un capataz y varios obreros que colaboran en la ejecución de tareas específicas en un proyecto de construcción como se refleja en la Ilustración 3. La cuadrilla es una unidad básica de organización en el sector de la construcción y desempeña un papel crucial en la coordinación y ejecución eficiente de las tareas diarias.

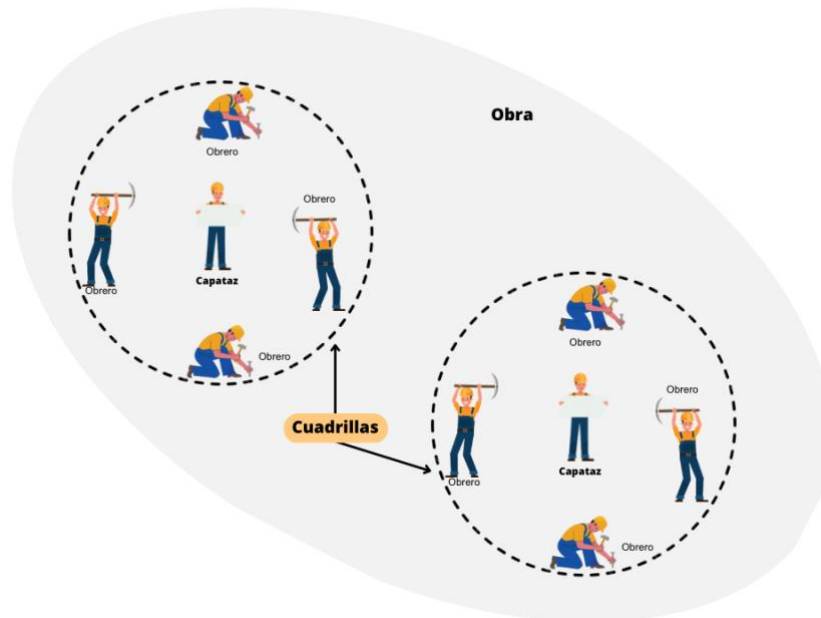


Ilustración 3 Estructura de Una Obra

La cuadrilla de trabajo en la construcción se compone típicamente de un capataz, también conocido como jefe de cuadrilla o encargado, y varios obreros o trabajadores. El capataz es responsable de supervisar y dirigir las actividades de los obreros, así como de coordinar y asignar las tareas de acuerdo con los requerimientos del proyecto.

Los obreros en la cuadrilla son los trabajadores que realizan las labores físicas en el lugar de construcción. Pueden ser albañiles, carpinteros, electricistas, encofradores u otros especialistas en construcción, dependiendo de las necesidades específicas del proyecto. Los obreros ejecutan las tareas asignadas por el capataz, siguiendo instrucciones y procedimientos establecidos, y colaboran entre sí para lograr los objetivos del proyecto.

La función principal de una cuadrilla es trabajar en equipo de manera coordinada para llevar a cabo las labores necesarias en la construcción. El capataz tiene la responsabilidad de liderar y supervisar al equipo, asegurándose de que se cumplan los plazos, la calidad del trabajo y las normas de seguridad. Además, el capataz puede proporcionar instrucciones técnicas, resolver problemas y brindar orientación a los obreros.

La formación de cuadrillas en la construcción permite una distribución eficiente de las tareas, optimiza la utilización de los recursos humanos disponibles y fomenta la colaboración y el trabajo en equipo. Cada cuadrilla puede ser asignada a una sección específica del proyecto o a una tarea particular, lo que permite una mayor especialización y eficacia en la ejecución de las labores constructivas.



### 3.5 Progressive Web App (PWA)

Una *Progressive Web App* (PWA) [13] es una aplicación web que combina características de una página web y de una aplicación móvil. Las PWAs están diseñadas para funcionar en cualquier navegador web y ofrecen una experiencia de usuario similar a la de una aplicación nativa, brindando características como notificaciones *push*, acceso offline, instalación en el escritorio o pantalla de inicio, y un rendimiento rápido y fluido.

En el contexto de un software para el control horario y la gestión laboral en empresas de construcción, una PWA puede ser una solución muy adecuada por varias razones:

- Accesibilidad multiplataforma: Al ser una aplicación web, una PWA puede ejecutarse en diferentes dispositivos y plataformas, como computadoras de escritorio, *tablets* y *smartphones*. Esto permite a los empleados acceder a la aplicación desde sus propios dispositivos personales, sin necesidad de instalar aplicaciones nativas específicas para cada plataforma.
- Instalación en el escritorio o pantalla de inicio: Una PWA se puede instalar en el escritorio de una computadora o en la pantalla de inicio de un dispositivo móvil, creando un acceso directo para acceder rápidamente a la aplicación. Esto brinda una experiencia similar a la de una aplicación nativa, sin requerir descargas o actualizaciones desde una tienda de aplicaciones.
- Notificaciones *push*: Una de las ventajas es la capacidad de enviar notificaciones *push* a los usuarios, incluso cuando la aplicación no está abierta en el navegador. Esto puede ser útil para enviar recordatorios de registro de horarios, anuncios importantes o comunicaciones relevantes a los empleados de manera oportuna.
- Funcionamiento offline: Las PWAs pueden funcionar en modo offline o con conectividad intermitente. Esto es especialmente beneficioso en el contexto de la construcción, donde los empleados pueden estar trabajando en lugares remotos o con conectividad limitada. La aplicación puede almacenar datos localmente y sincronizarlos posteriormente cuando haya conexión a Internet.
- Actualizaciones automáticas: Al ser una aplicación web, las PWAs se actualizan automáticamente cuando se implementan nuevas versiones o mejoras en el servidor. Esto evita la necesidad de que los usuarios realicen actualizaciones manuales y garantiza que siempre estén utilizando la versión más reciente de la aplicación.
- Acceso a hardware: aunque el acceso al hardware del dispositivo no es tan completo como con las aplicaciones nativas, es suficiente para la mayoría de los casos, siendo posible acceder a componentes como cámara, micrófono, altavoz o sensores del dispositivo.

### 3.6 Arquitectura *serverless*

Una arquitectura *serverless* es un enfoque de desarrollo de aplicaciones donde los desarrolladores se centran en escribir el código de la aplicación sin preocuparse por la administración y el mantenimiento de los servidores en los que se ejecuta. En lugar de tener que gestionar infraestructuras tradicionales, en una arquitectura *serverless*, los proveedores de servicios en la nube se encargan de escalar automáticamente la infraestructura según la demanda y cobran solo por el tiempo de ejecución real de la aplicación.

Las principales ventajas de una arquitectura *serverless* son:

1. Escalabilidad automática: La plataforma en la nube maneja automáticamente el escalado de recursos según la carga de trabajo, lo que permite que la aplicación se adapte a picos de tráfico sin problemas.
2. Mayor velocidad de desarrollo: Los desarrolladores pueden enfocarse en escribir el código y entregar rápidamente funcionalidades sin perder tiempo en configurar y gestionar servidores.
3. Alta disponibilidad: Los proveedores de servicios en la nube garantizan un alto nivel de disponibilidad, lo que mejora la confiabilidad de la aplicación.
4. Flexibilidad y agilidad: Se pueden agregar y eliminar componentes fácilmente, lo que facilita la adaptación a cambios en los requerimientos o en la demanda del usuario.

Para realizar este proyecto, se ha optado por utilizar Firebase [14] de Google para la arquitectura *serverless*. Gracias a esto, se ha podido concluir dentro de los plazos esperados, centrandose los esfuerzos en cuestiones particulares para este software y obviando las cuestiones generales de bases de datos[15], hosting [16], etc. Inicialmente el coste de estos servicios es cero hasta superar ciertos límites de número de usuarios, cantidad de datos o llamadas de lectura a la base de datos, posteriormente el coste de mantenimiento será proporcional al uso.

## 4 Técnicas y herramientas

En este apartado se explican las técnicas y herramientas utilizadas para el desarrollo de cada fase del proyecto.

### 4.1 VueJS

Se ha utilizado el *framework* VueJS para desarrollar la parte *front-end* del proyecto. Mediante él, se ha creado una interfaz de usuario intuitiva constituida por componentes.

Se ha empleado VueJS por ser más sencillo de aprender e implementar que otros *frameworks* como React [17] o Angular [17], además de contar con una creciente popularidad [18]. Este *framework* utiliza los lenguajes comunes de desarrollo web, HTML, CSS y JavaScript en conjunto con un patrón MVVM para desacoplar la lógica de datos con la interfaz.

### 4.2 Framework UI PrimeVue

Para los componentes de interfaz, se ha utilizado como plantilla para el desarrollo sakai [19], perteneciente a la familia PrimeFaces [20] y más concretamente a la orientada hacia VueJS, PrimeVue [21]. Se ha utilizado esta plantilla por estar actualizada a las últimas corrientes de diseño de interfaces como modo oscuro [22], diseño plano o componentes minimalistas y redondos. Contiene casi todos los componentes necesarios para el desarrollo de la interfaz (DatePicker, botones, cajas de texto, formularios, notificaciones toast...). Además, es gratuita y cuenta con una extensa y detallada documentación de todos sus componentes.

Junto con PrimeFlex, una biblioteca de CSS ligera y responsive, permite personalizar la interfaz de forma sencilla mediante clases predefinidas que agilizan el diseño.

### 4.3 NodeJS

Se ha usado Node.js [23] como entorno de ejecución ya que es de código abierto y que permite ejecutar JavaScript en el lado del servidor. Se ha utilizado por ser eficiente, ligero, escalable, multiplataforma y asíncronico para el desarrollo de aplicaciones web como es el caso además de contar con una gran comunidad, soporte y popularidad.

Además, con NPM (Node Package Manager) [24] que es el administrador de paquetes predeterminado para el ecosistema de Node.js y el utilizado durante este proyecto. Se pueden administrar las dependencias y bibliotecas de terceros utilizadas.

## 4.4 Bibliotecas

Algunas de las bibliotecas que se han usado para completar las funcionalidades del proyecto son:

- **FullCalendar**: herramienta que se utiliza para crear calendarios interactivos y personalizables en aplicaciones web [25]. Proporciona una amplia gama de funcionalidades, como la visualización de eventos, la navegación por fechas, la gestión de eventos recurrentes, la integración con fuentes de datos externas y la personalización de la apariencia del calendario. Esta biblioteca se ha utilizado para la gestión del calendario laboral de la empresa (Ilustración 4).

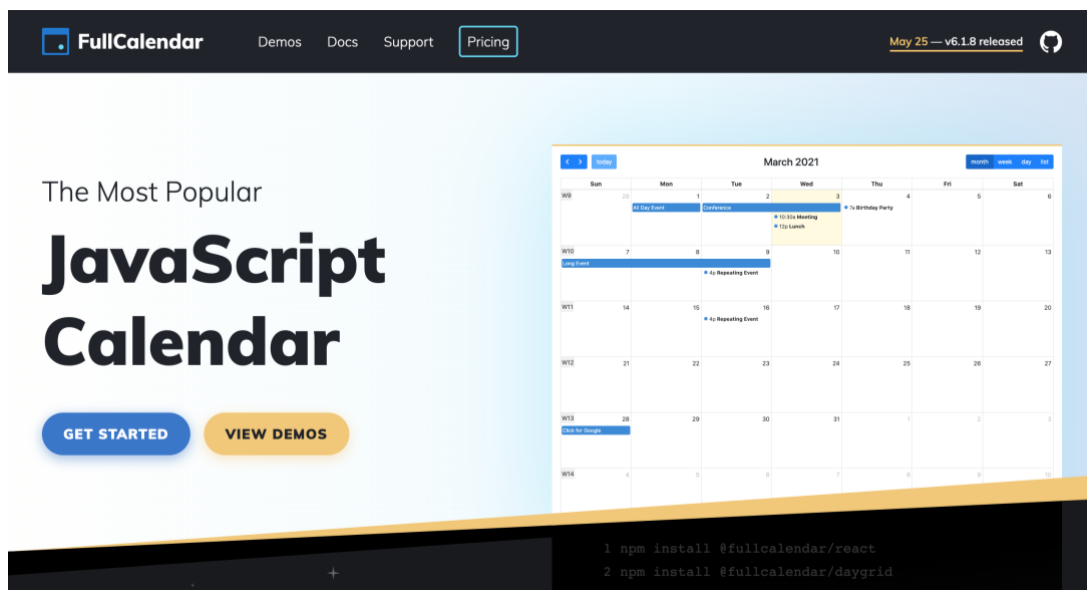


Ilustración 4 FullCalendar [25]

- **Json-As-Xlsx** [26]: es una herramienta que permite convertir datos en formato JSON a archivos de hojas de cálculo XLSX (Excel). Es útil en aplicaciones donde se requiere la generación y descarga de informes o datos en formato de hojas de cálculo para su procesamiento o análisis posterior. Es utilizada para generar hojas de cálculo a partir de los resultados obtenidos por los datos almacenados.
- **ApexCharts.js** [27]: es una biblioteca de gráficos interactivos y responsivos en JavaScript. Proporciona una amplia variedad de tipos de gráficos, como gráficos de líneas, barras o áreas entre otros como se puede ver en la Ilustración 5. Se utiliza en el apartado de resultados para visualizar y presentar datos forma de mapa de calor sin necesidad de exportarlos previamente como una hoja de cálculo.

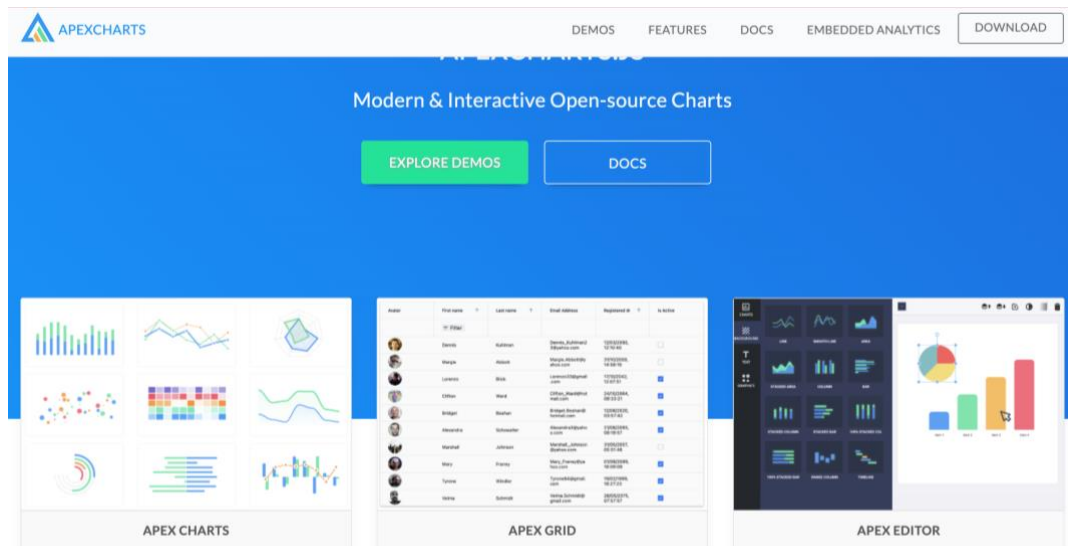


Ilustración 5 ApexCharts [27]

- **QR code Vue** [28]: una biblioteca de componentes para la generación de códigos QR en aplicaciones Vue.js. Permite generar códigos QR a partir de una cadena de texto, en este caso para representar las identificaciones de los empleados en forma de QR.
- **Vue3 QR Reader** [29]: permite la lectura de códigos QR en aplicaciones Vue.js. Permite tanto a empleados de oficina leer los códigos de acceso como a los capataces escanear las identificaciones de los obreros para pasar lista.
- **Jspdf** [30]: La biblioteca jspdf es una biblioteca de JavaScript que permite generar archivos PDF en el lado del cliente. Es una herramienta útil para generar documentos PDF dinámicos en aplicaciones web sin la necesidad de enviar solicitudes al servidor. Se utiliza para poder exportar las identificaciones de los empleados en formato pdf para posteriormente imprimirlos y poder facilitárselos.
- **vue/cli-plugin-pwa** [31]: permite convertir aplicaciones Vue.js en aplicaciones web progresivas. Proporciona herramientas y configuraciones automáticas para habilitar características como el almacenamiento en caché, notificaciones *push* y personalización de íconos, lo que permite a los desarrolladores crear aplicaciones web que se comportan y se sienten más como aplicaciones nativas.
- **I18n** [31]: proporciona una serie de funciones y métodos que permiten la gestión de cadenas de texto y la localización de contenido en la aplicación. Estas funciones permiten cargar y cambiar dinámicamente el contenido traducido según el idioma preferido o la ubicación del usuario.

### 4.5 Firebase

Firebase [14] es una plataforma de desarrollo de aplicaciones móviles y web, propiedad de Google, que ofrece una amplia gama de servicios y herramientas para facilitar el desarrollo, el crecimiento y la gestión de aplicaciones. Con Firebase, se pueden crear aplicaciones rápidamente sin preocuparse por la infraestructura, ya que proporciona servicios como almacenamiento en la nube, autenticación de usuarios, bases de datos en tiempo real, notificaciones *push*, análisis de datos, alojamiento web y más. Firebase se destaca por su facilidad de uso, escalabilidad, alta calidad y rendimiento.

En el proyecto se han utilizado estos servicios para las siguientes funcionalidades:

- Para la autenticación de usuarios, se ha utilizado **Firestore Authentication** [32]. Esta solución integral de identidad es compatible con cuentas de correo electrónico y contraseñas, así como con el acceso a través de Google, Facebook, entre otros. Proporciona servicios de *back-end*, como validación de cuentas de correo electrónico u opciones de recuperar la contraseña.
- Para la gestión de la base de datos, se ha utilizado **Cloud Firestore** [15]. Esta base de datos NoSQL de documentos permite almacenar, sincronizar y consultar datos de forma sencilla tanto para aplicaciones móviles como web. Es flexible, escalable y garantiza la sincronización de datos en tiempo real.
- Para el almacenamiento de caché y sistemas de claves dinámicas para fichar, se ha utilizado **Firestore RealTime Database** [33].
- Para enviar notificaciones *push*, se utiliza el servicio **Cloud Messaging**, almacenando los tokens de cada usuario en el caché de RealTime Database, agrupados por empresas e identificados con su correspondiente UID (*User Identification*).
- Para el alojamiento de la aplicación, se ha utilizado **Firestore Hosting** [16] en este caso como una aplicación web de una sola página o una aplicación web progresiva.
- Para gestionar las funciones *back-end* como el chequeo de ausencias, se ha utilizado **Cloud Functions** [34], que permite ejecutar de forma automática el código en respuesta a solicitudes HTTPS, o en respuesta a eventos de la base de datos.

### 4.6 Arduino

Arduino es una plataforma de hardware de código abierto diseñada para facilitar la creación de proyectos electrónicos interactivos. Existen múltiples microcontroladores compatibles con este *framework* y en este caso se ha empleado la placa de desarrollo M5Sick-CPlus y el IDE propio de Arduino [35]. Este dispositivo se utiliza como punto de control de accesos, que genera de forma periódica códigos QR que los empleados deben escanear para validar su asistencia.

La placa de desarrollo utilizada se basa en el microcontrolador ESP32 [36], [37] y para llevar a cabo las diferentes funcionalidades se han empleado distintas bibliotecas como:

- **Firestore** [38]: que permite al dispositivo acceder a la base de datos RealTime Database de Firestore y así almacenar los códigos además de escuchar por cambios en esta.
- **Wifi Manager** [37]: para facilitar la configuración del aparato una vez entregado al cliente. Esta biblioteca, da la posibilidad de crear una red Wi-Fi local a la que conectarse, para desde una interfaz web, seleccionar y conectar a una red Wi-Fi junto con otros parámetros configurables.

Se ha optado por este dispositivo por su bajo precio, sencillez y fiabilidad además de que aporta todo el hardware que necesitamos para este proyecto principalmente el módulo Wi-Fi ESP32 y pantalla LCD (del inglés, *Liquid Crystal Display*, en español, Pantalla de cristal líquido)<sup>1</sup> capaz de representar códigos QR.

#### 4.7 Herramientas de diseño

Los primeros bocetos de las interfaces de usuario se han realizado mediante *wireframes*, estos bocetos se han dibujado a mano a con una plantilla propia que distingue entre dispositivo móvil u ordenador.

Para la paleta de colores se ha utilizado la herramienta Adobe Color [39] que permite no solo obtener paletas de distintas combinaciones si no comprobar la accesibilidad por contraste entre distintos colores.

El logo de la aplicación se ha dibujado con la herramienta Canva [40], con su opción para logos y empleando los recursos de formas y objetos que contiene.

#### 4.8 Herramientas CASE

Para la realización de las diferentes partes de la documentación del proyecto se han utilizado las siguientes herramientas:

- Para la realización de la planificación temporal del proyecto se ha utilizado **Microsoft Project** [41], que es un software para la administración de que permite entre otras cosas el desarrollo de planificaciones temporales, asignación de recursos a tareas y administrar el presupuesto.
- Para la estimación del esfuerzo requerido en el desarrollo del proyecto se ha usado **EZEstimate**, que es una herramienta para calcular la estimación del esfuerzo de desarrollo de software utilizando la metodología Puntos Casos de Uso [42].
- Para la realización de los diagramas UML necesarios se ha utilizado **Visual Paradigm** [43], que es una herramienta CASE que permite la realización de diferentes tipos de diagramas compatibles con diversas notaciones de modelado. De forma conjunta también se han utilizado **draw.io** [43], una herramienta online de creación de diagramas y **PlantUML** [44], una herramienta online de creación de diagramas a partir de texto.

---

<sup>1</sup> Pantalla LCD: pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. Utiliza cantidades muy pequeñas de energía eléctrica.

## 5 Aspectos relevantes del desarrollo

Este apartado trata las particularidades más relevantes del desarrollo de este proyecto.

Se ha seguido el marco de trabajo de Proceso Unificado [45], un enfoque de desarrollo de software que se basa en principios de Ingeniería del Software para guiar y estructurar el ciclo de vida del desarrollo de proyectos. El Proceso Unificado es un enfoque de desarrollo iterativo e incremental compuesto por cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada una de estas fases se divide a su vez en una serie de iteraciones (la fase de inicio puede incluir varias iteraciones en proyectos grandes). Estas iteraciones generan un incremento en el producto desarrollado, que agrega o mejora las funcionalidades del sistema en desarrollo.

### 5.1 Ciclo de vida

Como se puede observar en la, el proceso unificado divide el ciclo de vida en cuatro fases con sus respectivas iteraciones, 11 en total en el ejemplo mostrado en la Ilustración 6.

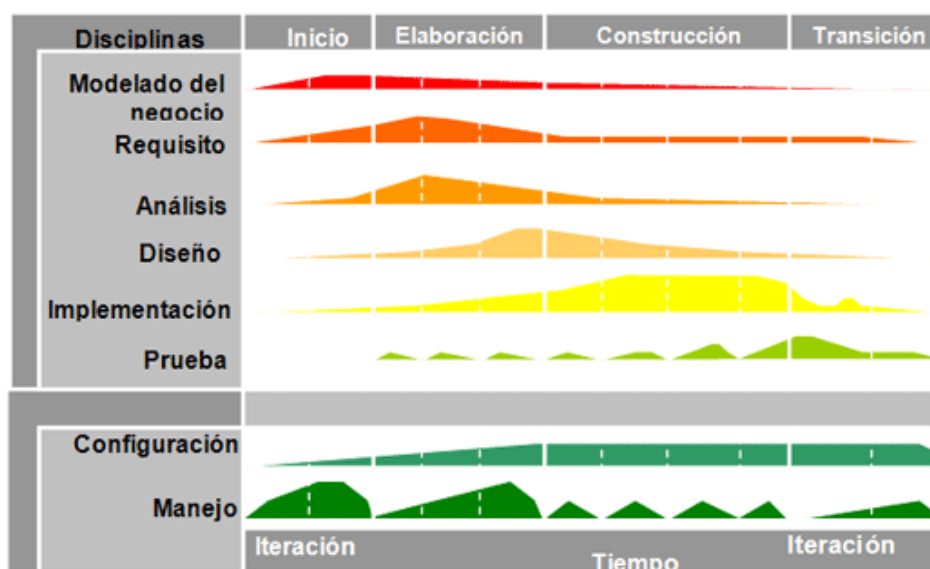


Ilustración 6 Fases Proceso Unificado [46]

Las cuatro fases del ciclo de vida son:

1. Inicio: En esta fase, se establece el alcance del proyecto, se identifican los objetivos y se define la visión general del sistema mediante reuniones con la empresa colaboradora. Se realiza un análisis de viabilidad para evaluar la factibilidad técnica, económica y operativa del proyecto. Además, se identifican los riesgos principales y se elabora un plan preliminar para el desarrollo del sistema. Al final de esta fase, se obtiene la aprobación para pasar a la siguiente fase.

2. Elaboración: En esta fase, se realiza un análisis detallado de los requisitos del sistema. Se definen los casos de uso, los actores (los roles de cada actor), se elabora el modelo de dominio y se identifican los elementos arquitectónicos clave. También se desarrolla un plan de proyecto más detallado, mediante los casos de uso se realiza una estimación del



esfuerzo para concretar los plazos y recursos necesarios. Al final de esta fase, se espera tener una arquitectura *serverless* definida y lista para comenzar a implementar.

3. Construcción: se lleva a cabo la implementación y construcción del sistema en iteraciones incrementales. Se desarrollan los componentes y vistas del sistema siguiendo la arquitectura definida, se realizan pruebas unitarias en conjunto con la arquitectura *serverless* comprobando su correcto funcionamiento. Se continúa refinando y validando que los requisitos se cumplen a medida que se avanza en el desarrollo. Al final de esta fase, se espera tener una versión funcional y probada del sistema.

4. Transición: En esta fase, se prepara el sistema para su despliegue en el hosting y uso en el entorno de producción. Se llevan a cabo pruebas de integración, pruebas de aceptación, pruebas *responsive* y se realizan actividades de capacitación y documentación. Una vez que el sistema se encuentra en producción, se monitorea su desempeño y se abordan las incidencias que puedan surgir dando soporte a los usuarios. Al final de esta fase, el sistema se considera completamente implementado y en funcionamiento.

Es importante destacar que el Proceso Unificado es un enfoque iterativo, lo que significa que las actividades se repiten en ciclos sucesivos a medida que se obtiene retroalimentación y se realizan mejoras incrementales. Esto permite una adaptación flexible a medida que se comprenden mejor los requisitos y se enfrentan nuevos desafíos durante el desarrollo del software.

En el Anexo I de esta documentación técnicas se pueden observar de forma detallada todas las fases con sus iteraciones. Esta planificación se ha desarrollado con el software Microsoft Project como se explica en el apartado 4.8 Herramientas CASE.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		Inicio	21,63 días	lun 01/09/23	mar 02/07/23		
2		Iteración 1	11,13 días	lun 01/09/23	mar 01/24/23		
3		Modelado de negocio	4,13 días	lun 01/09/23	vie 01/13/23		
4		Reunión con el cliente	0,5 días	lun 01/09/23	lun 01/09/23		Rodrigo de la Calle
5		Reunión con el tutor	0,5 días	lun 01/09/23	mar 01/10/23	4	Rodrigo de la Calle
6		Identificación de Plazos, costes y viabilidad	2 días	mar 01/10/23	jue 01/12/23	5	Rodrigo de la Calle
7		Primer informe	1 día	jue 01/12/23	vie 01/13/23	6	Rodrigo de la Calle
8		Requisitos	2 días	vie 01/13/23	mar 01/17/23	3	
9		Definición de Objetivos	1 día	vie 01/13/23	lun 01/16/23		Rodrigo de la Calle
10		Primer aproximación de requisitos funcionales	1 día	lun 01/16/23	mar 01/17/23	9	Rodrigo de la Calle
11		Análisis	5 días	mar 01/17/23	mar 01/24/23	8	
12		Modelo de Dominio	3 días	mar 01/17/23	vie 01/20/23		Rodrigo de la Calle[50%]
13		Paquete de Análisis	2 días	vie 01/20/23	mar 01/24/23	12	Rodrigo de la Calle[50%]
14		Hito fin iteración 1	0 días	mar 01/24/23	mar 01/24/23	2	

Ilustración 7 Ciclo de Vida Microsoft Project

## 5.2 Especificación de requisitos

En este apartado se describe el comportamiento del software mediante la especificación de los requisitos para la que se ha seguido la metodología de licitación de requisitos de Durán y Bernández [47].

En el Anexo II se detalla toda la especificación de requisitos. Contiene las tablas de los casos de uso, diagramas de arquitectura, diagramas de secuencia y diseño de la interfaz.

Como pueden verse en las siguientes ilustraciones, el sistema se ha organizado en tres paquetes:

- Gestión de Usuarios: se corresponde con los casos de uso de gestión de todos los usuarios del sistema (Ilustración 8).
- Empresa: se corresponde con los casos de uso de gestión de los contenidos de las empresas (empleados, obras, calendario de trabajo, etc.) (Ilustración 9).
- Empleados: se corresponde con los casos de uso que implican de forma activa a los empleados de las empresas (Ilustración 10).

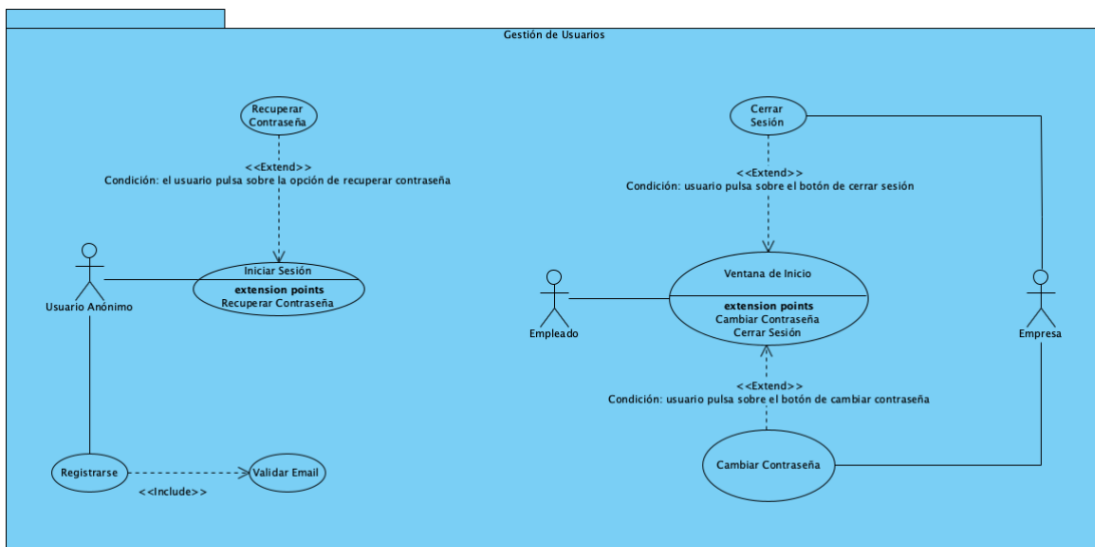


Ilustración 8 Paquete Gestión de Usuarios

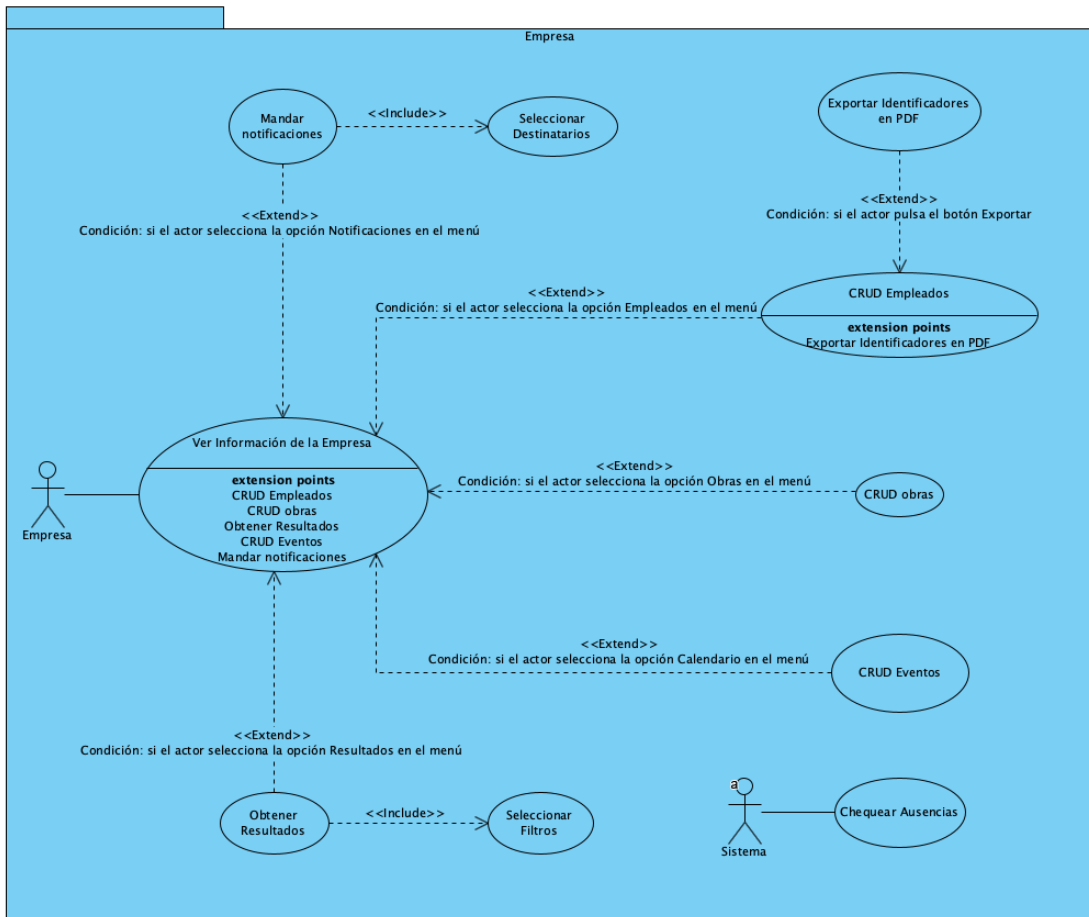


Ilustración 9 Paquete Empresa

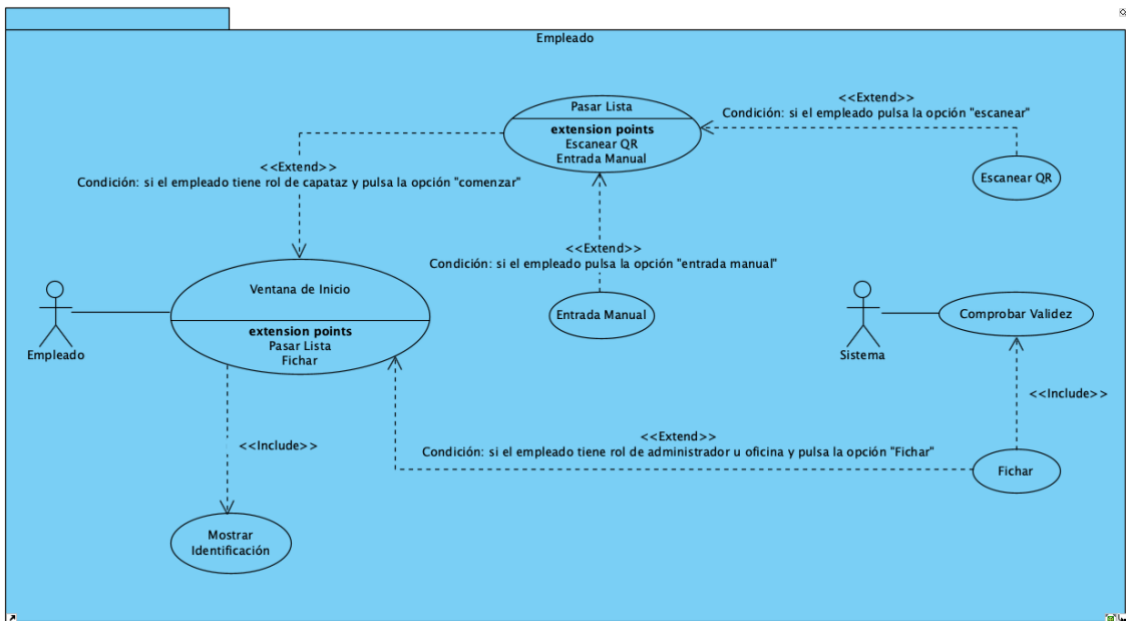


Ilustración 10 Paquete Empleado

## 5 Aspectos relevantes del desarrollo

### 5.3 Especificación de análisis

En este apartado se define el modelo del dominio del sistema mediante un diagrama de clases como el de la Ilustración 11 y se detalla la interacción de los casos de uso con los diagramas de secuencia como el de la Ilustración 12. El resto de los diagramas e información detallada se encuentran en el Anexo III.

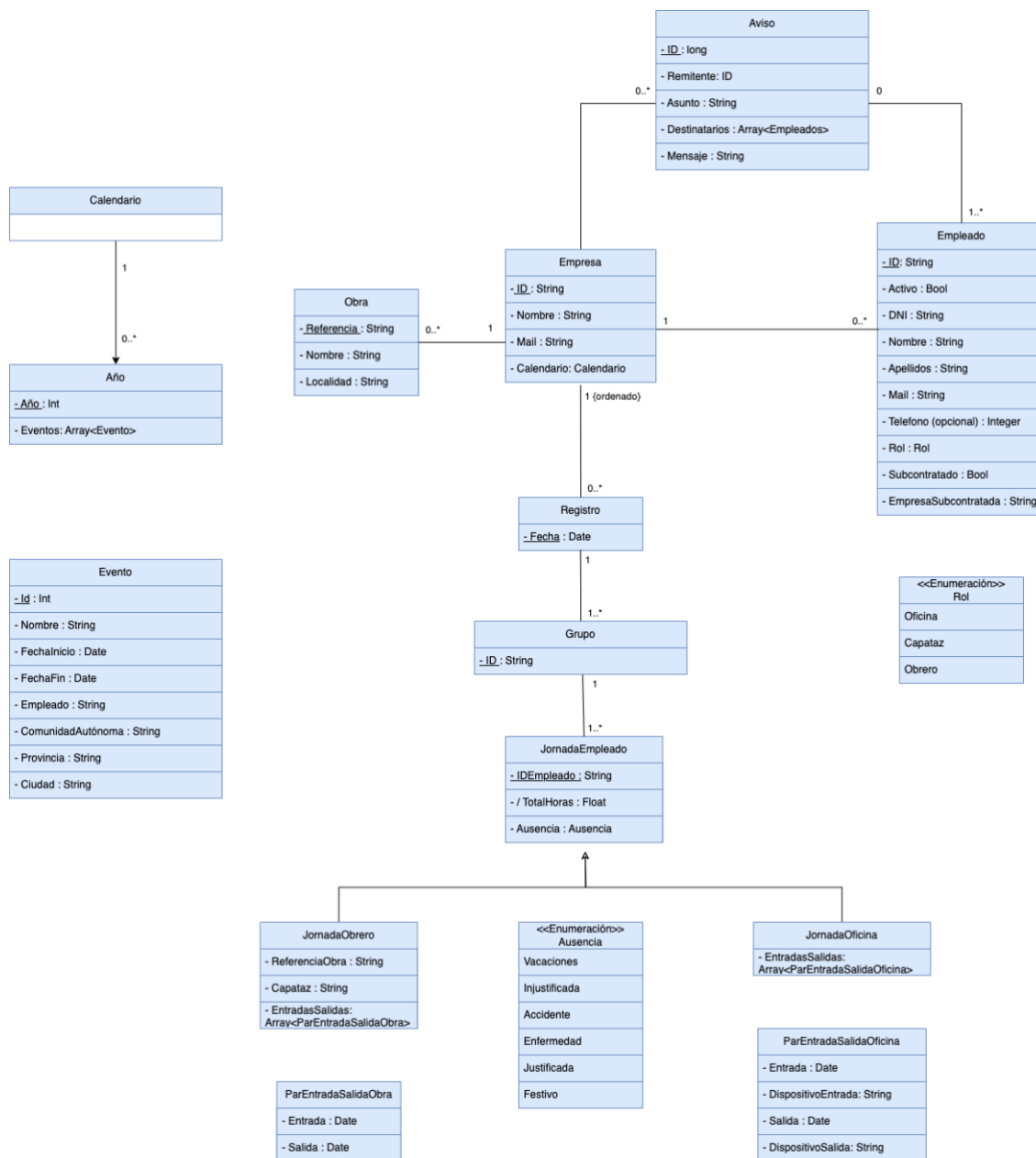


Ilustración 11 Diagrama de Clases

En la Ilustración 12, se puede observar el diagrama de secuencia para el caso de uso de chequear las ausencias. En este caso de uso, el actor sistema, una vez finalizado el día, comprueba en cada empresa qué empleados no han marcado ni asistencia ni ausencia, comprobando si hay algún tipo de evento asociado a ese empleado o ubicación para así ponerle la falta que le corresponde o marcarlo como injustificado.

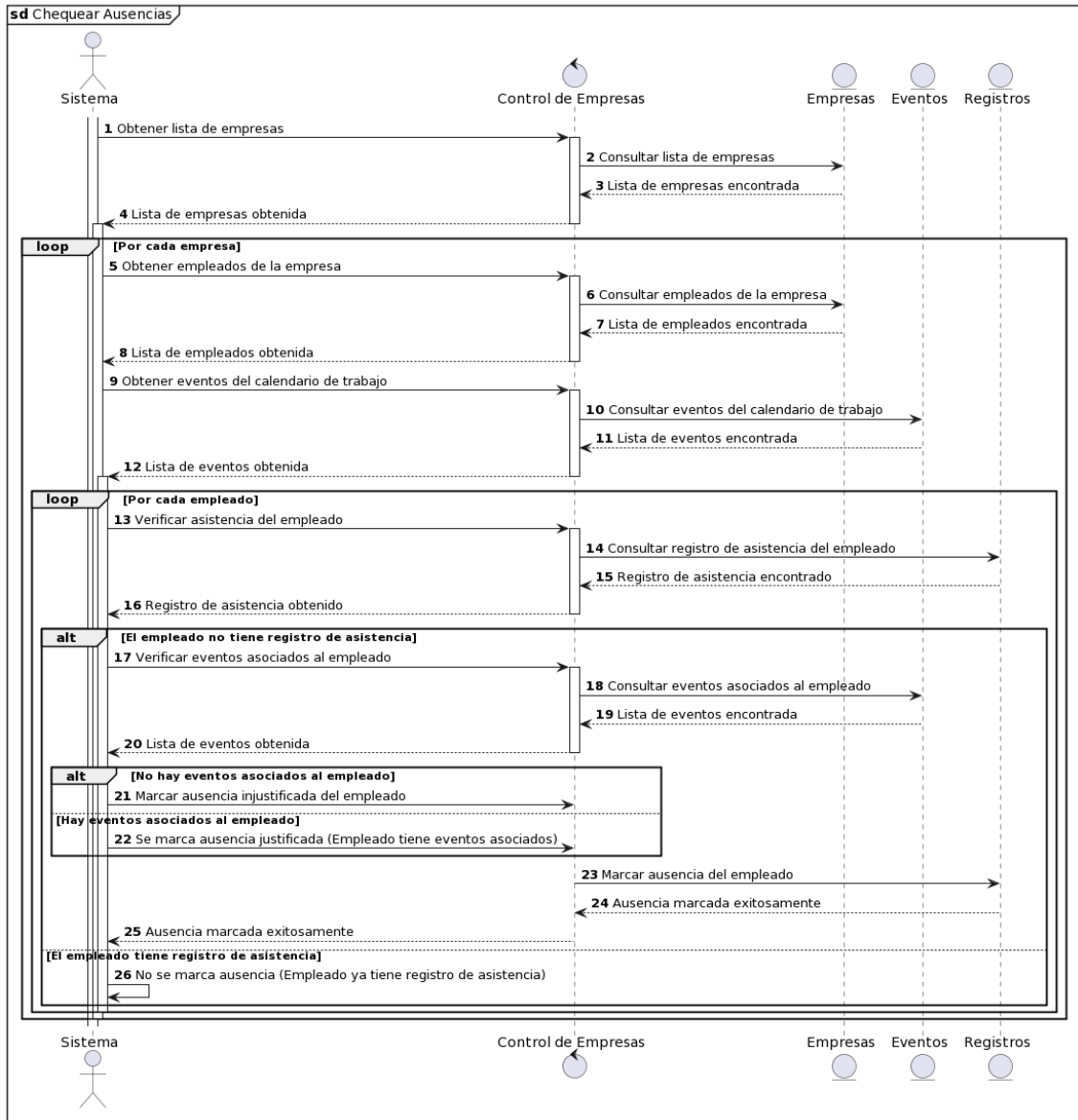


Ilustración 12 Diagrama de Secuencia

## 5.4 Especificación de diseño

En esta parte se detalla el diseño del sistema, indicando la arquitectura del sistema, el diseño final del modelo del dominio y diagramas de secuencia de los casos de uso con las interfaces diseñadas como el que se ve de ejemplo en la Ilustración 15. El resto de información se encuentra de forma detallada en el Anexo III.

Para realizar este proyecto, se ha optado por utilizar Firebase [14] de Google para la arquitectura *serverless*. Gracias a esto, se ha podido concluir dentro de los plazos esperados, centrando los esfuerzos en cuestiones particulares para este software y obviando las cuestiones generales de bases de datos [15], hosting [16], etc. Inicialmente el coste de estos servicios es cero hasta superar ciertos límites de número de usuarios, cantidad de datos o llamadas de lectura a la base de datos, posteriormente el coste de mantenimiento será proporcional al uso.

### 5.4.1 Patrones Arquitectónicos

En la parte de la aplicación del *front-end* se ha empleado el patrón *Model-View-ViewModel* (MVVM). La finalidad principal de este patrón es desacoplar lo máximo la interfaz y la lógica de negocio de la aplicación.

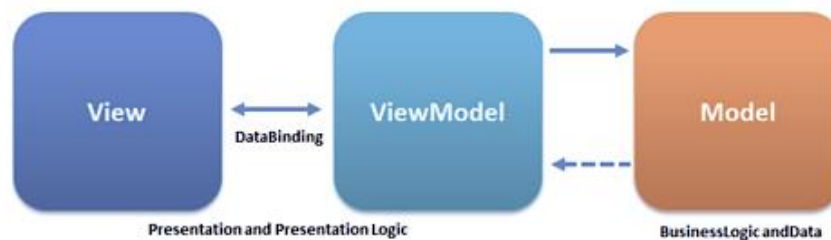


Ilustración 13 Patrón MVVM [48]

Las tres partes son:

- **El modelo (*Model*)** representa la capa de datos. Contiene la información, pero nunca las acciones o servicios que la manipulan. En ningún caso tiene dependencia alguna con la vista.
- **La vista (*View*)** representa la información a través de los elementos visuales que la componen. Contienen comportamientos, eventos y enlaces a datos que, en cierta manera, necesitan tener conocimiento del modelo subyacente.
- **El *ViewModel*** es un actor intermediario entre el modelo y la vista, contiene toda la lógica de presentación y se comporta como una abstracción de la interfaz. La comunicación entre la vista y el *ViewModel* se realiza por medio los enlaces de datos (*binders*).

Para evitar la dependencia entre el modelo de datos y la vista, se han utilizado *stores* de pinya donde se centralizan todos los datos y los métodos que enlazan estos datos con las Firestore Database. Se refleja en la Ilustración 14.

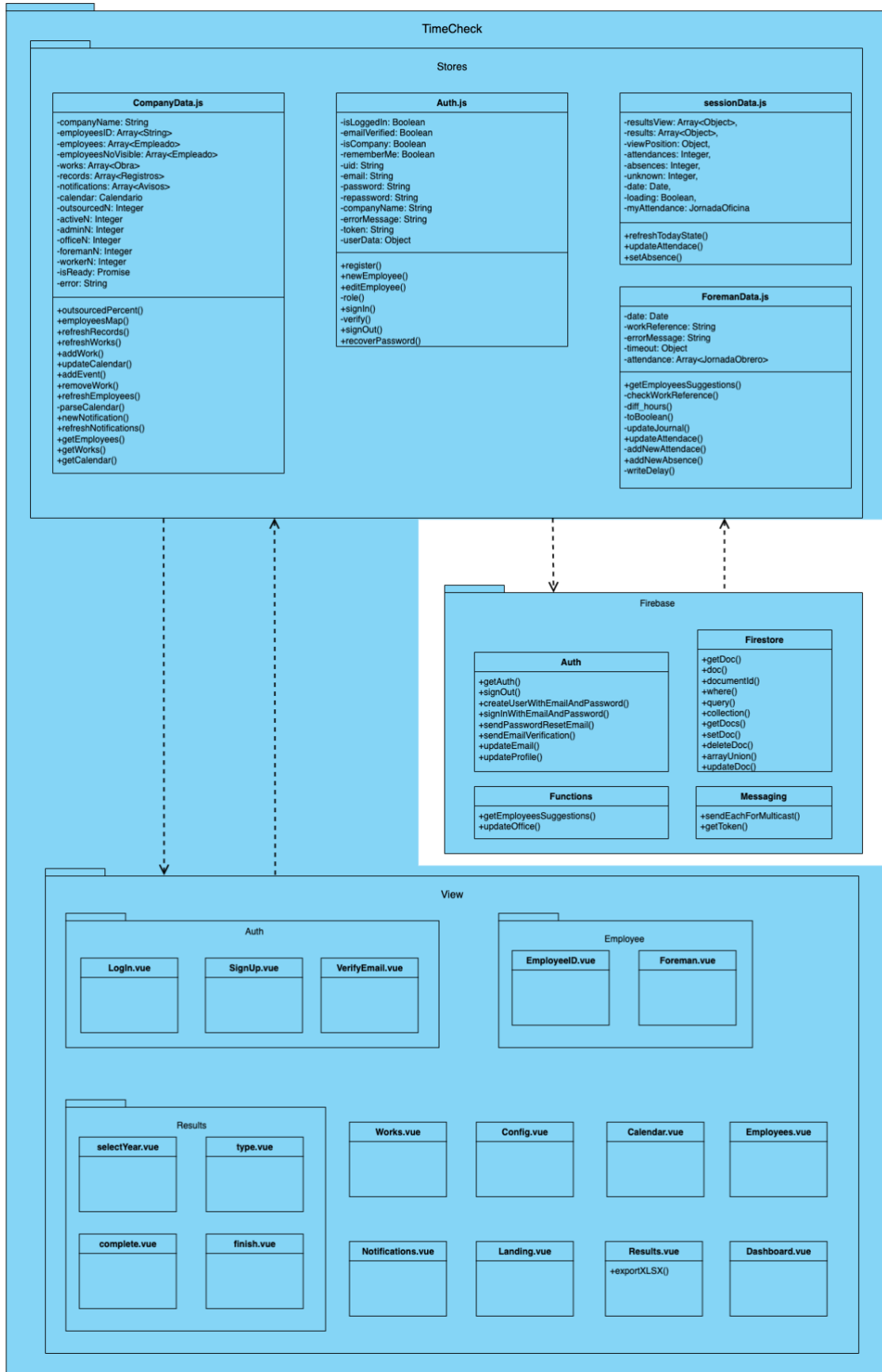


Ilustración 14 Diseño, Arquitectura

## 5 Aspectos relevantes del desarrollo

En la Ilustración 15, se puede observar el diagrama de secuencia de diseño en el que el capataz realiza el caso de uso de registrar asistencia. Primero se valida la referencia de la obra en la que van a trabajar y después escanea las identificaciones de sus obreros o marca las ausencias del que falte.

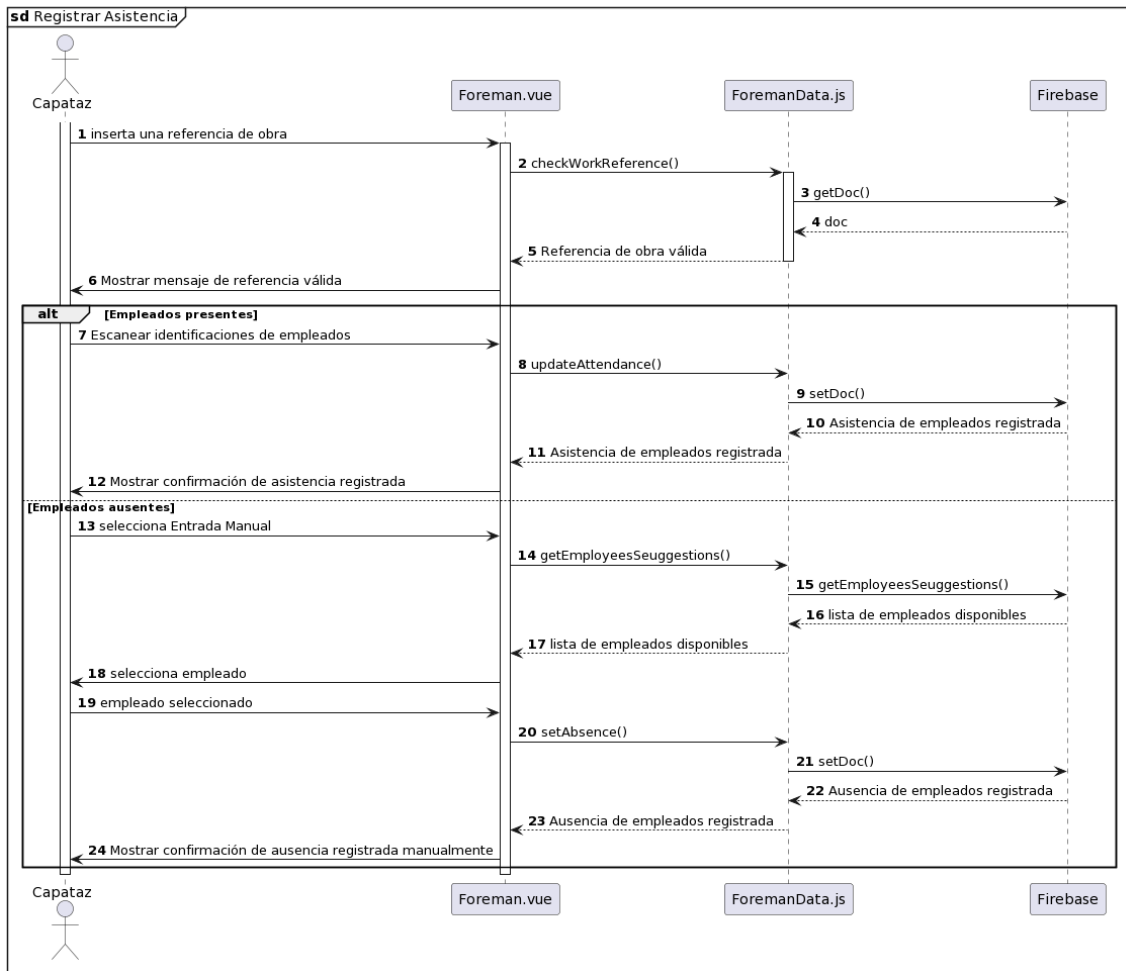


Ilustración 15 Ejemplo de Diagrama de Secuencia de Diseño



### 5.4.2 Modelo del dominio

El diagrama de clases del modelo del dominio final es el que se muestra en la Ilustración 16. Los métodos son en su mayoría *getters* y *setters*.

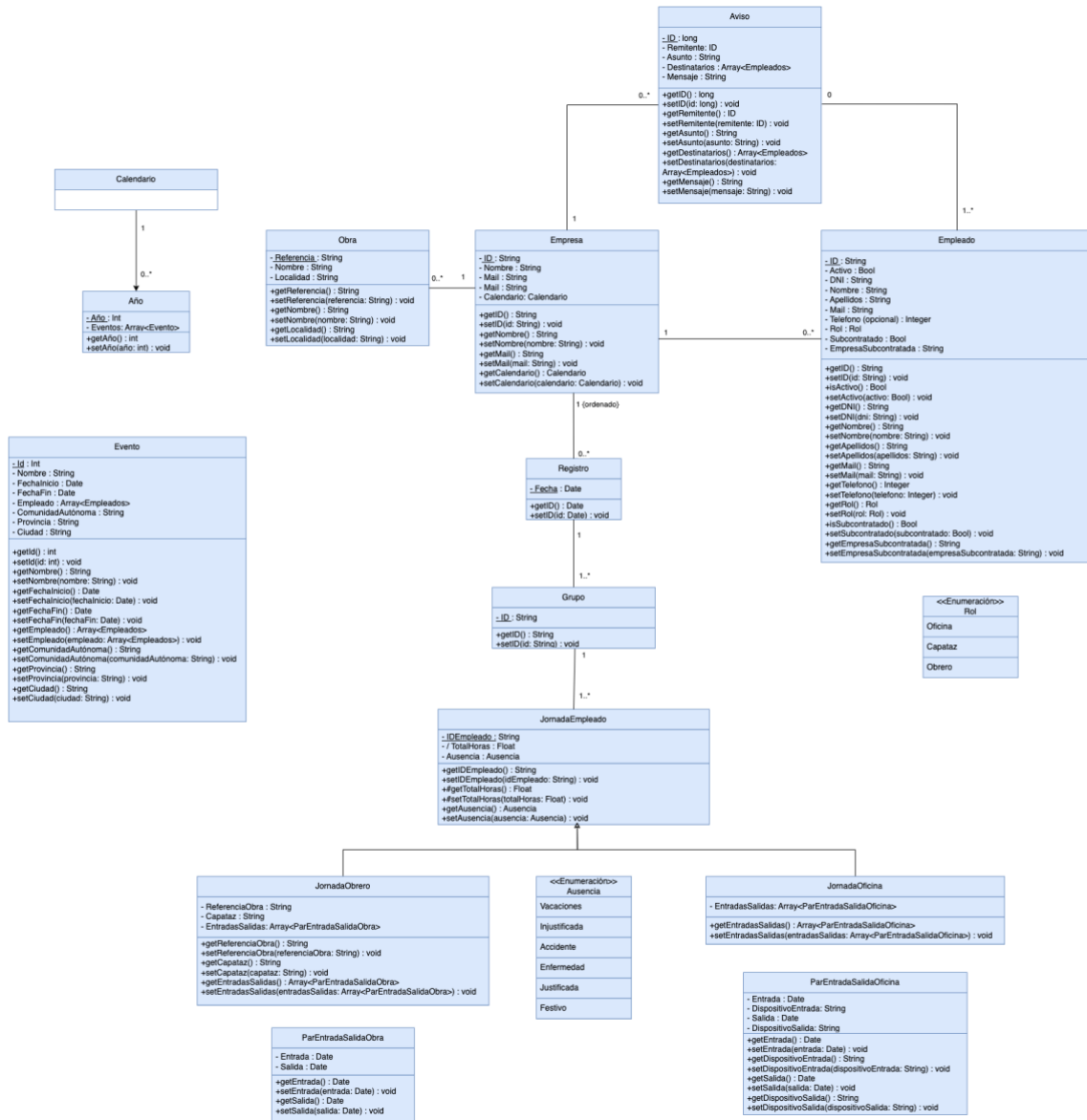


Ilustración 16 Diseño, Diagrama de Clases

A continuación, se muestra una breve explicación de lo que representa cada clase:

- **Empresa:** representa al usuario que se ha registrado por sí mismo en la plataforma identificándose como una empresa.
- **Empleado:** representa a los usuarios que un usuario empresa registra en el software y que estarán asociados a la empresa que los ha registrado.
- **Obra:** representa a las construcciones en las que trabaja la empresa.
- **Aviso:** representa a las notificaciones que se pueden intercambiar de forma interna entre empleados y su empresa.
- **Registro:** representa a los documentos diarios donde se almacenan las jornadas.

## 5 Aspectos relevantes del desarrollo

- **Grupo:** representa la forma de agrupar las jornadas por cuadrillas de trabajo o por oficina.
- **JornadaEmpleado:** representa la asistencia o ausencia de un empleado para un día junto con las horas totales que ha trabajado. En el caso de ser un obrero o capataz almacenará un atributo con la referencia de la obra en la que trabaja, otro con el capataz que le ha marcado la asistencia y un array de ParEntradaSalidaObra. En el caso de ser un empleado con rol de oficina, almacenará un array de ParEntradaSalidaOficina.
- **ParEntradaSalidaObra:** representa una entrada y salida que ha tenido un empleado durante su jornada.
- **ParEntradaSalidaOficina:** representa una entrada y salida que ha tenido un empleado durante su jornada y en que dispositivos ha fichado.
- **Calendario:** representa una colección de años.
- **Año:** representa una colección que contiene todos los eventos y festivos de un año.
- **Evento:** representa un evento junto con su fecha y localidad.

### 5.4.3 Base de datos

La estructura de datos se basa en una colección “Users” que contiene los datos de cada usuario del sistema. En el caso de que el usuario sea de tipo empresa (no empleado), tendrá además una estructura para el calendario de trabajo y tres subcolecciones. Una subcolección que almacenará cada obra en un documento, otra para almacenar las notificaciones y una última que contendrá un documento diario con el registro de asistencia para ese día (Ilustración 17).

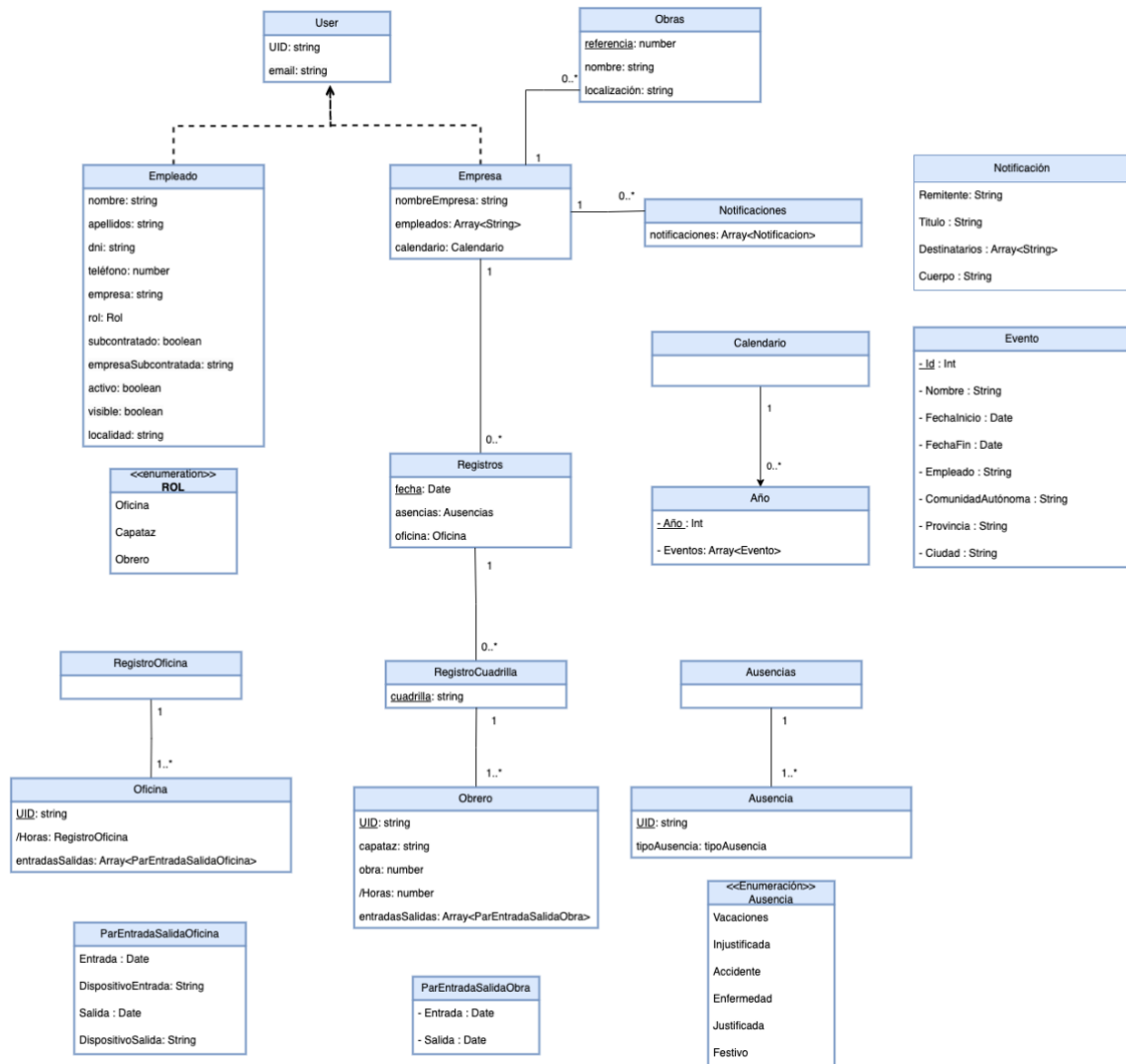


Ilustración 17 Diseño, Base de Datos

## 5 Aspectos relevantes del desarrollo

### 5.4.4 Interfaz de usuario

Se realizan los primeros bocetos de las interfaces de usuario de la aplicación. Estos bocetos se han dibujado a mano a con una plantilla propia que distingue entre dispositivo móvil u ordenador. En la Ilustración 18, se puede observar el *wireframe* diseñado para un dispositivo móvil, para el caso de uso en el que el obrero inicia sesión y ve su identificación. En la Ilustración 19, se puede observar el *wireframe* diseñado para un ordenador, para el caso de uso en el que el administrador quiere obtener los resultados de asistencia almacenados.

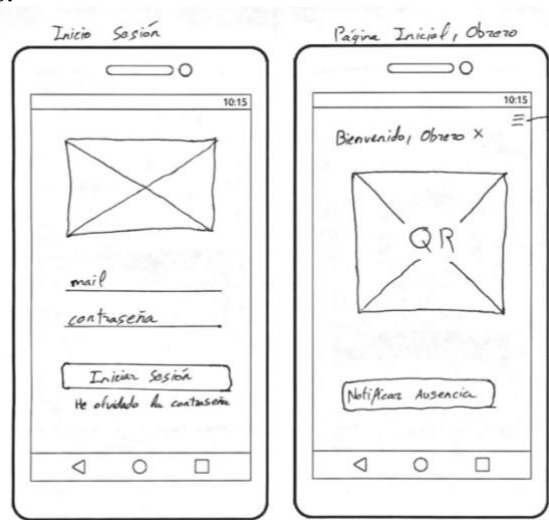


Ilustración 18 Diseño, Wireframe Obrero

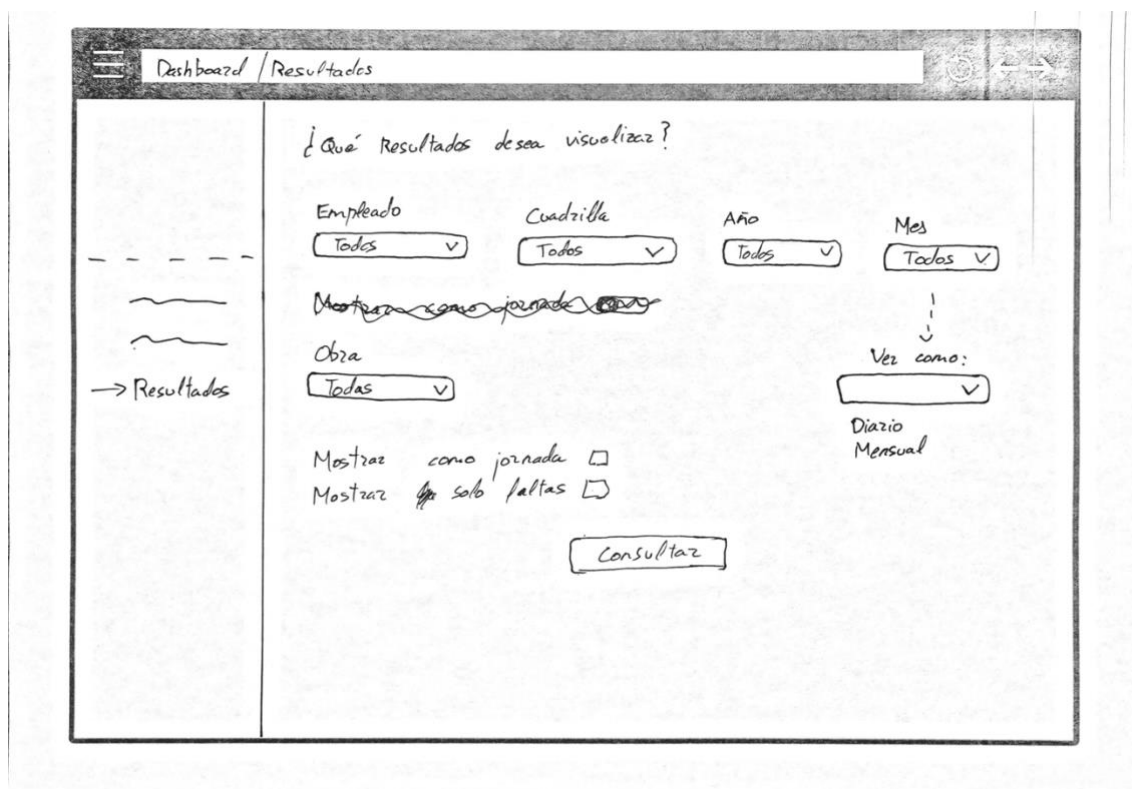


Ilustración 19 Diseño, Wireframe Resultados

### 5.4.5 Pruebas

Para asegurar la correcta evolución del proyecto, después de cada iteración del ciclo de vida (Anexo I) se han ido realizando pruebas manuales en las cuales se navega por los diferentes componentes del sistema, depurando y revisando errores que puedan surgir para así refinar el producto.

Las pruebas que se han llevado a cabo son las siguientes:

- **Pruebas unitarias:** pruebas de bajo nivel, cercanas al código fuente de la aplicación. Se enfocan en probar de forma individual las funciones y/o métodos utilizados por el software. Estas pruebas son específicas y generalmente automatizadas, lo que las hace de menor costo y permite su rápida ejecución mediante un servidor de integración continua. Idealmente, se aísla la funcionalidad para escribir pruebas independientes, verificando la adecuación de nombres, parámetros y resultados. Para evitar dependencias, se reemplazan los servicios externos por funciones locales similares. En algunos casos, también se reemplazan consultas a bases de datos para centrarse en los valores de entrada. Si no es posible aislar el uso de bases de datos, es importante optimizar las consultas para evitar pruebas unitarias de larga duración que puedan ralentizar el desarrollo.
- **Pruebas de integración:** verifican que los diferentes módulos y/o servicios usados por nuestra aplicación funcionen en armonía cuando trabajan en conjunto. Por ejemplo, la interacción con una base de datos,
- **Pruebas funcionales:** se centran en los requerimientos de negocio de una aplicación. Estas pruebas verifican la salida (resultado) de una acción, sin prestar atención a los estados intermedios del sistema mientras se lleva a cabo la ejecución.
- **Pruebas de punta a punta:** replican el comportamiento de los usuarios con el software, en un entorno de aplicación completo. Estas pruebas verifican que los flujos que sigue un usuario trabajen como se espera, y pueden ser tan simples como iniciar sesión.
- **Pruebas de aceptación:** son pruebas formales, ejecutadas para verificar si un sistema satisface sus requerimientos de negocio. Es una mezcla entre las pruebas de integración, funcionales y de punta a punta.

### 5.5 Implementación

En esta sección se describe cómo se han implementado los distintos componentes del sistema. Toda esto se puede encontrar de forma más detallada en el Anexo III.

Para la parte del *front-end*, se ha utilizado el *framework* VueJS (4.1 VueJS) junto con JavaScript, CSS y HTML. Se ha optado por usar este *framework* ya que a pesar de ser reciente y no tan famoso como React [17], su popularidad está creciendo recientemente debido a que es muy ligero, junto con su facilidad de uso y aprendizaje. Para gestionar el estado de la aplicación, se ha utilizado la biblioteca Pinia [49] para poder almacenar de forma local los datos de la aplicación.

También, se han empleado otras bibliotecas de componentes como FullCalendar para el calendario de trabajo o ApexCharts para gráficas como se explica en el punto 4.4 Bibliotecas.

Como *framework* se ha empleado Primefaces [20], más concretamente para la versión de vue3 PrimeVue [21]. La plantilla elegida dentro de las que se ofrecen ha sido sakai [19], esta plantilla nos proporciona los componentes necesarios, botones, cajas de texto, selectores de fecha, menús de diversos tipos, etc. Principalmente la interfaz se organiza en tarjetas.

#### 5.5.1 Despliegue

Para los servicios web se ha empleado Firebase, por ser estable, económico, con una gran comunidad, flexible y sencillo de utilizar. Hay que destacar de cara al despliegue, que se puede separar en *hosting*, que se encuentra alojado en Firebase Hosting y en funciones remotas de Cloud Functions, como se puede ver en la Ilustración 20.

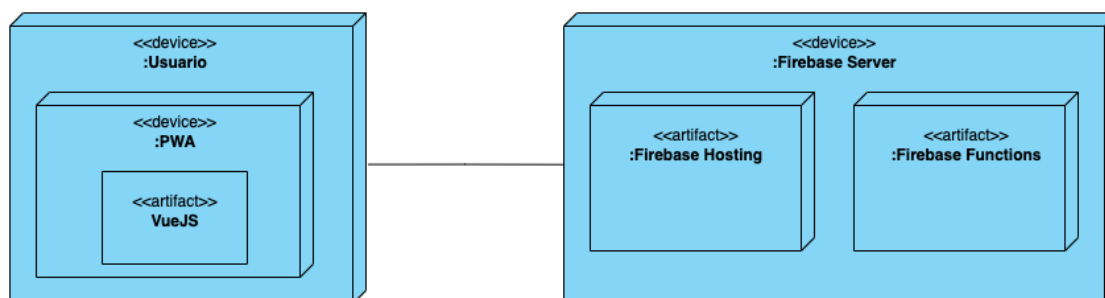


Ilustración 20 Diagrama de Despliegue

Para el despliegue local y realizar las pruebas se realiza de la siguiente manera:

1. **Instalar Node.js y Npm:** `sudo apt install nodejs / sudo apt install npm`
2. **Intalación:** `npm install`
  - a. **npm run dev:** ejecución de la aplicación en modo de desarrollo de forma local.
  - b. **npm run build:** construcción del proyecto para producción.
  - c. **npm run preview:** se utiliza para ejecutar una aplicación ya construida como una vista previa de la compilación de producción.

Para poder usar Firebase y sus utilidades se tiene que instalar la utilidad de consola denominada Firebase CLI [14], usando npm, y asociándose con un proyecto de Firebase creado. Pasos:

- **npm install -g firebase-tools:** instalación.
- **firebase login:** Inicio sesión en Firebase.
- **Hosting:**
  - **firebase init (Seleccionando Hosting):** Inicio proyecto de Firebase Hosting.
  - **firebase deploy --only hosting:** Despliegue de únicamente del hosting
- **Functions:**
  - **firebase init (Seleccionando Functions):** Inicio proyecto de Firebase Functions.
  - **firebase deploy --only functions:** Despliegue de únicamente las funciones.

### 5.5.2 Base de datos

Para almacenar los datos producidos se ha empleado Firestore Database, una base de datos NoSQL organizada por colecciones de documentos, que permite aplicar índices, ejecutar transacciones y almacenar todo tipo de variables comunes. Junto con esto, se han aplicado unas reglas de seguridad a la base de datos para asegurar los datos personales, de accesos ajenos. En la Ilustración 21, se puede ver cómo está estructurada la base de datos para cada tipo de usuario y colección.

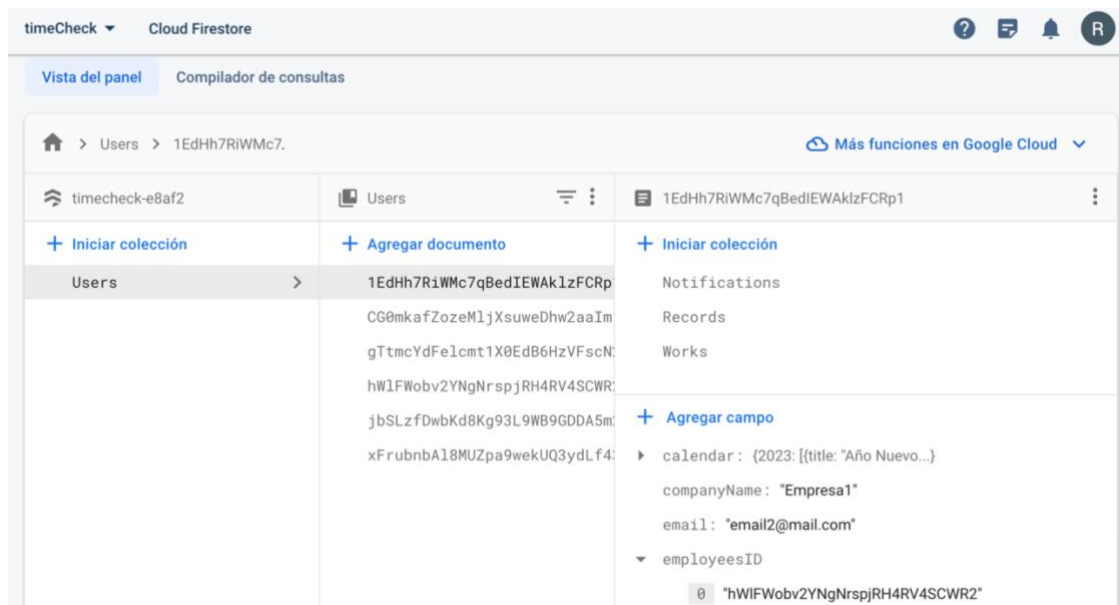


Ilustración 21 Firestore Database

De forma adicional, en RealTime Database, se almacenan las claves dinámicas que generan los dispositivos de oficina y una caché de algunos datos de Firestore, útiles para

## 5 Aspectos relevantes del desarrollo

agilizar el proceso de chequeo de ausencias. Se puede ver el esquema seguido en la Ilustración 22.

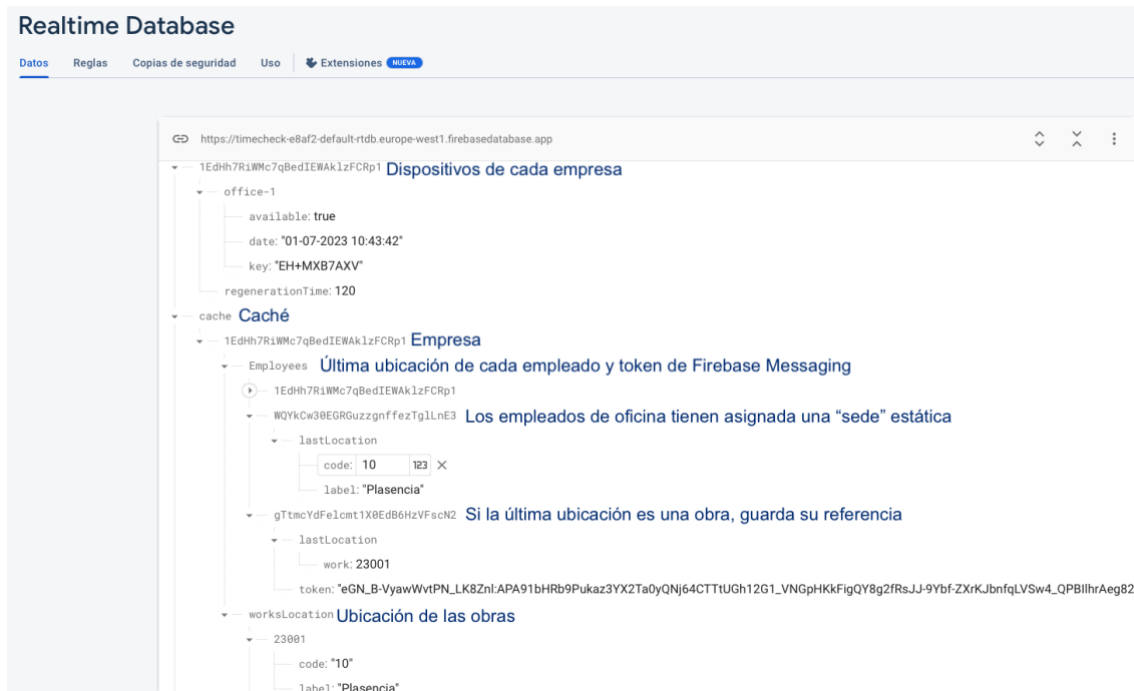


Ilustración 22 RealTime Database



## 5.6 Hardware

Para que los empleados de oficina puedan fichar en los espacios de trabajo determinados como almacenes u oficinas, se emplea el dispositivo M5Stick CPlus. Se ha utilizado este dispositivo por ser económico y cumplir todos los requisitos necesarios:

- Módulo ESP32: permite conectarse a redes Wi-Fi y además es compatible con RealTime Database.
- *Buzzer*: Permite reproducir sonidos usados como respuesta a acciones.
- Pantalla LCD: permite representar las claves dinámicas mediante códigos QR.
- Compatible con WifiManager: WifiManager [37] es una biblioteca para módulos ESP32 [36] que permite gestionar de forma sencilla las conexiones Wi-Fi mediante una interfaz web como las mostradas en la Ilustración 24,b y c. Primero, se establece como un punto de acceso al que conectarnos como se ve en la Ilustración 24, a. Después, de conectarse al dispositivo y acceder a la dirección IP que se muestra en su pantalla, podremos configurarlo como en la Ilustración 24,b e Ilustración 24,c.

La descripción de este componente se puede ver en la Ilustración 23.

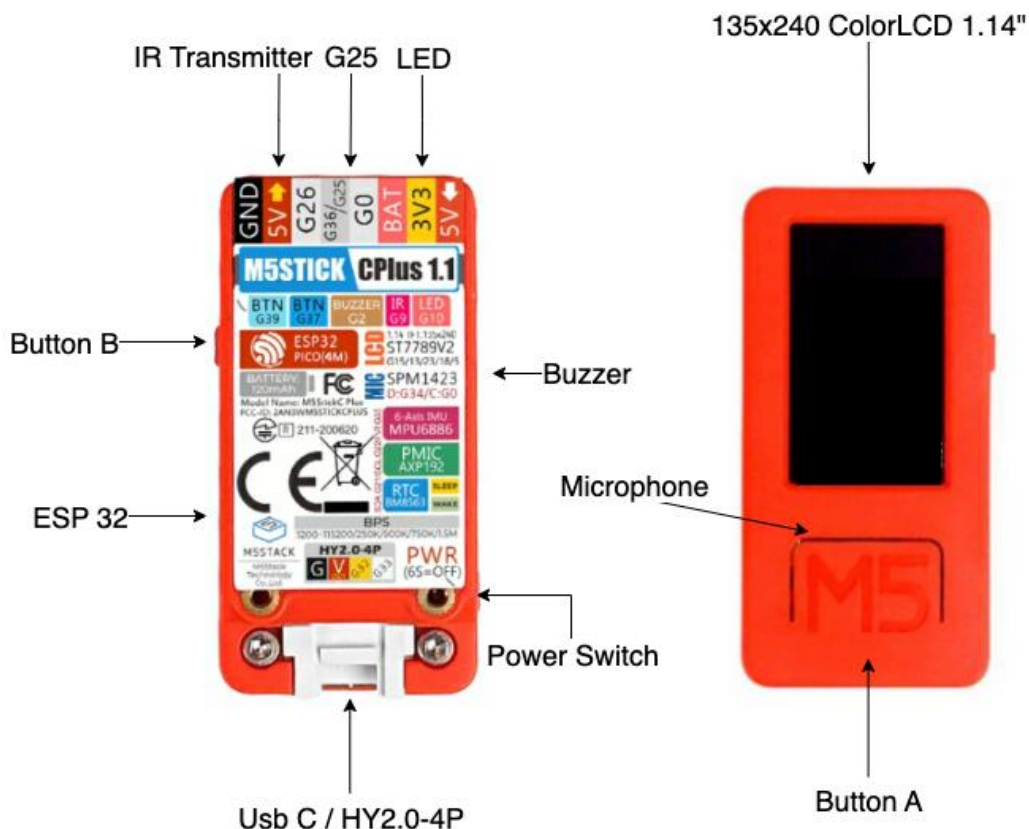


Ilustración 23 M5Stick CPlus

## 5 Aspectos relevantes del desarrollo

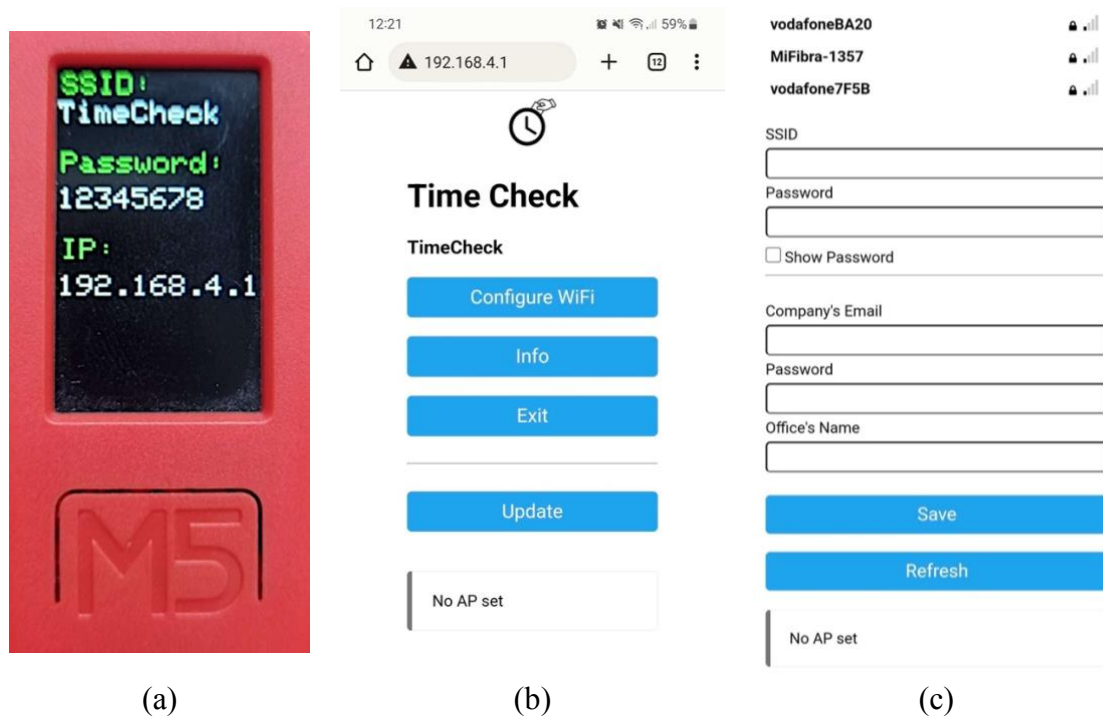


Ilustración 24 Configuración M5

El esquema de funcionamiento es tal que, el dispositivo muestra una clave en forma de código QR, esta clave está almacenada bajo la identidad de la empresa y del propio dispositivo en RealTime Database. Cuando el empleado lea el código QR con su aplicación, una *Cloud Function*, validará la clave, actualizando el estado de trabajo del empleado y marcando esa clave como no disponible. Este cambio de disponibilidad de la clave será detectado por el dispositivo, generando una respuesta visual, acústica y una nueva clave. Este esquema se puede corroborar en la Ilustración 25.

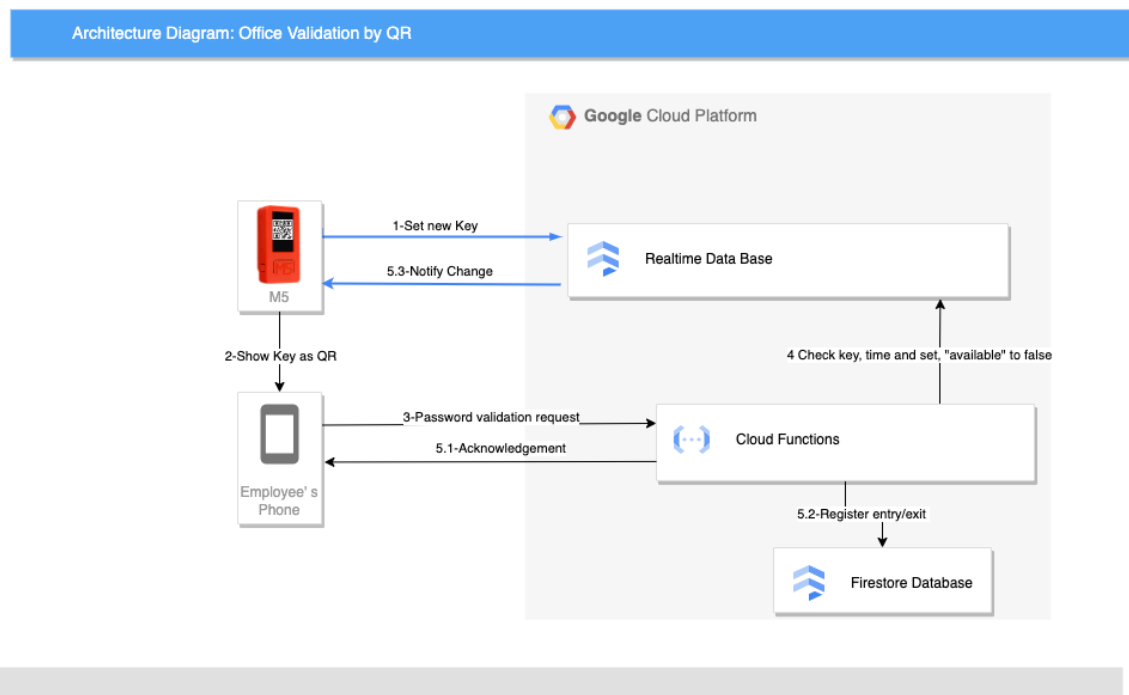


Ilustración 25 Validación Mediante M5

## 5.7 Presentación de la aplicación

Para esta aplicación se ha diseñado un logo formado por un reloj y una mano que evoca la antigua forma de fichar mediante tarjetas como se muestra en la Ilustración 26.



*Ilustración 26 Registrador de Tiempo Analógico [50]*

El logo se encuentra en una versión clara y otra oscura para poder adaptarse a los temas de las interfaces (Ilustración 27, Ilustración 28).



*Ilustración 27 Logo Oscuro*



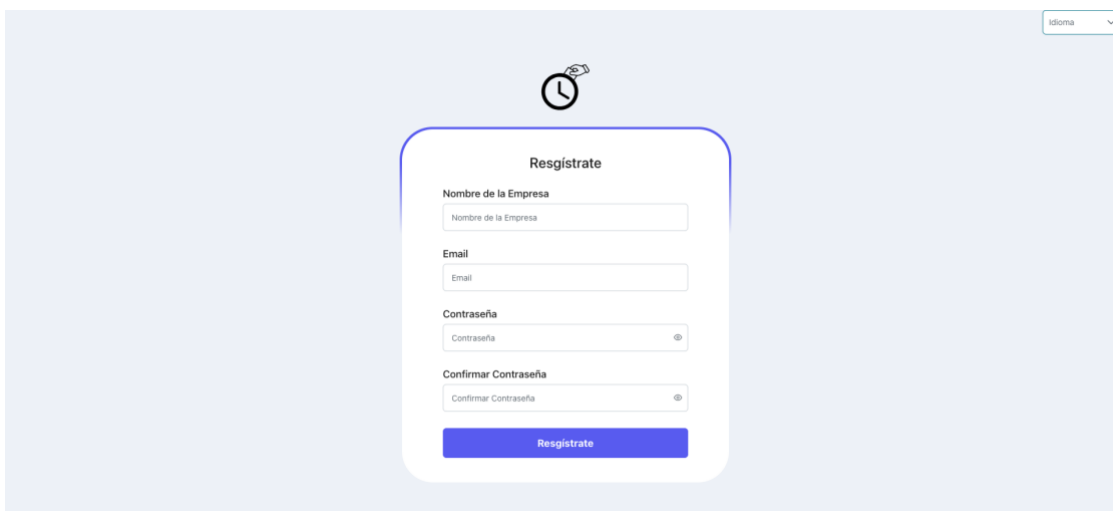
*Ilustración 28 Logo Claro*

## 5 Aspectos relevantes del desarrollo

Se puede acceder a la aplicación web desde cualquier navegador web y dispositivo, aunque se recomienda su uso a los administradores desde un ordenador o dispositivo de gran pantalla para poder visualizar mejor los datos y acceder desde Google Chrome para poder instalarla. Las opciones y vistas de la aplicación, se encuentran de forma más detallada en el manual de usuario, Anexo V. En la página de inicio: <https://timecheck.page/#/landing>, podrá encontrar un vídeo de demostración.

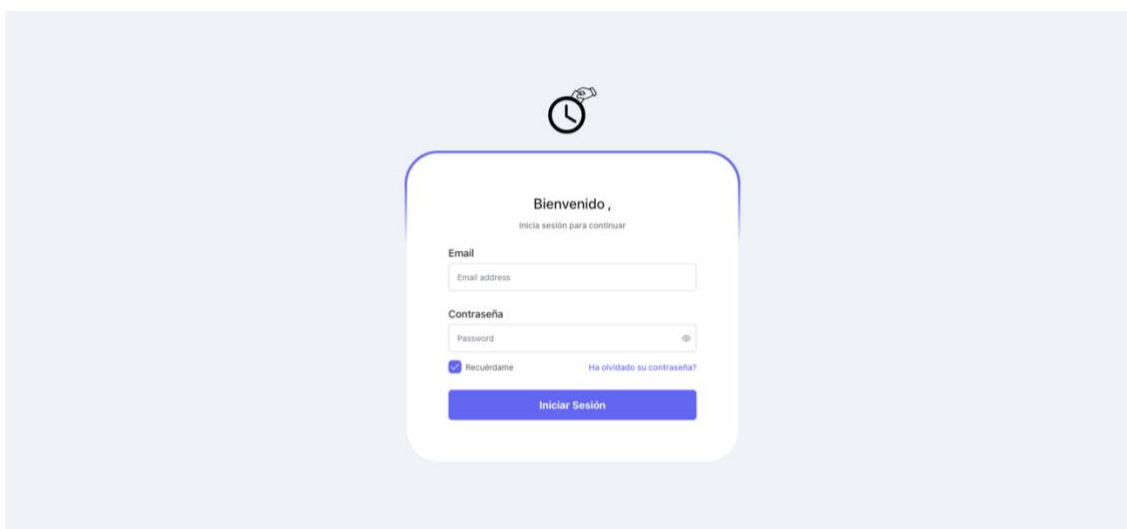
El inicio de sesión de la aplicación es igual para todos los usuarios. Una vez se validen los datos de este, se dirigirá a una interfaz u otra en función del rol.

Las vistas de registro (Ilustración 29), inicio de sesión (Ilustración 30) y validar email (Ilustración 31) se muestran en las siguientes ilustraciones:



The screenshot shows a registration form titled "Regístrate" centered on a light blue background. At the top center is a clock icon with a checkmark. In the top right corner, there is a small "Idioma" dropdown menu. The form itself is a white rounded rectangle with a blue border. It contains four input fields: "Nombre de la Empresa", "Email", "Contraseña", and "Confirmar Contraseña". Each field has a small icon to its right. Below the fields is a blue button labeled "Regístrate".

*Ilustración 29 Registrarse*



The screenshot shows a login form titled "Bienvenido," centered on a light blue background. At the top center is a clock icon with a checkmark. The form is a white rounded rectangle with a blue border. It contains two input fields: "Email" and "Contraseña". Below the "Contraseña" field, there is a checked checkbox labeled "Recuérdame" and a link that says "¿Ha olvidado su contraseña?". At the bottom is a blue button labeled "Iniciar Sesión".

*Ilustración 30 Iniciar Sesión*

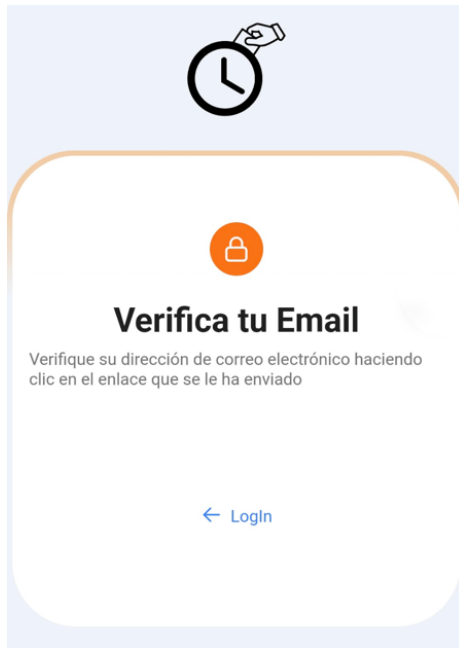


Ilustración 31 Verificar Email

La aplicación cuenta con un menú de configuración donde se pueden ajustar ciertos aspectos de la interfaz como el idioma, tamaño o modo oscuro como se puede observar en la Ilustración 32.

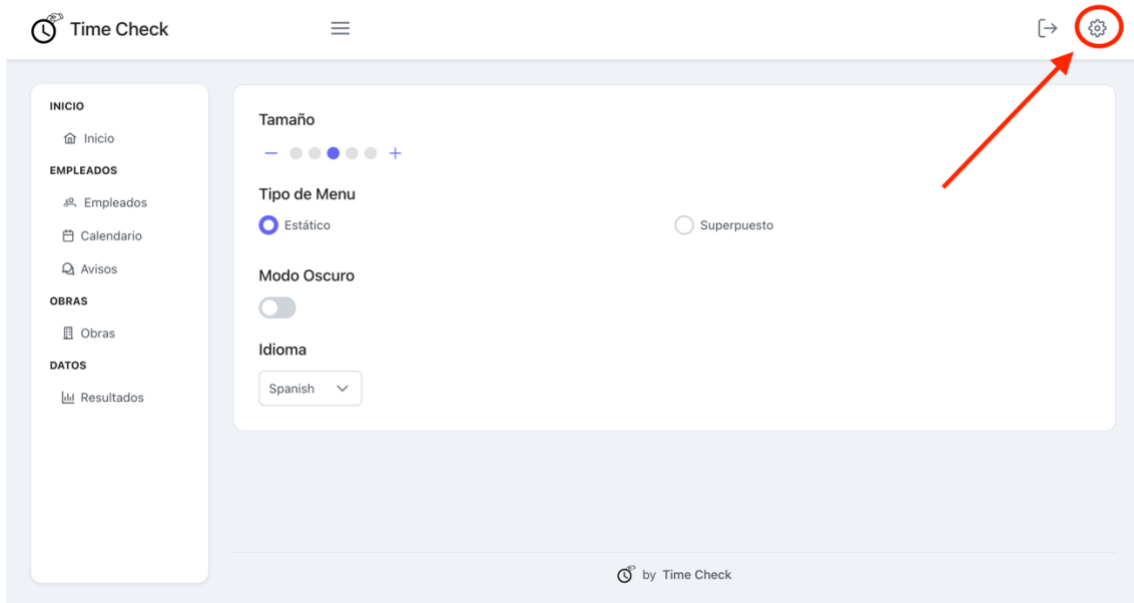


Ilustración 32 Menú Configuración

## 5 Aspectos relevantes del desarrollo

### 5.7.1 Interfaces del administrador

Toda la interfaz se estructura en tarjetas que contienen componentes, como se muestra en la ventana de inicio del administrador en la Ilustración 33. En esta interfaz, el administrador de la empresa puede visualizar de forma rápida información de la empresa como el número de empleados totales y activos o el porcentaje de empleados para cada rol. El apartado más importante de esta vista es el primero, que muestra una tabla con la información actual de la empresa, es decir, que empleados han asistido, cuales han marcado una ausencia y cuales todavía no han marcado nada.

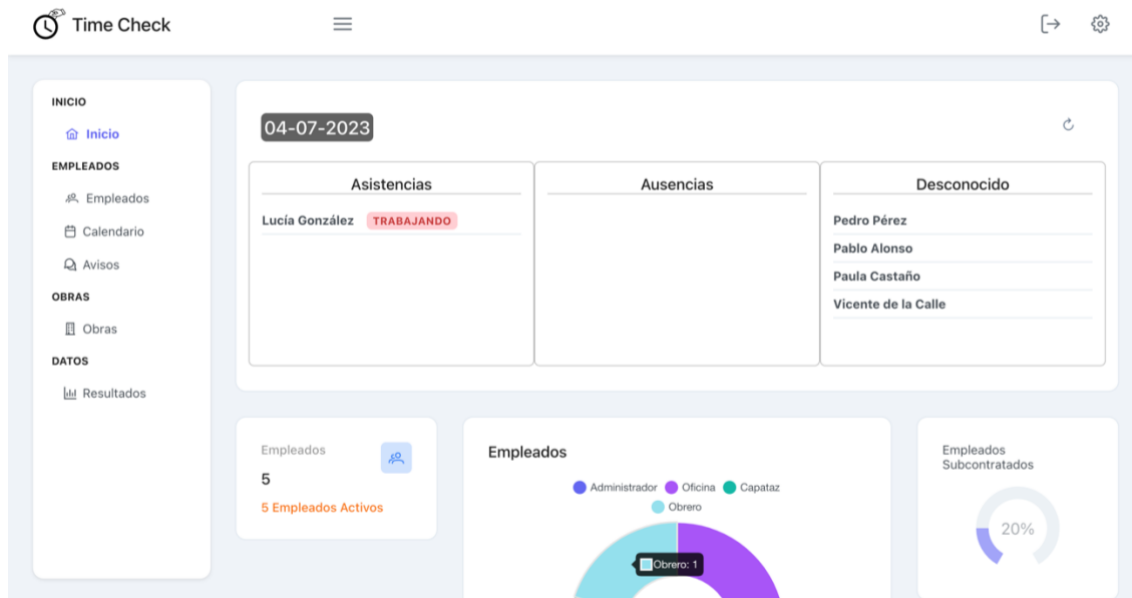


Ilustración 33 Ventana Inicio Administrador

Continuando con el administrador, cuenta con dos interfaces, las mostradas en la Ilustración 34 e Ilustración 35, tipo CRUD (*Create, Read, Update, Delete*) con las que gestionar las obras y los empleados. En esta interfaz, es en la que podrá registrar a nuevos empleados para su empresa en el sistema, de forma instantánea, estos podrán ya acceder a su perfil desde su usuario en la aplicación. También se pueden exportar las identificaciones de los usuarios en formato PDF por si se desean hacer tarjetas de acceso o similares con ellas, para los obreros.

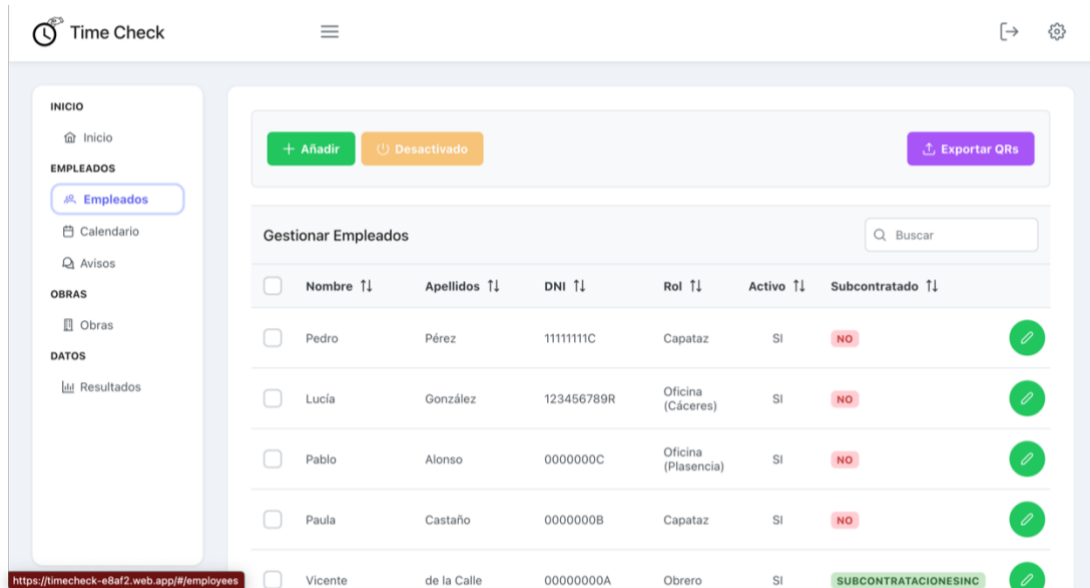


Ilustración 34 Empleados Administrador (nombres ficticios)

Tanto para las obras como para los empleados con rol de oficina, hay que especificar su localización para que cuando el algoritmo de chequear las ausencias se ejecute, pueda asociar los festivos locales al lugar de trabajo.

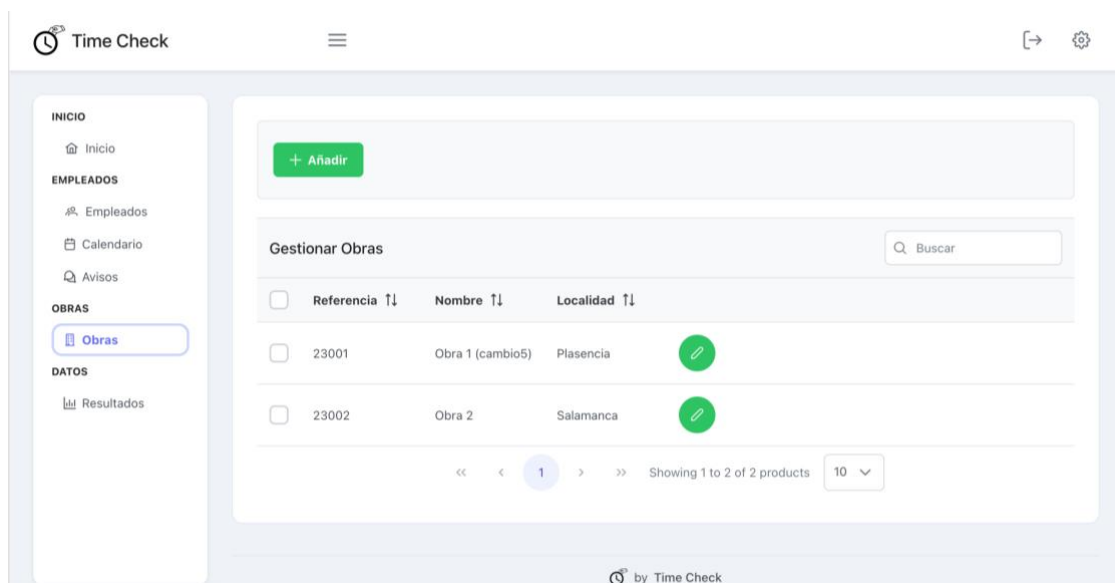


Ilustración 35 Obras Administrador

## 5 Aspectos relevantes del desarrollo

El calendario de trabajo es una pieza fundamental de este proyecto ya que permitirá marcar ausencias con anterioridad. Pueden afectar a una lista de empleados determinados como se ve en la Ilustración 36 o a una ubicación o región como se ve en Ilustración 37. Los festivos nacionales y autonómicos son obtenidos mediante la API (del inglés, *application programming interface*, en español, interfaz de programación de aplicaciones) Nager.Date [51], la primera vez que se accede a un año del calendario, luego son almacenados junto con el resto de los datos de la empresa para que el administrador pueda editarlos.

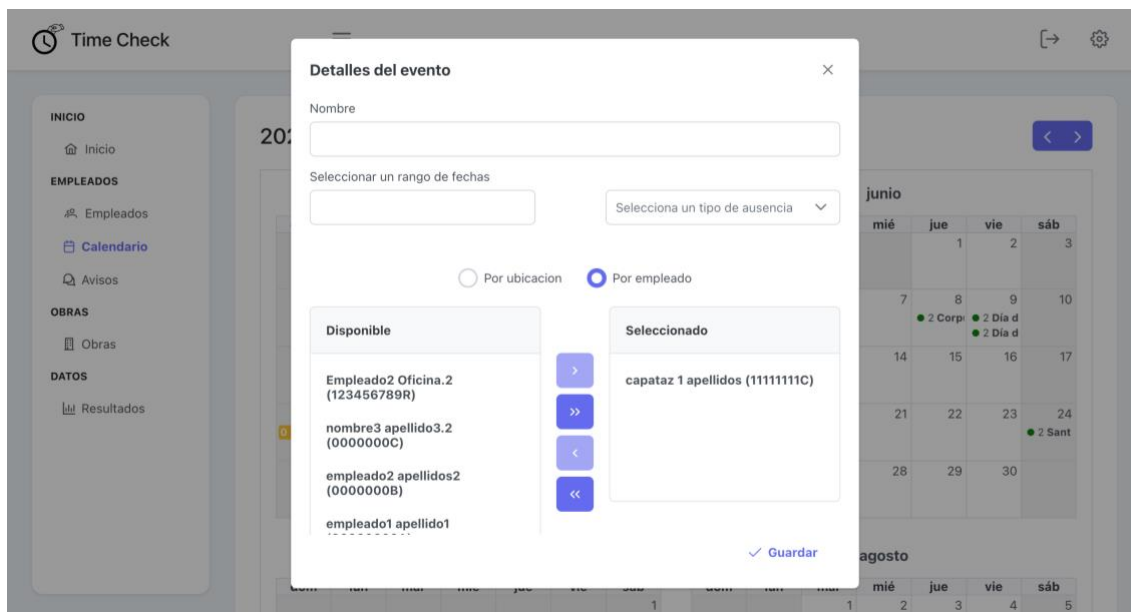


Ilustración 36 Evento Empleado

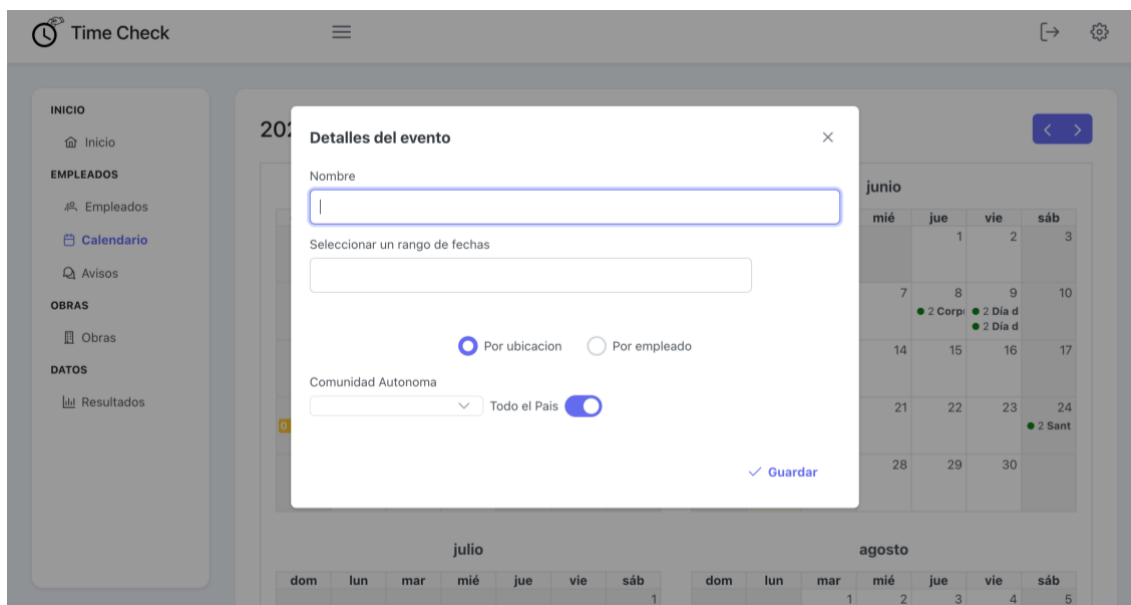


Ilustración 37 Evento Ubicación



Los eventos se muestran en el calendario de trabajo con una referencia de colores:

- Amarillo: festivos nacionales.
- Verde: festivos autonómicos.
- Morado: festivos asociados a empleados.

Y pueden abarcar desde uno a varios días como se ve en la Ilustración 38.

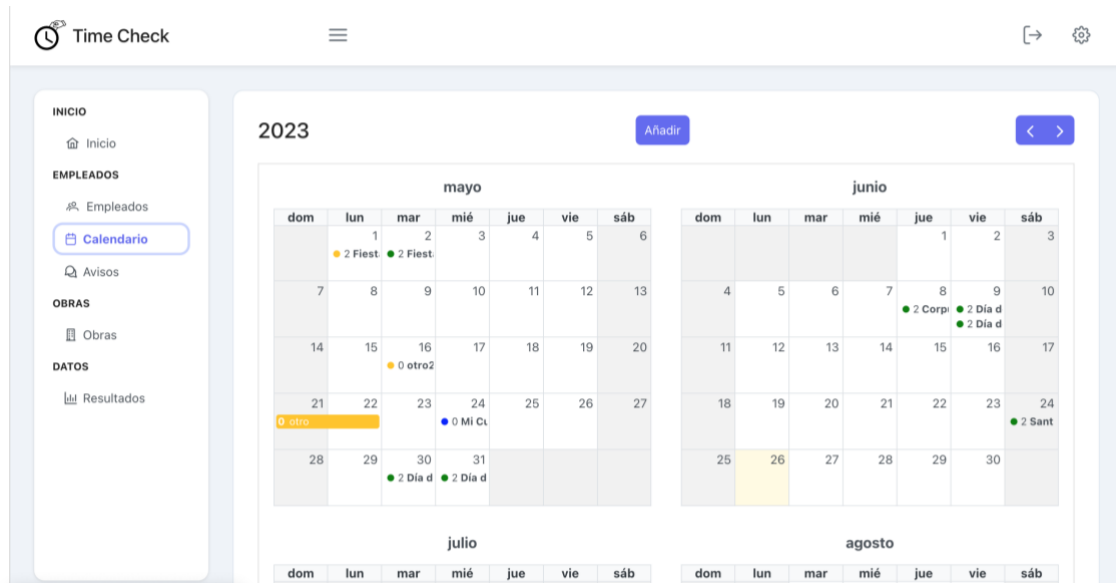
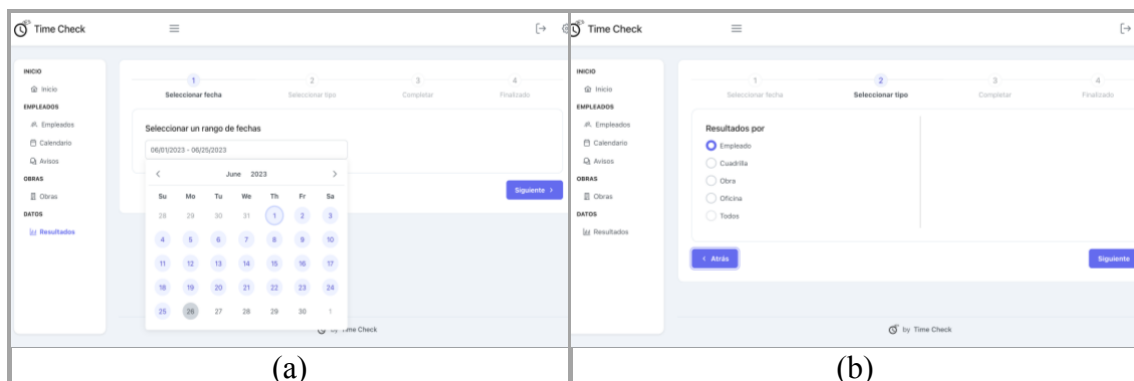


Ilustración 38 Calendario de Trabajo

Otra parte importante del sistema, son los resultados. Estos se obtienen mediante los registros de asistencia almacenados en la base de datos. Para poder obtenerlos, se debe de seguir el menú por pasos (*wizard*) de la Ilustración 39.

Primero habrá que seleccionar las fechas entre las que se quieren obtener los resultados (Ilustración 39, a) y después el tipo de resultado (Ilustración 39, b). En el caso de que se requiera especificar algún detalle más como, el nombre del empleado o de la obra, se completará en el tercer paso (Ilustración 39, c). Una vez finalizado (Ilustración 39, d), permite también su exportación a un fichero tipo Excel (Ilustración 39, e) para que la empresa pueda realizar sus propios análisis de datos o incluso presentarlos ante inspección de trabajo si fuese necesario ya que son perfectamente válidos como se explica en 3.3 Ley de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo.



## 5 Aspectos relevantes del desarrollo

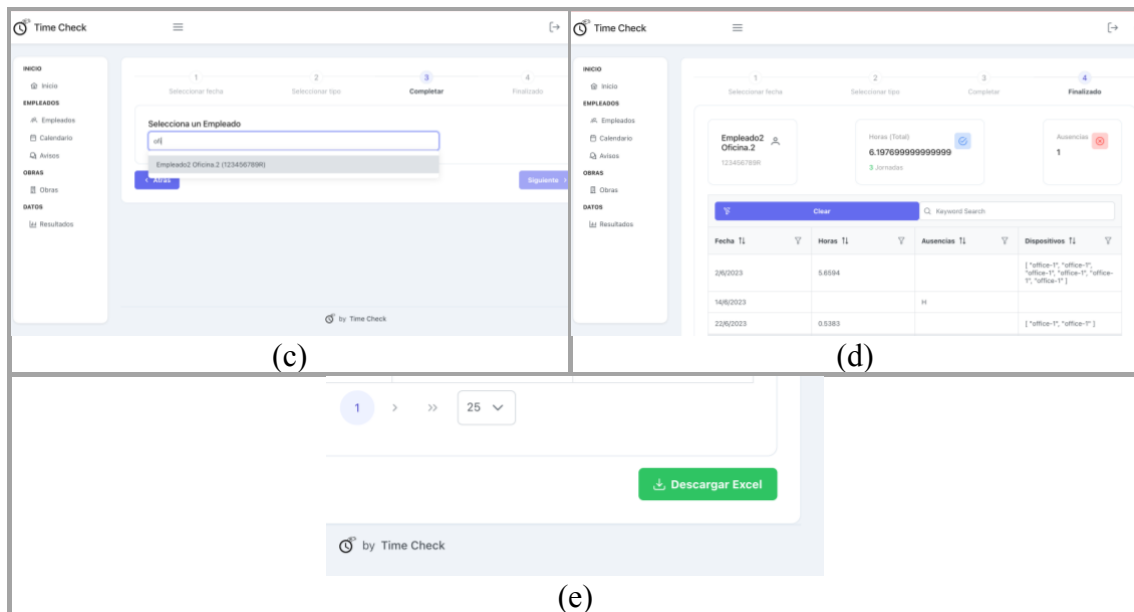


Ilustración 39 Resultados

Por último, por parte del administrador, nos encontramos con la interfaz de avisos (Ilustración 40), mediante la que se pueden enviar notificaciones *push* a los empleados aprovechando así una de las ventajas de las PWA.

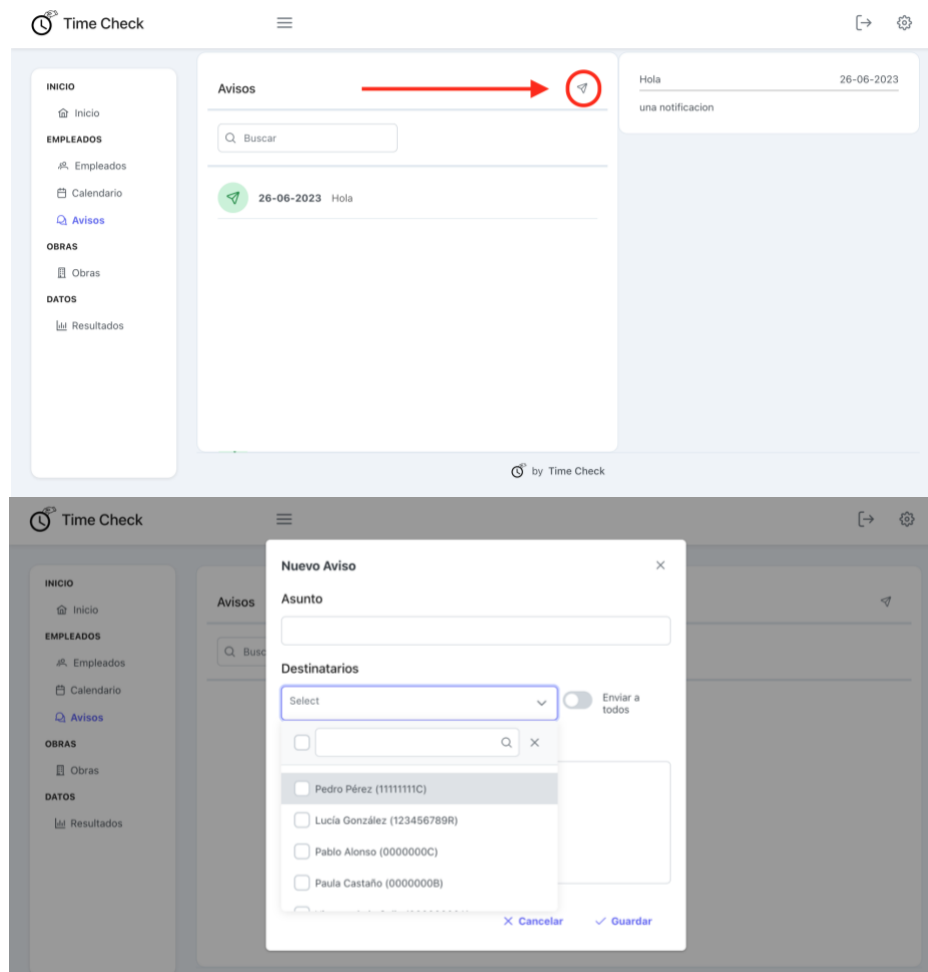


Ilustración 40 Avisos

### 5.7.1 Interfaces del empleado

Las interfaces del empleado están principalmente dirigidas a ser visualizadas desde smartphones tanto Android como iPhone. Son interfaces más sencillas ya que no requieren de tantas funcionalidades y están destinadas a usuarios con menores conocimientos sobre tecnología. Además, en este apartado es donde más se valora la elección de interfaces grandes y paleta de color con contrastes altos, explicadas al final del Anexo III, debido a que los obreros y capataces, normalmente trabajan en el exterior donde las pantallas pueden llegar a ser difíciles de visualizar por la luz del sol.

Para medir los contrastes entre colores y elegir los más adecuados, se han usado herramientas como Adobe Color [39], Ilustración 41.

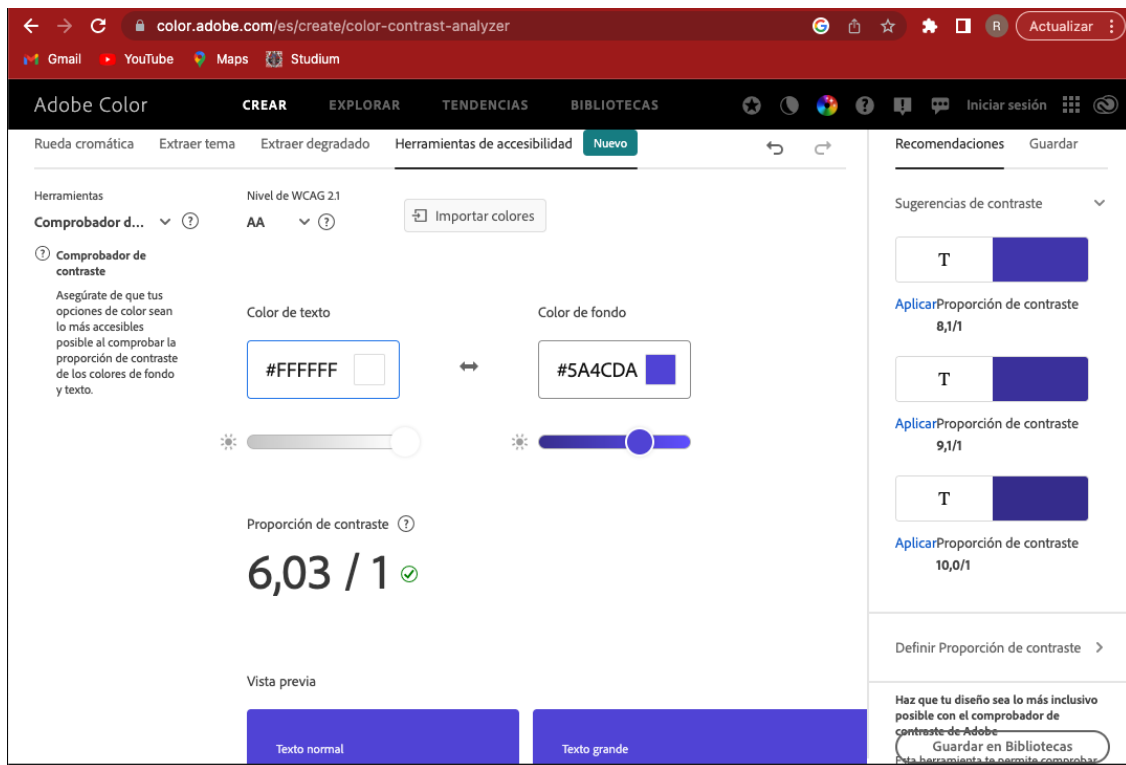


Ilustración 41 Adobe Color

Comenzando con el empleado tipo oficina. Cuanta con una vista muy sencilla donde simplemente podrá escanear el código del dispositivo de acceso o marcar una ausencia como se puede ver en la Ilustración 42. En ella podrá ver su tiempo total trabajado ese día, su estado actual y las horas de las entradas y salidas que ha realizado. Para ahorrar recursos, no se consideran los tiempos de trabajo o descanso de menos de 5 minutos.

## 5 Aspectos relevantes del desarrollo

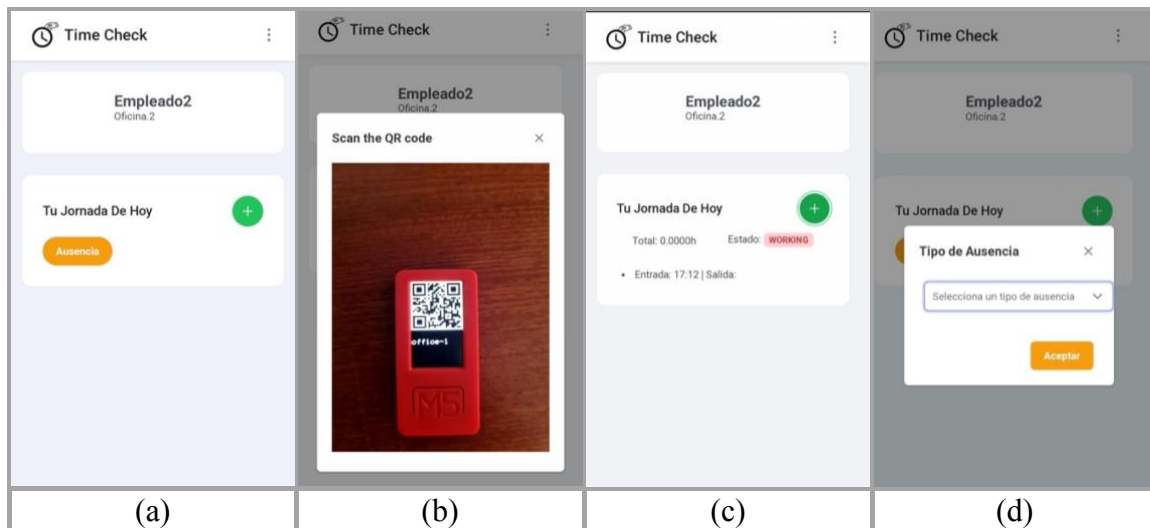


Ilustración 42 Oficina

Por otro lado, los capataces cuentan con una interfaz más compleja, ya que tienen que llevar el recuento de todos sus obreros a cargo.

En su perfil les aparecerá una tarjeta donde deben de introducir la referencia de la obra en la que van a trabajar (Ilustración 43, a), una vez introducida, podrán “pasar lista” a sus empleados mediante el escaneo de sus identificaciones (Ilustración 43, b) o de forma manual (Ilustración 43, c), pensado sobre todo para las ausencias.

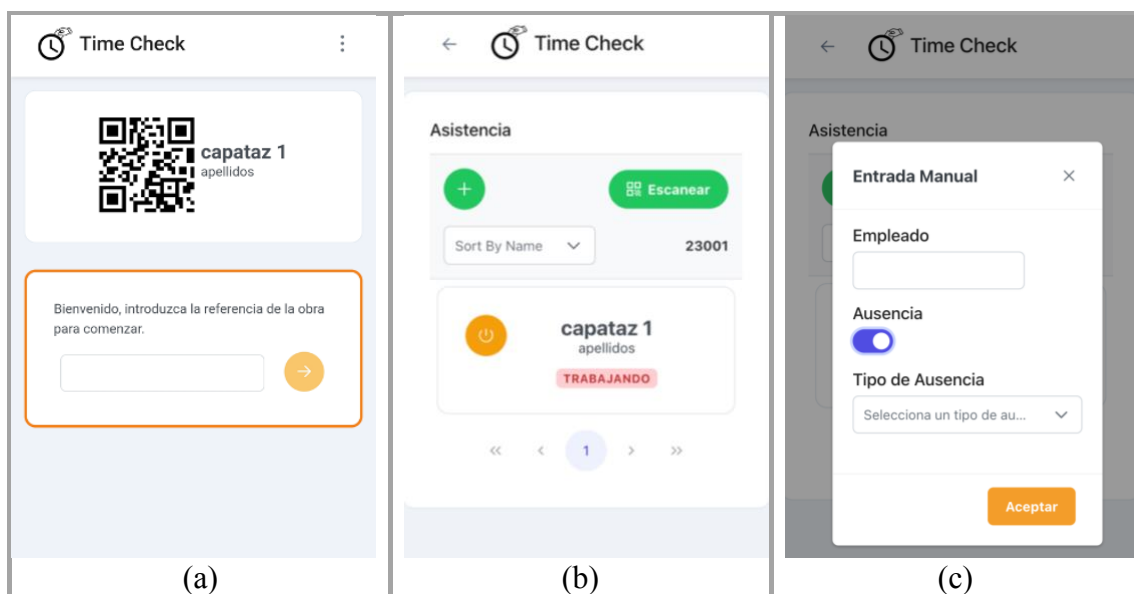


Ilustración 43 Capataz

## 6 Conclusiones y líneas de trabajo futuras

### 6.1 Conclusiones

Tras el desarrollo de este proyecto, se puede afirmar que se han cumplido los objetivos y requisitos propuestos inicialmente:

- Se ha conseguido la creación software capaz de gestionar las entradas y salidas de empleados pese a las peculiaridades de la industria de la construcción.
- Se ha conseguido crear una aplicación web mediante la cual las empresas puedan gestionar a sus empleados y los empleados puedan fichar en su puesto de trabajo mediante distinción de roles.
- Se ha conseguido implementar una base de datos eficiente y segura para administrar todos los registros de asistencia y usuarios.
- Se ha conseguido desarrollar un dispositivo para poder fichar en sitios de trabajo fijo como oficinas, que sea de bajo coste y además que pueda servir como base para implementarlo a otros dispositivos para así adaptarse a nuevas situaciones.
- Se ha conseguido implementar un sistema de identificación mediante códigos QR.
- Se ha conseguido implementar un procesamiento de datos eficiente y útil para las empresas de cara a tareas como inspección laboral o análisis internos.

Aunque las pruebas de este sistema en un entorno real de explotación todavía están en una fase temprana, se espera tanto por parte del desarrollador como por parte de la empresa voluntaria, que este sistema sirva de gran ayuda, ahorrando tiempo de gestión de hojas de firmas de asistencia que antes tenían que ser transformadas a Excel de forma manual además de aportar un significativo mayor grado de fiabilidad y control de estos datos. Todo ello con una inversión mínima para la empresa y además pudiéndose aplicar a otras tantas de este sector.

A nivel personal, este proyecto ha supuesto un alto grado de esfuerzo e implicación, dando un resultado con el que estoy gratamente satisfecho. Mediante este proyecto he tenido podido aplicar lo aprendido durante casi todas las disciplinas del Grado en Ingeniería Informática: Diseño y Estructura de Bases De Datos, Estructuras De Datos y Algoritmos, Interfaces Gráficas de Usuario, Ingeniería De Software, Gestión De Proyectos, Interacción Persona-Ordenador, Programación, entre otras.

Asimismo, he tenido que investigar y ser autodidacta en otros muchos puntos que quedan fuera del ámbito del Grado, pero para lo que también nos preparan durante él. He conseguido adquirir nuevos conocimientos sobre VueJS, JavaScript, CSS, HTML, Arduino, servicios de Google Cloud, IoT (*Internet of Things*), PWA (*Progressive Web App*). Me ha brindado la oportunidad de presentar este proyecto al plan de prototipos TCUE [7] mediante el cual he podido adquirir nociones sobre la parte más empresarial y comercial del desarrollo software.

Pero, sobre todo, por haber aprovechado esta oportunidad para realizar un proyecto que no solo quede forma teórica, sino que sea realmente útil y aplicable en el mundo real. Es decir, cumplir el propósito como ingeniero de buscar soluciones viables y eficientes a problemas reales.

### 6.2 Líneas de trabajo futuras

Este proyecto supone únicamente el inicio de un sistema cuya funcionalidad puede extenderse mucho más allá del ámbito del control horario. Entre otras posibles funcionalidades que se han planteado durante el desarrollo destacan:

- Añadir gestión de costes de recursos
- Mejorar la identificación mediante QRs dinámicos con sistemas como identificación por audio [52] o NFC.
- Aplicar sistemas de Inteligencia Artificial para obtener automáticamente, información relevante sobre todos los datos almacenados como la eficiencia de los trabajadores o incluso predicción de ausencias o bajas.

Tras esto, también se podrían ir añadiendo otros servicios útiles para la gestión de recursos humanos y materiales de las empresas de este sector como sistemas de planificación o gestión de inventario y almacenes entre otros.

## Referencias

- [1] Jefatura del Estado, “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.”
- [2] U. Farooq, M. ul Hasan, M. Amar, A. Hanif, and M. U. Asad, “RFID based security and access control system,” *International Journal of Engineering and Technology*, vol. 6, no. 4, p. 309, 2014.
- [3] Y. Mittal, A. Varshney, P. Aggarwal, K. Matani, and V. K. Mittal, “Fingerprint biometric based access control and classroom attendance management system,” in *2015 Annual IEEE India Conference (INDICON)*, IEEE, 2015, pp. 1–6.
- [4] S. Wu, A. Sheth, J. Miller, and Z. Luo, “Authorization and access control of application data in workflow systems,” *J Intell Inf Syst*, vol. 18, pp. 71–94, 2002.
- [5] Unión Europea, “Agenda 2030”, Accessed: Jul. 02, 2023. [Online]. Available: [https://commission.europa.eu/strategy-and-policy/international-strategies/sustainable-development-goals/eu-and-united-nations-common-goals-sustainable-future\\_es](https://commission.europa.eu/strategy-and-policy/international-strategies/sustainable-development-goals/eu-and-united-nations-common-goals-sustainable-future_es)
- [6] Naciones Unidas, *Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación*. Accessed: Jul. 03, 2023. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/infrastructure/>
- [7] Universidad de Salamanca, “Prototipos TCUE.” <https://tcue.usal.es/prototipos-orientados-al-mercado> (accessed Jul. 02, 2023).
- [8] “Control de Accesos RFID.” <https://www.segurilatam.com/wp-content/uploads/sites/5/2020/08/control-de-accesos-tarjeta-rfid.jpg> (accessed Jul. 05, 2023).
- [9] N. B. Sukhai, “Access control & biometrics,” in *Proceedings of the 1st annual conference on Information security curriculum development*, 2004, pp. 124–127.
- [10] B. Cui and T. Xue, “Design and realization of an intelligent access control system based on voice recognition,” in *2009 ISECS international colloquium on computing, communication, control, and management*, IEEE, 2009, pp. 229–232.
- [11] A. P. Olimid, L. M. Rogozea, and D. A. Olimid, “Ethical approach to the genetic, biometric and health data protection and processing in the new EU General Data Protection Regulation (2018),” *Rom J Morphol Embryol*, vol. 59, no. 2, pp. 631–636, 2018, [Online]. Available: <http://europepmc.org/abstract/MED/30173275>
- [12] Grupo Spec, “INFORMES DE CONTROL HORARIO VÁLIDOS PARA LA INSPECCIÓN DE TRABAJO.” <https://grupospec.com/blog/informes-de-control-horario-validos-para-la-inspeccion-de-trabajo/> (accessed Jul. 02, 2023).
- [13] hrlog, “Plantilla de excel para el registro horario.” <https://hrlog.es/wp-content/uploads/2020/07/excel-cabecera.png> (accessed Jul. 05, 2023).

## Referencias

- [14] Iván Ramírez, “¿Qué es una Aplicación Web Progresiva o PWA?” <https://www.xataka.com/basics/que-es-una-aplicacion-web-progresiva-o-pwa>.
- [15] Google for Developers, “Referencia de Firebase CLI.” Accessed: May 27, 2023. [Online]. Available: <https://firebase.google.com/docs/cli?hl=es-419>
- [16] Google, “Cloud Firestore.” Accessed: Jun. 21, 2023. [Online]. Available: <https://firebase.google.com/docs/firestore?hl=es-419>
- [17] Google, “Firebase Hosting”, Accessed: Jun. 21, 2023. [Online]. Available: <https://firebase.google.com/docs/hosting?hl=es-419>
- [18] E. Saks, “JavaScript Frameworks: Angular vs React vs Vue.,” 2019.
- [19] R. de Arriba Garcia, “Estudio de la popularidad del framework VueJS”.
- [20] Primefaces, “Sakai Vue.” Accessed: Jun. 29, 2023. [Online]. Available: <https://github.com/primefaces/sakai-vue>
- [21] PrimeTek, “Primefaces.” <https://www.primefaces.org/> (accessed Jun. 29, 2023).
- [22] PrimeVue, “PrimeVue.” Accessed: Jul. 03, 2023. [Online]. Available: <https://primevue.org>
- [23] H. Eisfeld and F. Kristallovich, “The rise of dark mode: A qualitative study of an emerging user interface design trend.” 2020.
- [24] B. Syed, *Beginning Node.js*. Apress, 2014.
- [25] D. B. Duldulao and R. J. L. Cabagnet, “Getting Started with the Node Package Manager,” in *Practical Enterprise React: Become an Effective React Developer in Your Team*, Springer, 2021, pp. 11–19.
- [26] “FullCalendar.” <https://fullcalendar.io> (accessed Jun. 29, 2023).
- [27] LuisEnMarroquin, “json-as-xlsx.” Accessed: Jul. 03, 2023. [Online]. Available: <https://www.npmjs.com/package/json-as-xlsx>
- [28] “ApexCharts.” <https://apexcharts.com> (accessed Jun. 29, 2023).
- [29] scholtz, “qrcode-vue3.” Accessed: Jul. 03, 2023. [Online]. Available: <https://www.npmjs.com/package/qrcode-vue3>
- [30] storozhok, “vue3-qr-reader.” Accessed: Jul. 03, 2023. [Online]. Available: <https://www.npmjs.com/package/@storozhok/vue3-qr-reader>
- [31] MrRio, “jspdf.” Accessed: Jul. 03, 2023. [Online]. Available: <https://www.npmjs.com/package/jspdf>
- [32] kazupon, “I18n.” Accessed: Jul. 03, 2023. [Online]. Available: <https://github.com/intlify/vue-i18n-next>



- [33] Google, “Firebase Auth.” Accessed: Jun. 21, 2023. [Online]. Available: <https://firebase.google.com/docs/auth?hl=es-419>
- [34] Google, “RealTime Database.” Accessed: Jun. 21, 2023. [Online]. Available: <https://firebase.google.com/docs/firestore?hl=es-419>
- [35] Google, “Firebase Functions”, Accessed: Jun. 21, 2023. [Online]. Available: <https://firebase.google.com/docs/functions?hl=es-419>
- [36] Arduino, “Arduino IDE.” Accessed: May 27, 2023. [Online]. Available: <https://www.arduino.cc/en/software>
- [37] A. Kurniawan, *Internet of Things Projects with ESP32: Build exciting and powerful IoT projects using the all-new Espressif ESP32*. Packt Publishing Ltd, 2019.
- [38] tzapu, “WiFiManager.” Accessed: Jul. 02, 2023. [Online]. Available: <https://github.com/tzapu/WiFiManager>
- [39] mobizt, “Firebase-ESP32.” Accessed: Jul. 02, 2023. [Online]. Available: <https://github.com/mobizt/Firebase-ESP32>
- [40] “Adobe Color.” <https://color.adobe.com/es/create/color-wheel> (accessed Jun. 21, 2023).
- [41] “Canva.” Accessed: Jun. 21, 2023. [Online]. Available: [https://www.canva.com/es\\_es/](https://www.canva.com/es_es/)
- [42] Microsoft, “Microsoft Project.” Accessed: May 28, 2023. [Online]. Available: <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- [43] C. A. Remón and P. J. Thomas, “Análisis de estimación de esfuerzo aplicando puntos de caso de uso,” in *XVI Congreso Argentino de Ciencias de la Computación*, 2010.
- [44] Visual Paradigm, “Visual Paradigm.” Accessed: Jun. 29, 2023. [Online]. Available: <https://www.visual-paradigm.com/>
- [45] “PlantUML”, Accessed: Jun. 29, 2023. [Online]. Available: <https://plantuml.com/es/>
- [46] I. Jacobson, G. Booch, and J. Rumbaugh, *El proceso unificado de desarrollo de software*, vol. 7. Addison Wesley Madrid, 2000.
- [47] “Proceso Unificado Diagrama.” <http://informatica-uitl.blogspot.com/2013/03/proceso-unificado-de-desarrollo.html> (accessed Jun. 29, 2023).
- [48] A. Durán Toro and B. Bernárdez Jiménez, “Metodología para la Elicitación de Requisitos de Sistemas Software, versión 2.1, Informe Técnico LSI–2000–10,” 2000.

## Referencias

- [49] Anton Zuev, “Patrón MVVM en SwiftUI,” 2020. <https://www.adictosaltrabajo.com/2020/06/05/patron-mvvm-en-swiftui/> (accessed Jun. 16, 2023).
- [50] Eduardo San Martin Morote, “Pinia.” Accessed: Jul. 03, 2023. [Online]. Available: <https://pinia.vuejs.org>
- [51] “Reloj de Tiempo Analógico.” [https://img.fruugo.com/product/9/39/813469399\\_max.jpg](https://img.fruugo.com/product/9/39/813469399_max.jpg) (accessed Jun. 29, 2023).
- [52] nager.at, “Nager.Date.” <https://date.nager.at> (accessed Jul. 02, 2023).
- [53] S. Wu, J. Huang, D. Huang, and Y. Q. Shi, “Efficiently self-synchronized audio watermarking for assured audio data transmission,” *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 69–76, 2005.