

**UNIR FP**

**Informática y Comunicaciones**

Técnico Superior en DAM

Sistema de fichajes

|  |  |
| --- | --- |
| Trabajo fin de estudio presentado por: | Laguna Rebolo, Alejandro  Estrada Amat, Javier  Garijo Blanco, Antonio |
| Tipo de trabajo: | Sistema de Fichajes |
| Tutor/a: | Damián Sualdea Soy |
| Fecha: | 30/03/2025 |

Resumen

Este Trabajo de Fin de Grado consiste en el desarrollo de un sistema de fichajes para empleados, destinado a facilitar y automatizar el control de la jornada laboral en empresas, cumpliendo con la normativa vigente en materia de registro horario. El sistema está basado en una arquitectura cliente-servidor, con un backend desarrollado en **Spring Boot** utilizando **JPA**, **Lombok**, y controladores **REST**, y un frontend implementado en **Angular**. La base de datos utilizada es **MySQL**, y todo el sistema se organiza siguiendo buenas prácticas de desarrollo, con capas diferenciadas.

El sistema contempla distintos roles de usuario: empresa, dirección y empleado, cada uno con permisos específicos. Entre sus funcionalidades principales se encuentran el **registro de fichajes**, la **gestión de empleados**, la **validación de horas extra** y la **generación de informes**. Además, se ha diseñado un sistema de autenticación segura y escalable, con la posibilidad de integrar autenticación en dos pasos (2FA) con herramientas como Google Authenticator.

Este proyecto demuestra competencias en el desarrollo de aplicaciones web modernas, gestión de base de datos, implementación de sistemas seguros y en la aplicación de patrones de diseño y arquitecturas robustas para entornos empresariales.

[javier-estrada-amat/TFG-DAM: Este es el proyecto del TFG de DAM](https://github.com/javier-estrada-amat/TFG-DAM/)

**Palabras clave:** (Máximo 5 palabras)

Abstract

This Final Degree Project consists of the development of an employee time tracking system designed to facilitate and automate the control of working hours within companies, in compliance with current labor regulations regarding time registration. The system is based on a client-server architecture, with a backend developed in **Spring Boot** using **JPA**, **Lombok**, and **REST** controllers, and a frontend built with **Angular**. The database used is **MySQL**, and the entire system follows best development practices, with a layered structure.

The system includes different user roles: company, management, and employee, each with specific permissions. Its main features include **time clocking (check-in/out)**, **employee management**, **overtime validation**, and **report generation**. Additionally, a secure and scalable authentication system has been implemented, with the possibility of integrating two-factor authentication (2FA) using tools such as Google Authenticator.

This project demonstrates proficiency in modern web application development, database management, secure system implementation, and the application of design patterns and robust architectures suited for enterprise environments.

[javier-estrada-amat/TFG-DAM: Este es el proyecto del TFG de DAM](https://github.com/javier-estrada-amat/TFG-DAM/)

**Keywords:** (5 words max)

Índice de contenidos

[1. Introducción 9](#_Toc180827987)

[1.1. Justificación 9](#_Toc180827988)

[1.2. Objetivos 9](#_Toc180827989)

[2. Módulos formativos aplicados en el trabajo 10](#_Toc180827990)

[3. Herramientas y lenguajes utilizados 11](#_Toc180827991)

[4. Metodologías utilizadas 12](#_Toc180827992)

[5. Componentes del equipo y aportaciones realizadas por cada alumno 13](#_Toc180827993)

[5.1. Estudio de mercado 13](#_Toc180827994)

[5.1.1. Tabla comparativa de aplicaciones actualmente en el mercado 13](#_Toc180827995)

[5.1.2. Análisis DAFO de nuestra solución 14](#_Toc180827996)

[5.1.3. Subapartado 1.2 14](#_Toc180827997)

[5.2. Modelo de datos 15](#_Toc180827998)

[5.3. Diagramas UML 15](#_Toc180827999)

[5.3.1. Diagrama de clases 15](#_Toc180828000)

[5.3.2. Clasificación de usuarios 16](#_Toc180828001)

[5.3.3. Caso de uso 01 16](#_Toc180828002)

[5.4. Diseño de interfaces 17](#_Toc180828003)

[5.4.1. Wireframes 17](#_Toc180828004)

[5.4.2. Prototipo de interfaz de alta definición 18](#_Toc180828005)

[5.4.3. Paleta de colores 18](#_Toc180828006)

[5.4.4. Logotipo 18](#_Toc180828007)

[5.5. Planificación temporal y trabajo en equipo 19](#_Toc180828008)

[5.5.1. Presupuesto temporal de tareas 19](#_Toc180828009)

[5.5.2. Organización de tareas y tiempos finales 20](#_Toc180828010)

[5.5.3. Trabajo en equipo 21](#_Toc180828011)

[6. Conclusiones 22](#_Toc180828012)

[6.1. Análisis de desviaciones temporales y de tareas 22](#_Toc180828013)

[6.2. Conclusiones generales del proyecto 22](#_Toc180828014)

[6.2.1. Evaluación global del proyecto. 22](#_Toc180828015)

[6.2.2. Reflexión sobre el proceso de aprendizaje y desarrollo. 22](#_Toc180828016)

[6.2.3. Recomendaciones para futuros proyectos similares. 22](#_Toc180828017)

[6.3. Limitaciones y prospectiva 22](#_Toc180828018)

[6.3.1. Posibles mejoras y ampliaciones del proyecto. 22](#_Toc180828019)

[6.3.2. Nuevas líneas de investigación o desarrollo que podrían derivarse del proyecto. 22](#_Toc180828020)

[6.3.3. Sugerencias para la implementación en entornos reales. 22](#_Toc180828021)

[7. Referencias bibliográficas 23](#_Toc180828022)

[Anexo A. Diagramas de GANTT 24](#_Toc180828023)

[Anexo B. Código fuente de la solución y pruebas 25](#_Toc180828024)

[Anexo C. Manual de instalación - despliegue 26](#_Toc180828025)

[Anexo D. Documentación de la API 27](#_Toc180828026)

[Anexo E. Otros anexos de interés 28](#_Toc180828027)

Índice de figuras

[Tabla 1: Herramientas, lenguajes, frameworks y APIs utilizadas 10](#_Toc180824791)

[Tabla 2 Presupuesto temporal de tareas 11](#_Toc180824792)

[Tabla 3 Comparativa de aplicaciones actualmente en el mercado 13](#_Toc180824793)

[Ilustración 1 Análisis DAFO 14](#_Toc180824794)

[Ilustración 2 Diagrama E/R 15](#_Toc180824795)

[Ilustración 3 Diagrama de clases 16](#_Toc180824796)

[Ilustración 4 Caso de uso "Recepción de pedido" 16](#_Toc180824797)

[Ilustración 5 Wireframes 17](#_Toc180824798)

[Ilustración 6 Prototipo de interfaz de alta definición 17](#_Toc180824799)

[Ilustración 7 Paleta de colores 18](#_Toc180824800)

[Ilustración 8 Logotipo en positivo 18](#_Toc180824801)

[Ilustración 9 Logotipo en negativo 19](#_Toc180824802)

Índice de tablas

[Tabla 1: Herramientas, lenguajes, frameworks y APIs utilizadas 10](#_Toc180824803)

[Tabla 2 Presupuesto temporal de tareas 11](#_Toc180824804)

[Tabla 3 Comparativa de aplicaciones actualmente en el mercado 13](#_Toc180824805)

# Introducción

En la actualidad, la correcta gestión del tiempo laboral se ha convertido en un aspecto fundamental para las empresas. Tanto por motivos de organización interna como por cumplimiento de la normativa legal, llevar un registro de las horas de trabajo de los empleados es obligatorio en España desde el Real Decreto-ley 8/2019.

Este proyecto tiene como objetivo el desarrollo de un sistema de fichajes que permita a una empresa gestionar de forma sencilla y eficiente los horarios de entrada y salida de sus trabajadores. Además, el sistema contempla funcionalidades como el control de horas extra, la validación por parte de responsables, la auditoría de actividad y la autenticación segura mediante verificación en dos pasos (2FA).

La aplicación está pensada para ser utilizada tanto por empleados como por el equipo de dirección y administración de la empresa. Por ello, se ha desarrollado una estructura basada en roles que permite adaptar las funcionalidades según el tipo de usuario.

## Justificación

Desde la entrada en vigor del Real Decreto-ley 8/2019, todas las empresas están obligadas a registrar diariamente la jornada laboral de sus empleados. Esta medida busca garantizar los derechos de los trabajadores y prevenir el fraude laboral.

Muchas empresas, sin embargo, aún utilizan métodos rudimentarios como hojas de cálculo o registros en papel. Esto puede derivar en errores, pérdidas de información o incluso sanciones legales.

Este sistema de fichajes nace para resolver esa necesidad, aportando una solución digital, segura y adaptable que no solo cumple con la normativa, sino que también mejora la gestión interna del tiempo de trabajo.

## Objetivos

* Objetivo general:

Desarrollar un sistema web que permita a las empresas registrar y gestionar el horario laboral de sus empleados, cumpliendo la normativa vigente y optimizando la gestión del tiempo.

* Objetivos específicos:

Implementar el registro de entrada y salida de empleados.

Permitir la solicitud y validación de horas extra.

Desarrollar un sistema de roles (empleado, dirección, administrador).

Garantizar la seguridad mediante autenticación 2FA.

Registrar un historial de actividad para auditoría.

Gestionar usuarios, empresas y permisos.

Utilizar una arquitectura moderna cliente-servidor.

Realizar pruebas que verifiquen el correcto funcionamiento del sistema.

# Módulos formativos aplicados en el trabajo

**Programación:** Lógica de la aplicación, uso de estructuras y algoritmos.

**Entornos de Desarrollo:** Diagramas UML y casos de uso.

**BBDD:** base de datos MySQL.

**Lenguajes de Marcas y Sistemas de Gestión de Información:** Html y css.

**Acceso a datos:** Conexión a bases de datos relacionales usando JPA/Hibernate.

**Desarrollo de Interfaces:** Frontend en Angular o React.

**Sistemas de gestión empresarial:** Organización y gestión de proyectos.

**Desarrollo de Software y Operaciones con Metodología Dev-Ops:** Docker Compose.

**Empresa e iniciativa emprendedora:** Estudio de mercado y análisis DAFO.

**Inglés:** Descripción en inglés.

**Formación en centros de trabajo:** Aplicación práctica del proyecto.

# Herramientas y lenguajes utilizados

|  |  |
| --- | --- |
| Tecnología / Herramienta | Descripción |
| Java + Spring Boot | Framework robusto para el desarrollo de APIs RESTful en Java. Permite crear aplicaciones backend estructuradas y escalables con facilidad. |
| Hibernate (JPA) | ORM (Object-Relational Mapping) que facilita la conexión y gestión de bases de datos desde Java. Permite mapear objetos Java a tablas de base de datos. |
| MySQL | Sistema de gestión de bases de datos relacional, muy utilizado por su fiabilidad, rendimiento y compatibilidad con múltiples entornos. |
| Angular | Frameworks modernos para el desarrollo frontend. Permiten crear interfaces de usuario dinámicas y reactivas en aplicaciones web. |
| TypeScript, Tailwind, Angular Material | Tecnologías para estructurar y diseñar páginas web con un diseño responsive, moderno y adaptado a distintos dispositivos. |
| DBeaver | Herramienta visual para la administración de bases de datos. Facilita la ejecución de consultas SQL y la gestión de esquemas. |
| Postman / Thunder Client | Herramientas para probar y documentar APIs REST. Permiten enviar peticiones HTTP y verificar respuestas del servidor. |
| GitHub | Sistema de control de versiones y plataforma colaborativa. Permiten gestionar cambios en el código y trabajar en equipo de forma organizada. |
| Google Authenticator | Aplicación para implementar verificación en dos pasos (2FA), mejorando la seguridad de acceso al sistema con códigos temporales. |
| Figma | Herramienta de diseño colaborativo para crear prototipos de interfaces. Ideal para planificar visualmente la estructura del frontend. |

# Metodologías utilizadas

Hemos seguido una **metodología ágil**, dividiendo el trabajo en iteraciones (sprints) con tareas para cada integrante del equipo. Hemos realizado reuniones semanales, estamos utilizando GitHub para coordinación y revisión del código, y se trabaja de forma colaborativa dividiendo roles y responsabilidades.

# Componentes del equipo y aportaciones realizadas por cada alumno

## Estudio de mercado

Existen aplicaciones de control horario en el mercado, pero muchas son costosas, poco intuitivas o no se adaptan a PYMEs. Este proyecto busca cubrir esa necesidad mediante una solución accesible, segura y fácil de usar.

### Tabla comparativa de aplicaciones actualmente en el mercado

Tabla 3 Comparativa de aplicaciones actualmente en el mercado

|  |  |  |  |
| --- | --- | --- | --- |
| Característica | Sesame HR | Planday | Factorial |
| Descripción | Plataforma integral para la gestión del tiempo, incluyendo fichajes y RRHH. | Herramienta para la planificación de turnos y fichajes, ideal para empresas con horarios rotativos. | Software de recursos humanos con sistema de fichaje y gestión documental. |
| Funcionalidades | Fichaje por app o web, geolocalización, control de vacaciones, informes. | Fichaje móvil, planificación de turnos, notificaciones, integraciones con nóminas. | Fichaje desde app/web, control horario, vacaciones, ausencias, gestión de documentos. |
| Público Objetivo | Empresas de cualquier tamaño que necesitan controlar la jornada laboral. | Negocios con turnos cambiantes, como restaurantes, comercios y sanidad. | PYMES que buscan una solución completa de RRHH y control horario. |
| Plataformas | iOS, Android, Web | iOS, Android, Web | iOS, Android, Web |
| Precio | Desde 3€/usuario/mes. | Modelo por suscripción, desde 2€/usuario/mes. | Plan gratuito y planes de pago desde 4€/usuario/mes. |
| Puntuación | 4.6/5.0 | 4.5/5.0 | 4.6/5.0 |

### Análisis DAFO de nuestra solución

Ilustración 1 Análisis DAFO

1. Fortalezas

* Cumplimiento legal: El sistema responde a la normativa legal vigente que obliga a las empresas a registrar la jornada laboral de sus trabajadores, lo que lo hace esencial para cualquier organización.
* Arquitectura moderna: Uso de tecnologías actuales como Spring Boot, JPA, Angular/React y MySQL, que garantizan escalabilidad, rendimiento y mantenimiento a largo plazo.
* Sistema de roles: Separación clara entre usuarios (empleados, dirección y empresa) permite un control detallado de permisos y responsabilidades.
* Seguridad: Implementación de autenticación segura (con posibilidad de 2FA como Google Authenticator) que protege los datos sensibles de los trabajadores.
* Informes y estadísticas: Generación de informes y gráficos permite a la dirección tomar decisiones basadas en datos reales y verificables.
* Facilidad de uso: Interfaz intuitiva y clara que facilita la adopción del sistema por parte de los empleados sin necesidad de formación intensiva.

1. Debilidades

* Dependencia tecnológica: El uso de tecnologías modernas puede suponer una barrera para pequeñas empresas sin conocimientos técnicos o recursos informáticos avanzados.
* Requiere conexión a internet: Si se aloja en la nube, una mala conexión puede afectar el funcionamiento del sistema.
* Curva de aprendizaje inicial: Aunque la interfaz es amigable, los usuarios administrativos pueden necesitar algo de tiempo para familiarizarse con todas las funcionalidades avanzadas.
* Mantenimiento necesario: Como cualquier sistema informático, necesita actualizaciones, copias de seguridad y soporte técnico periódico.

1. Oportunidades

* Expansión a otros sectores: Puede adaptarse fácilmente para distintos sectores que necesiten control horario (construcción, hostelería, educación, etc.).
* Integración con nóminas: Puede evolucionar para integrarse con sistemas de gestión de nóminas, ofreciendo una solución más completa a las empresas.
* Escalabilidad: Posibilidad de ofrecerlo como servicio SaaS (Software as a Service) para múltiples empresas.
* Concienciación legal creciente: Cada vez más empresas están siendo conscientes de sus obligaciones legales respecto al control horario, lo que aumenta la demanda de soluciones como esta.
* Automatización de procesos: Potencial para incluir inteligencia artificial en el futuro, como detección de anomalías en los fichajes o recomendaciones para optimizar horarios.

1. Amenazas

* Competencia: Existen otras soluciones comerciales de fichaje muy consolidadas en el mercado.
* Cambios legislativos: Una modificación en la normativa laboral podría hacer que ciertas funcionalidades del sistema queden obsoletas o innecesarias.
* Ciberseguridad: Al manejar datos personales y laborales, el sistema puede ser objetivo de ataques si no se refuerzan bien las medidas de protección.
* Resistencia al cambio: Algunas empresas o empleados pueden mostrarse reacios a implementar un sistema de control horario digital por miedo al control excesivo o a la pérdida de privacidad.

### Subapartado 1.2

Texto Normal del menú de estilos.

## Modelo de datos

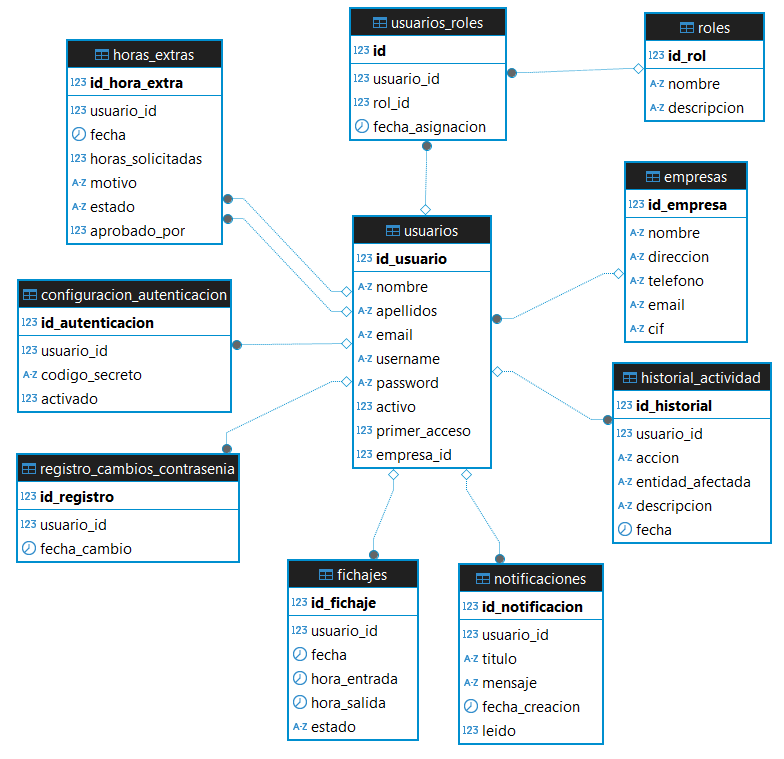


Ilustración 2 Diagrama E/R

## Diagramas UML

Breve introducción a lo que son los diagramas de clases y de casos de uso

Enumeración de los diagramas aportados

### Diagrama de clases

Breve explicación del diagrama, sus clases y relaciones reseñables

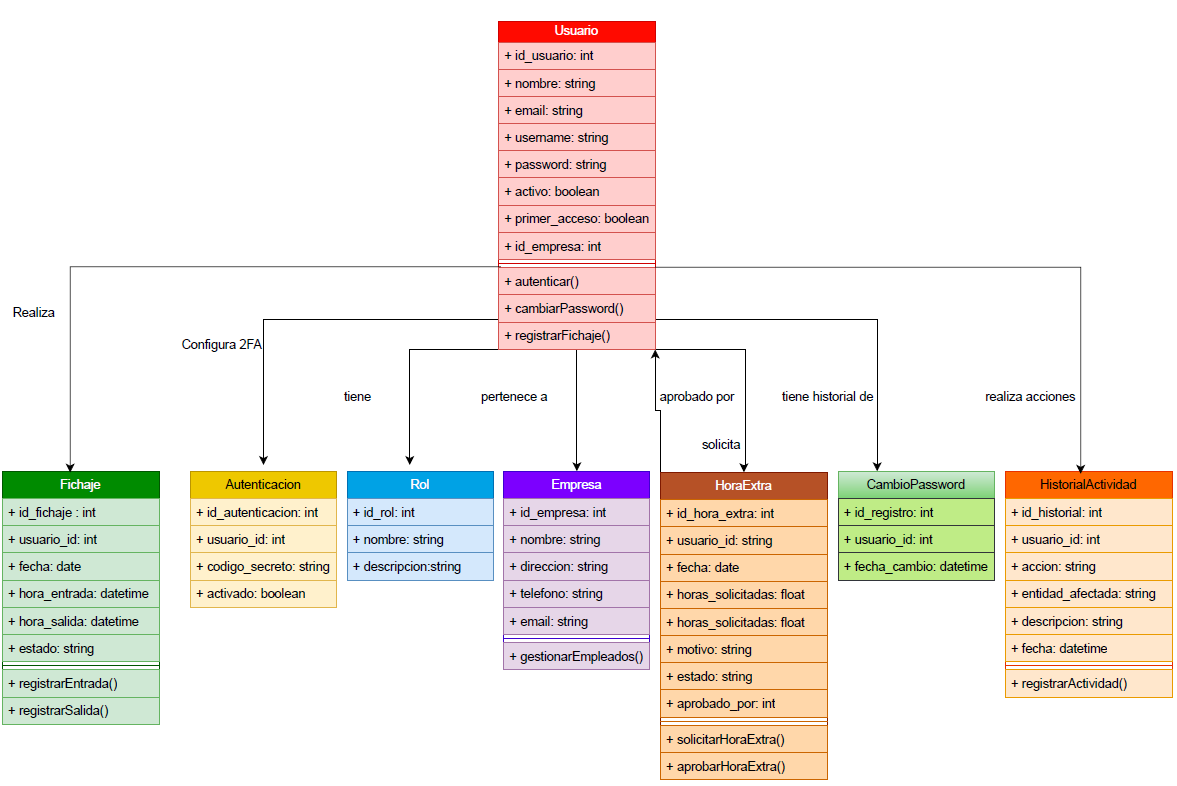


Ilustración 3 Diagrama de clases

### Clasificación de usuarios

1. **Administrador**

* **Gestión de empleados**:
  + Crear, leer, actualizar y eliminar empleados.
  + Asignar roles y gestionar datos de la empresa.
* **Generar reportes**:
  + Consultar estadísticas e informes sobre fichajes y horas extras.

1. **Dirección**

* **Validación de horas extras**:
  + Aprobar o rechazar solicitudes de horas extras.
* **Consulta de fichajes**:
  + Acceder a reportes detallados (entradas, salidas y pausas).

1. **Empleado**

* **Registro de fichajes**:
  + Registrar hora de entrada y salida.
* **Gestión de horas extras**:
  + Solicitar horas extras.
* **Gestión de cuenta**:
  + Cambiar contraseña tras el primer acceso.
  + Consultar historial de fichajes.

**Persona 1: Laura, la Administradora**

* **Edad:** 44 años
* **Ocupación:** Responsable de Recursos Humanos
* **Objetivos:**
  + Mantener actualizados los datos de los empleados.
  + Generar informes de actividad y asistencia.
* **Frustraciones:**
  + Sistemas lentos o complejos.
  + Tener que recopilar información manualmente.
* **Necesidades:**
  + Una interfaz clara para gestionar empleados.
  + Exportación rápida de reportes.

**Persona 2: Jesús, el director de departamento**

* **Edad:** 47 años
* **Ocupación:** Director de Operaciones
* **Objetivos:**
  + Aprobar o rechazar solicitudes de horas extras.
  + Consultar la productividad del equipo.
* **Frustraciones:**
  + Falta de visibilidad en tiempo real.
  + Demoras en la validación de fichajes.
* **Necesidades:**
  + Acceso rápido a estadísticas.
  + Notificaciones sobre solicitudes pendientes.

**Persona 3: Roberto, empleado**

* **Edad:** 29 años
* **Ocupación:** Técnico administrativo
* **Objetivos:**
  + Fichar correctamente su jornada.
  + Consultar su historial de horas.
* **Frustraciones:**
  + Olvidar fichar.
  + No saber si una hora extra ha sido aprobada.
* **Necesidades:**
  + Sistema accesible desde el móvil.
  + Confirmación clara de sus acciones (fichajes, solicitudes).

### Caso de uso 01

Explicación de los distintos casos de uso:

**Empleado**: Puede realizar el registro de fichajes, solicitar horas extras y cambiar su contraseña. Además, tras su primer acceso, debe configurar un método de autenticación adicional (como autenticación en dos pasos), lo cual extiende la funcionalidad de cambio de contraseña.

**Dirección**: Se encarga de validar o rechazar las solicitudes de horas extras generadas por los empleados y puede consultar los fichajes registrados. Esta validación se incluye dentro del caso de uso “Registrar horas extras”.

**Administrador**: Tiene control total sobre la gestión del sistema. Sus funcionalidades incluyen la asignación de roles a los usuarios, la gestión completa (CRUD) de empleados y empresas, así como la generación de reportes e informes sobre la actividad del personal.

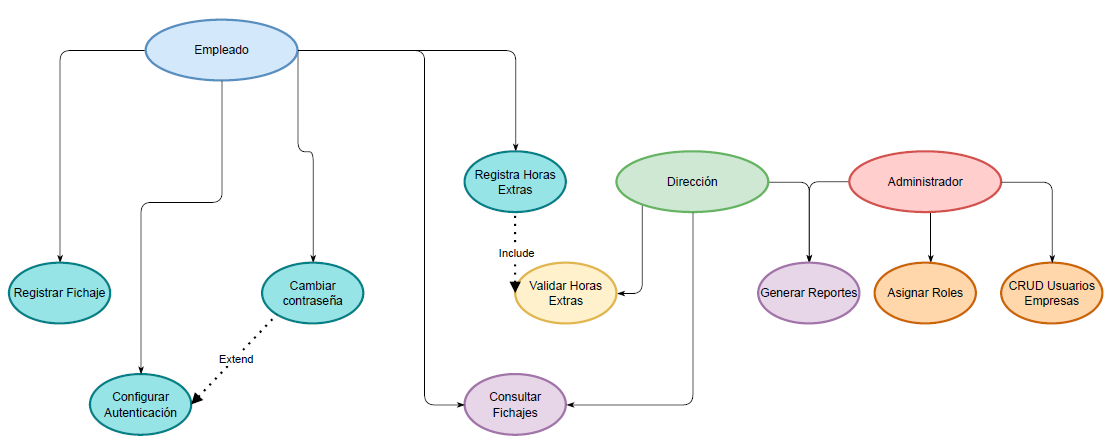


Ilustración 4 Caso de uso "Recepción de pedido"

## Diseño de interfaces

### Wireframes

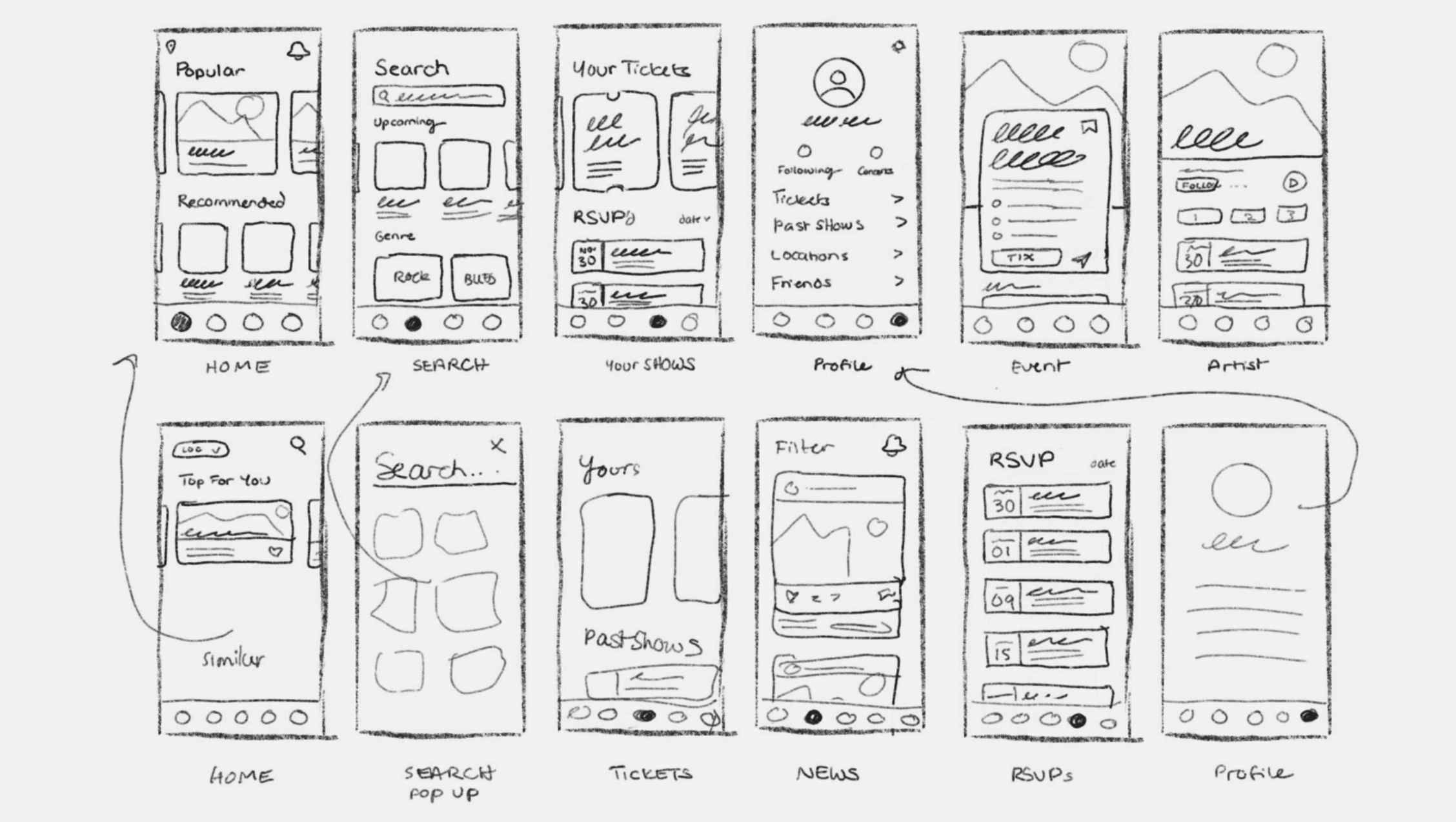


Ilustración 5 Wireframes

### Prototipo de interfaz de alta definición

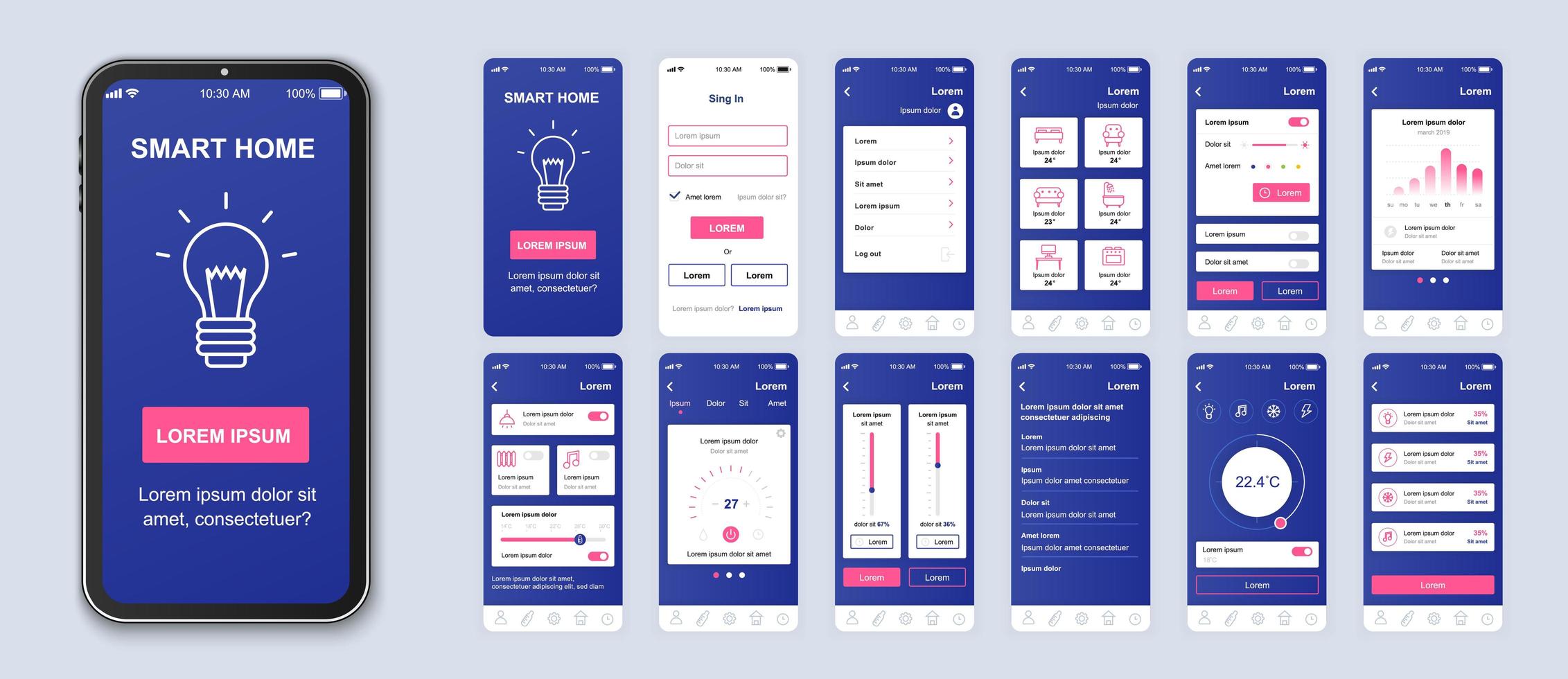


Ilustración 6 Prototipo de interfaz de alta definición

### Paleta de colores

Explicación de la elección de colores

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Ilustración 7 Paleta de colores

<https://color.adobe.com/es/create/image>

### Logotipo

Explicación de la elección de logotipo

Icono

Descripción generada automáticamente

Ilustración 8 Logotipo en positivo

Logotipo en negativo

Icono

Descripción generada automáticamente

Ilustración 9 Logotipo en negativo

## Planificación temporal y trabajo en equipo

### Presupuesto temporal de tareas

Puedes basarte en la tabla siguiente

Tabla 2 Presupuesto temporal de tareas

|  |  |  |  |
| --- | --- | --- | --- |
| **Tareas** | **Subtareas** | **Duración (horas)** | **Persona/s asignada/s** |
| **Planificación del Proyecto** | Definir objetivos y alcance |  |  |
| Identificar recursos |  |  |
| Crear plan de proyecto |  |  |
| … |  |  |
| **Diseño de la Aplicación** | Crear wireframes y maquetas |  |  |
| Diseñar arquitectura de BD |  |  |
| Definir experiencia de UX |  |  |
| … |  |  |
| **Desarrollo Frontend** | Implementar interfaz de usuario |  |  |
| Integrar diseño responsivo |  |  |
| Realizar pruebas de usabilidad |  |  |
| … |  |  |
| **Desarrollo Backend** | Configurar servidor y BD |  |  |
| Implementar lógica de negocio |  |  |
| Crear APIs y servicios web |  |  |
| … |  |  |
| **Integración y Pruebas** | Integrar frontend y backend |  |  |
| Realizar pruebas funcionales |  |  |
| Corregir errores y optimizar |  |  |
| … |  |  |
| **Despliegue y Lanzamiento** | Configurar entorno de producción |  |  |
| Desplegar la aplicación |  |  |
| Realizar pruebas finales |  |  |
| … |  |  |

### Organización de tareas y tiempos finales

Puedes basarte en la tabla anterior

En los anexos incorpora el diagrama de Gantt correspondiente al presupuesto y el resultante al final del proyecto.

### Trabajo en equipo

Debes definir cuáles han sido vuestras aportaciones.

Se debe aportar, además, el/los gráficos de aportaciones al repositorio de GitHub

# Conclusiones

## Análisis de desviaciones temporales y de tareas

Realiza un análisis de las diferencias entre el presupuesto inicial de tareas y tiempos y las tareas y tiempos realizados finalmente.

Puedes apoyarte en una tabla comparativa, comentando los motivos de las posibles desviaciones.

## Conclusiones generales del proyecto

### Evaluación global del proyecto.

### Reflexión sobre el proceso de aprendizaje y desarrollo.

### Recomendaciones para futuros proyectos similares.

## Limitaciones y prospectiva

### Posibles mejoras y ampliaciones del proyecto.

El sistema de fichajes desarrollado cumple con los objetivos propuestos inicialmente, pero se han identificado varias posibles mejoras que permitirían ampliar su funcionalidad y adaptarlo a necesidades más complejas, especialmente en empresas de mayor tamaño.

**1. Adaptación para grandes empresas y gestión avanzada de roles**

Aunque la solución actual está orientada a pequeñas y medianas empresas, se plantea su evolución para dar soporte a grandes organizaciones. Esto implicaría:

* Permitir la creación de **nuevos roles personalizados** por parte del administrador, con posibilidad de definir permisos específicos.
* Incorporar un sistema de **gestión de permisos detallado**, que permita configurar el acceso a funcionalidades concretas para cada rol.
* Posibilitar la **asignación jerárquica o por grupo**, útil en estructuras empresariales con varios niveles de responsabilidad.

**2. Organización por departamentos y estructura jerárquica**

Para facilitar una gestión más precisa, especialmente en entornos con muchos empleados, se propone:

* Añadir una tabla de **departamentos**, que permita agrupar a los usuarios según su área o función dentro de la empresa.
* Asociar cada departamento a un usuario con rol de dirección (u otro rol definido), que se encargue de la gestión específica de su equipo.
* Esto permitiría que la supervisión de fichajes y la validación de horas extra se realice de forma descentralizada, mejorando la eficacia del sistema.

**3. Importación y sincronización de datos con sistemas externos**

Pensando en la facilidad de integración con otras herramientas empresariales, sería útil:

* Desarrollar una **API que permita importar automáticamente empleados** desde bases de datos externas del cliente.
* Incluir también la opción de **subir ficheros CSV o Excel**, para facilitar la migración de datos sin necesidad de intervención técnica.
* Implementar un sistema de **sincronización periódica**, útil para mantener actualizada la información en ambos sistemas cuando se utilicen de forma conjunta.

**4. Autenticación en dos pasos (2FA) completamente operativa**

La autenticación en dos pasos ya está contemplada como parte del sistema, pero se prevé su integración completa en versiones futuras:

* Permitir que el uso de 2FA sea opcional o obligatorio, dependiendo del rol o del propio usuario.
* Incluir el uso de herramientas como **Google Authenticator**, con códigos QR y tokens temporales (TOTP).
* Registrar cada intento de autenticación como parte del sistema de auditoría.

**5. Evolución hacia una solución SaaS y modular**

A largo plazo, el sistema podría ofrecerse como un servicio bajo demanda (SaaS), lo que permitiría:

* Gestionar múltiples empresas desde una misma plataforma, cada una con su propia configuración, usuarios y datos independientes.
* Activar únicamente los módulos necesarios (fichajes, horas extra, informes, etc.), adaptando la herramienta a cada caso concreto.
* Mejorar la escalabilidad mediante tecnologías como Docker y Kubernetes, para facilitar el despliegue y la gestión de entornos.

### Nuevas líneas de investigación o desarrollo que podrían derivarse del proyecto.

### Sugerencias para la implementación en entornos reales.

# Referencias bibliográficas

Design Thinking Services. (2023). *Método persona.* (D. T. Services, Ed.) Recuperado el 01 de 10 de 2024, de https://www.designthinking.services/herramientas-design-thinking/metodo-persona/

Figma. (2023). *Figma: Herramienta de diseño de interfaces.* Recuperado el 20 de 10 de 2024, de https://www.figma.com

Kranio. (29 de 08 de 2023). *Descubriendo Git: Características y Ventajas.* Recuperado el 10 de 2024, de https://www.kranio.io/blog/descubriendo-git-caracteristicas-y-ventajas

1. Diagramas de GANTT
2. Código fuente de la solución y pruebas
3. Manual de instalación - despliegue
4. Documentación de la API

En caso de que se implemente una API

1. Otros anexos de interés

Puedes crear los adjuntos que consideres necesarios – de acuerdo con el tutor de proyecto – según la necesidad y naturaleza del proyecto.