



PROEDUCA

UNIR FP

Informática y Comunicaciones

Técnico Superior en DAM - DAW - ASIR

Título del Trabajo Fin de Estudios

Trabajo fin de estudio presentado por:	Ana González Bueno Daniel Torres Retuerto Carlos Otazo
Tipo de trabajo:	TFC
Tutor/a:	Damián Sualdea
Fecha:	14/05/2025

Resumen:

Presentamos una solución de control horario. Desarrollamos una aplicación que cumpliendo con la normativa laboral, permita el registro horario y el control y gestión de turnos de trabajo.

Ha de permitir que los empleados fichen sin geolocalización ni biometrías para asegurar cumplir la normativa en materia de protección de datos personales. Lo que implica la generación de sistema de acceso mediante claves para cada usuario y proponemos el uso en las instalaciones de trabajo para evitar la geolocalización.

Los datos de los empleados, han de ser accesibles para cada trabajador de forma personal y para la administración competente.

Los turnos de trabajo, han de poder gestionarse cumpliendo la normativa aplicable en cuanto a cómputo anual de horas trabajadas, horas extras de cualquier tipo, y horas complementarias.

Ha de asegurarse el personal mínimo requerido por la empresa para el desempeño de su actividad.

Palabras clave: turnos de trabajo, control horario, seguridad y accesibilidad.

Índice de contenidos:

Introduccion	5
Modulos formativos aplicados en el trabajo	6
Herramientas y lenguajes utilizados	7
Metodologias utilizadas	11
Analisis DAFO	13
Modelo de datos.....	14
Diseno de interfaces	19
Planificacion temporal y trabajo en equipo.....	22
Conclusiones	24
Referencias Bibliograficas.....	26
Anexo B.....	26
Anexo C	26

Índice de figuras:

Tabla 1: Módulos formativos aplicados al trabajo.....	6
---	---

Tabla 2: Herramientas, lenguajes, frameworks y APIs utilizadas	7
Tabla 3 Comparativa de aplicaciones actualmente en el mercado.....	11
Tabla 4 Presupuesto temporal de tareas.....	22
Ilustración 1 Diagrama E/R.....	14
Ilustración 2 Diagrama de clases	15
Ilustración 3 Caso de uso "Recepción de pedido"	18
Ilustración 4 Wireframes	19
Ilustración 5 Prototipo de interfaz de alta definición	20
Ilustración 6 Logotipo	21

- **Introducción:**
- **Justificación:**

Desarrollo de una aplicación de fácil manejo para el registro horario, y la gestión de turnos de trabajo (solicitud de cambios y gestión de los mismos) de forma sencilla y semiautomática.

- **Objetivos:**

Automatizar cambios de turno y horarios asegurando el cumplimiento de la normativa general aplicable, dejando lugar pequeños ajustes para adaptarlo a cada convenio laboral que aplique cambios de mejora respecto del general.

- Objetivo general: automatizar cambios de turno.
- Objetivos específicos : ampliar a la necesidad legal de control horario y asegurar cumplimiento de requisitos horarios fijados por cada convenio colectivo de forma automática.

- **Módulos formativos aplicados en el trabajo:**

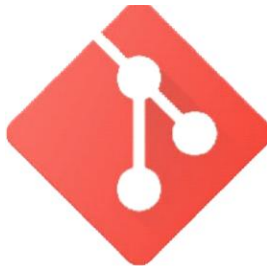
	Nuestra aplicacion de gestion de turnos, requiere de una base de datos en la que almacenar las fichas de cada trabajador, para
--	--

BASES DE DATOS	desarrollar esta parte utilizamos lo aprendido en este modulo, junto con el modulo de desarrollo web en entorno servidor, que mencionamos debajo.
LENGUAJE DE MARCAS	Utilizamos html y css para los estilos de la aplicacion, por tanto implementando contenidos del modulo de lenguaje de marcas para toda la parte visual.
DISEÑO DE INTERFACES	Continuando con el apartado grafico, nos apoyamos de figma para la creacion de los wireframes y los mockups de alta definicion, para poder guiarnos mejor a la hora de desarrollar y tener las pautas a seguir.
ENTORNOS DE DESARROLLO	Utilizamos github para el control de versiones, permitiendo una eficaz coordinacion y seguimiento del proyecto.
DESARROLLO WEB EN ENTORNO CLIENTE	Para el desarrollo de la parte frontend, hemos utilizado JavaScript, TypeScript y angular, implementando gran parte del temario de la asignatura.
DESARROLLO WEB EN ENTORNO SERVIDOR	El proyecto utiliza las distintas tecnologías aprendidas en el modulo de desarrollo web en entorno servidor.

- **Herramientas y lenguajes utilizados:**

Especifica los lenguajes, frameworks, APIs y herramientas utilizadas, junto con una breve descripción de estas.

Git



Git es un sistema de control de versiones distribuido, creado por Linus Torvalds en 2005. Su propósito principal es gestionar el desarrollo del kernel de Linux, pero su flexibilidad y eficiencia lo han convertido en una herramienta ampliamente adoptada en la industria del software. Git permite a los desarrolladores rastrear cambios en el código fuente, colaborar en proyectos y revertir a versiones anteriores si es necesario. Utiliza un modelo de datos basado en instantáneas, lo que garantiza la integridad y consistencia de los datos. Además, Git facilita la creación de ramas y fusiones, lo que permite a los equipos trabajar en paralelo sin conflictos (Kranio, 2023).

Figma



Figma es una herramienta de diseño de interfaces basada en la web que permite a los usuarios trabajar de manera colaborativa en tiempo real desde cualquier dispositivo con acceso a Internet. Ofrece un espacio de trabajo flexible para crear diseños y prototipos de manera eficiente, facilitando la colaboración y la consistencia en los proyectos.

Angular

Angular es un framework de desarrollo web basado en TypeScript, desarrollado y mantenido por Google. Se utiliza para crear aplicaciones de una sola página



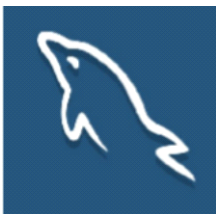
(SPA) con una arquitectura modular y basada en componentes. Ofrece herramientas como el enlace de datos bidireccional, inyección de dependencias y una estructura escalable que facilita el mantenimiento y desarrollo de aplicaciones complejas.

Spring Tools



Spring Tools Suite (STS) es un entorno de desarrollo basado en Eclipse, diseñado específicamente para aplicaciones Java con Spring Framework. Incluye herramientas que facilitan la configuración, desarrollo y depuración de aplicaciones Spring Boot, proporcionando soporte para integraciones con bases de datos, servidores y microservicios.

MySQL Workbench



MySQL Workbench es una herramienta de diseño y administración para bases de datos MySQL. Permite la modelación de bases de datos, ejecución de consultas SQL, administración de servidores y optimización de rendimiento. Es ampliamente utilizado en el desarrollo de aplicaciones que requieren almacenamiento y gestión de datos estructurados.

--	--

<p><u>TypeScript</u></p> 	<p>TypeScript es un lenguaje de programación desarrollado por Microsoft que extiende JavaScript con tipado estático opcional. Es utilizado en el desarrollo de aplicaciones modernas, especialmente con frameworks como Angular, permitiendo un código más estructurado y mantenible.</p>
<p><u>Java</u></p> 	<p>Java es un lenguaje de programación orientado a objetos y multiplataforma, ampliamente utilizado en el desarrollo de aplicaciones empresariales, web y móviles. Su compatibilidad con el ecosistema de Spring Boot lo convierte en una opción ideal para el desarrollo de aplicaciones backend robustas y escalables.</p>
<p><u>MySQL</u></p> 	<p>MySQL es un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado en aplicaciones web y empresariales. Ofrece alto rendimiento, escalabilidad y seguridad, siendo una de las opciones más populares para almacenar y gestionar grandes volúmenes de datos.</p>
<p><u>Bootstrap</u></p> 	<p>Bootstrap es un framework de diseño web basado en HTML, CSS y JavaScript, utilizado para la creación de interfaces responsivas y atractivas. Proporciona una amplia variedad de componentes predefinidos como botones, formularios, alertas y barras de navegación, facilitando el desarrollo de aplicaciones con un diseño moderno y consistente.</p>

Swagger UI



Swagger UI es una herramienta de código abierto que permite visualizar y probar APIs REST de manera interactiva. Forma parte del ecosistema Swagger y facilita la documentación de servicios web, permitiendo a los desarrolladores explorar endpoints, enviar peticiones y comprender la estructura de las APIs sin necesidad de herramientas adicionales. Es ampliamente utilizado en aplicaciones que exponen servicios RESTful con Spring Boot.

- Metodologías utilizadas

Describe la metodología de desarrollo de proyecto aplicada

- Estudio de mercado:

Estudio de la temática a desarrollar, análisis de la competencia y mejoras a realizar, estudio del usuario de la aplicación.

- Tabla comparativa de aplicaciones actualmente en el mercado

Tabla 3 Comparativa de aplicaciones actualmente en el mercado

Característica	Bizneo	Clockify	Factorial	FichaWork
Descripción	Software para gestión de tiempo	Software para gestión de tiempo	Software para gestión de tiempo	Software para gestión de tiempo
		Control horario, gestión de turnos, control de ausencias y vacaciones.	Control horario, gestión de turnos, control de ausencias y vacaciones.	Control horario, control de ausencias y vacaciones
Funcionalidades	Ausencias y vacaciones. Permite integración de más servicios de rrhh	Gestión de tiempo, registro actividad e incluso proyectos. Gastos y ganancias	Permite integración de más servicios de rrhh.	
Público	Pequeña,	Pequeña	Pequeña,	Pequeña

Característica	Bizneo	Clockify	Factorial	FichaWork
Objetivo	Mediana y gran empresa	Empresa e individual	Mediana y gran empresa.	Empresa e individual
Plataformas	Mac y Windows, iOS, Android.	iOS, Android.	Mac y Windows, iOS, Android.	

Precio	Empleado/mes	Gratis con compras dentro de la app (suscripción Pro).	Suscripción anual.	Empleado/mes
Puntuación	4.6/5.0	4.7/5.0	4.7/5.0	4.6/5.0

- **Análisis DAFO de nuestra solución:**

Debilidades:

- Al ser un proyecto desarrollado por alumnos en formación, el proyecto carece del alcance técnico frente a soluciones comerciales ya consolidadas.
- La aplicación no cuenta con pruebas reales en entornos de producción, con lo cual puede resultar complicado identificar errores en operación.

- Requiere conexión constante al servidor.

Amenazas:

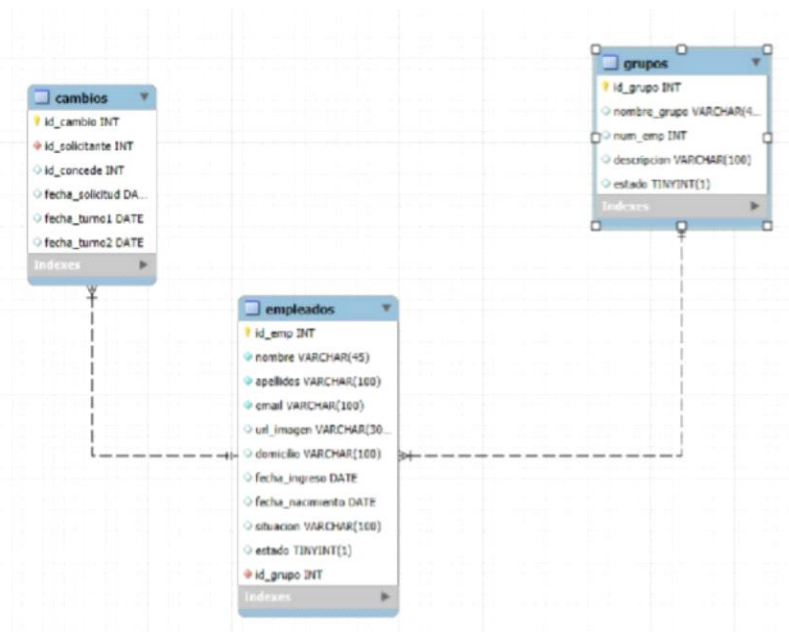
- El mercado de aplicaciones de control horario tiene competencia, la cual ofrece soluciones de la mano de una fuerte inversión de marketing.
- Riesgo de que los usuarios con menos afinidad tecnológica presenten inconvenientes a la hora de interactuar con la aplicación.

Fortalezas:

- La arquitectura es modular y esta basada en tecnologías actuales (Angular, SpringBoot, MySQL), lo que permite tener buena escalabilidad a mediano y largo plazo.
- Gracias a herramientas como Figma o Bootstrap, la interfaz es clara y funcional.

Oportunidades:

- La demanda para las aplicaciones de gestión horaria es creciente.
- El sistema es versátil y podría escalar y evolucionar hacia una herramienta más completa de gestión interna, incluyendo por ejemplo gestión de ausencia o vacaciones.
- **Modelo de datos:**
 - **Diseño de la base de datos:**



- **Diagramas UML:**

El diagrama de clases es un modelo estático que representa la estructura del sistema, mostrando sus clases, atributos, métodos y las relaciones entre ellas. Es una representación fundamental en el diseño orientado a objetos y ayuda a definir cómo los objetos interactúan en el sistema.

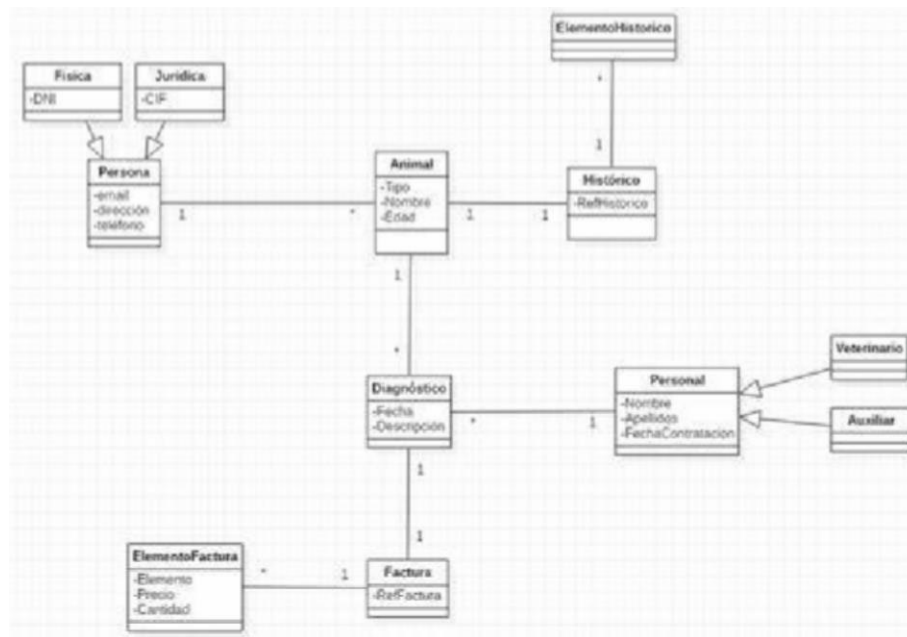
El diagrama de casos de uso es un modelo dinámico que representa las interacciones entre los actores (usuarios o sistemas externos) y el sistema. Se utiliza para describir los requisitos funcionales desde la perspectiva del usuario.

- **Diagrama de clases:**

Empleado 1 → * Grupo (Un empleado pertenece a un grupo)

Grupo 1 → * Cambio (Un grupo puede tener muchos cambios)

Empleado 1 → * Cambio (Un empleado puede solicitar muchos cambios)



- **Clasificación de usuarios**

En el sistema desarrollado, existen diferentes tipos de usuarios con roles y permisos específicos. Para definir estos perfiles, se ha utilizado el método Persona (UX) del Design Thinking Services (2023), que ayuda a representar los usuarios con base en sus necesidades, objetivos y comportamientos.

Tipos de Usuarios y Perfiles:

- **Administrador**

Usuario encargado de la gestión del sistema. Tiene acceso total para administrar empleados, grupos y realizar modificaciones en la estructura del sistema.

Objetivos y Necesidades

- Gestionar empleados (altas, bajas y modificaciones).
- Asignar empleados a grupos y definir turnos.
- Cambiar el estado de los empleados (activo/inactivo).
- Monitorear la organización de los equipos de trabajo.

Funciones en el sistema:

- Acceso completo a la plataforma.
- Creación y edición de empleados y grupos.
- Modificación de la asignación de empleados.
- Control de acceso y permisos.

- **Empleado**

Usuario estándar que pertenece a un grupo de trabajo dentro de la empresa. Puede visualizar su información personal y recibir asignaciones de turnos.

Objetivos y Necesidades

- Consultar su grupo y horario de trabajo.
- Acceder a su perfil y estado laboral.
- Notificar cambios en su disponibilidad.

Funciones en el sistema:

- Visualización de su información personal.
- Consulta de horarios y grupo asignado.

- Notificación de disponibilidad y estado.

Caso de Uso: Gestión de Empleados y Grupos

Descripción del Caso de Uso:

Este caso de uso describe cómo los administradores pueden gestionar los empleados y asignarlos a grupos dentro del sistema. El flujo incluye la verificación del estado del empleado y la asignación de grupos según disponibilidad.

Actores:

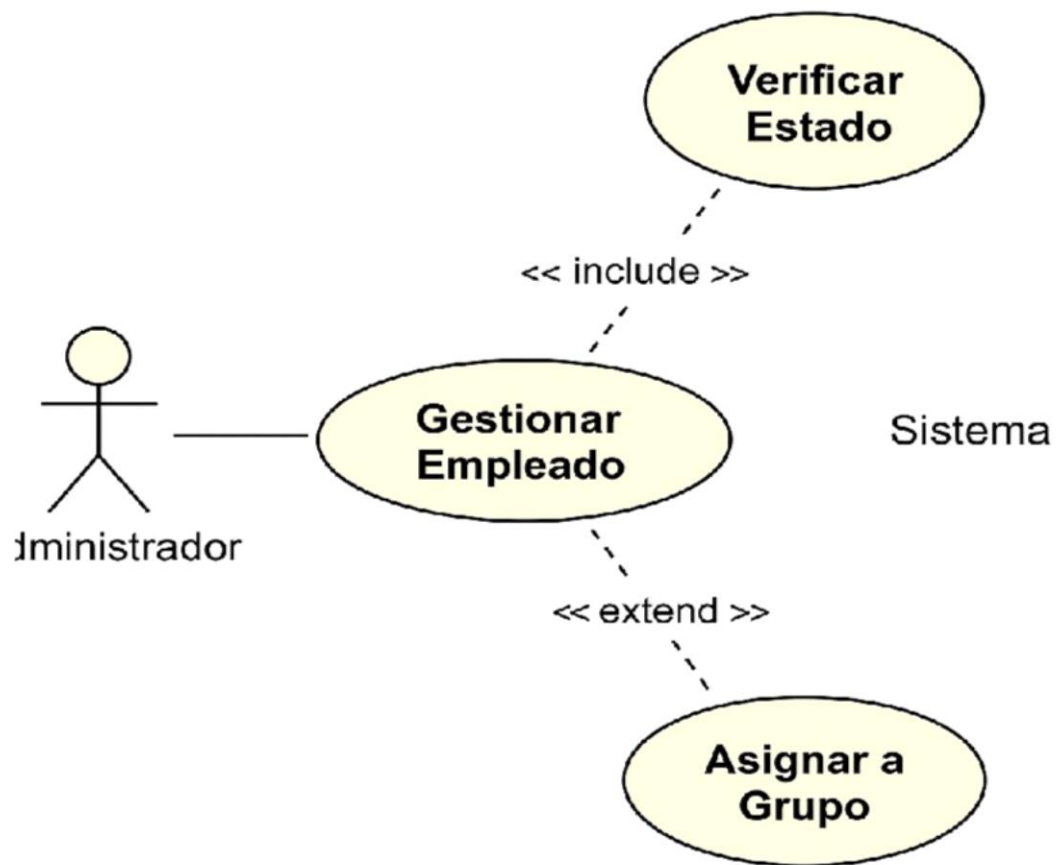
- **Administrador:** Tiene permisos para gestionar empleados y asignarlos a grupos.
- **Sistema:** Plataforma que permite visualizar y modificar la información de empleados y grupos.

Flujo Principal:

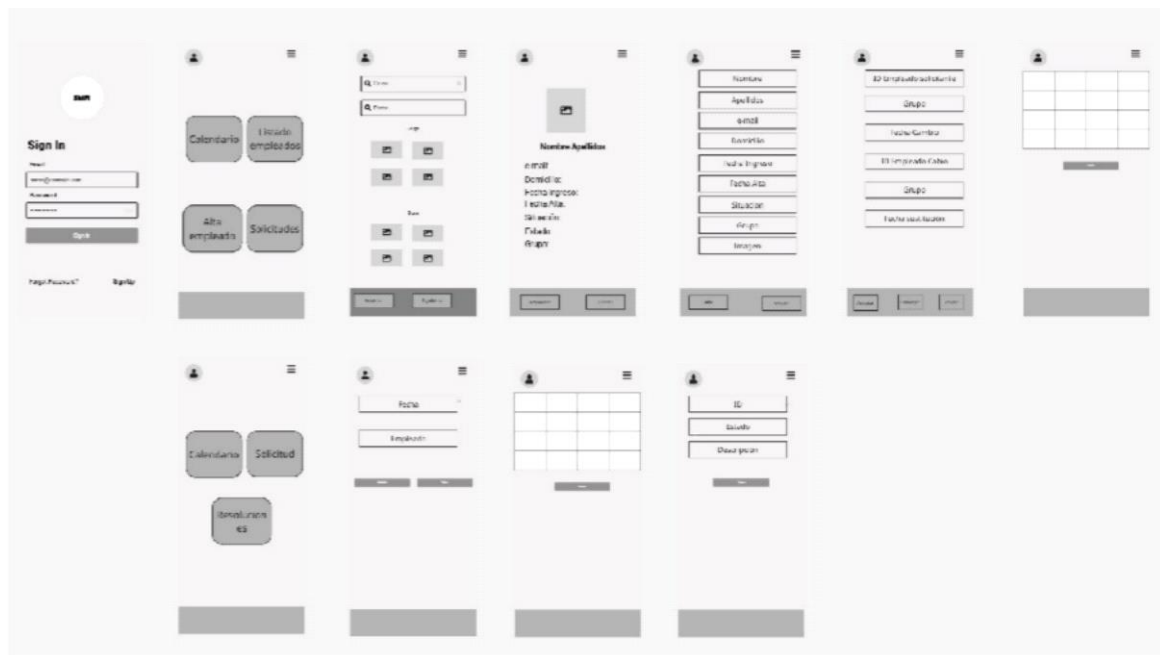
- El **administrador** accede a la sección de gestión de empleados.
- Selecciona un empleado de la lista.
- Verifica su estado actual (**activo/inactivo**).
- Si es necesario, cambia su estado.
- Asigna o cambia el grupo del empleado.
- El sistema actualiza la información en la base de datos.

Extensiones:

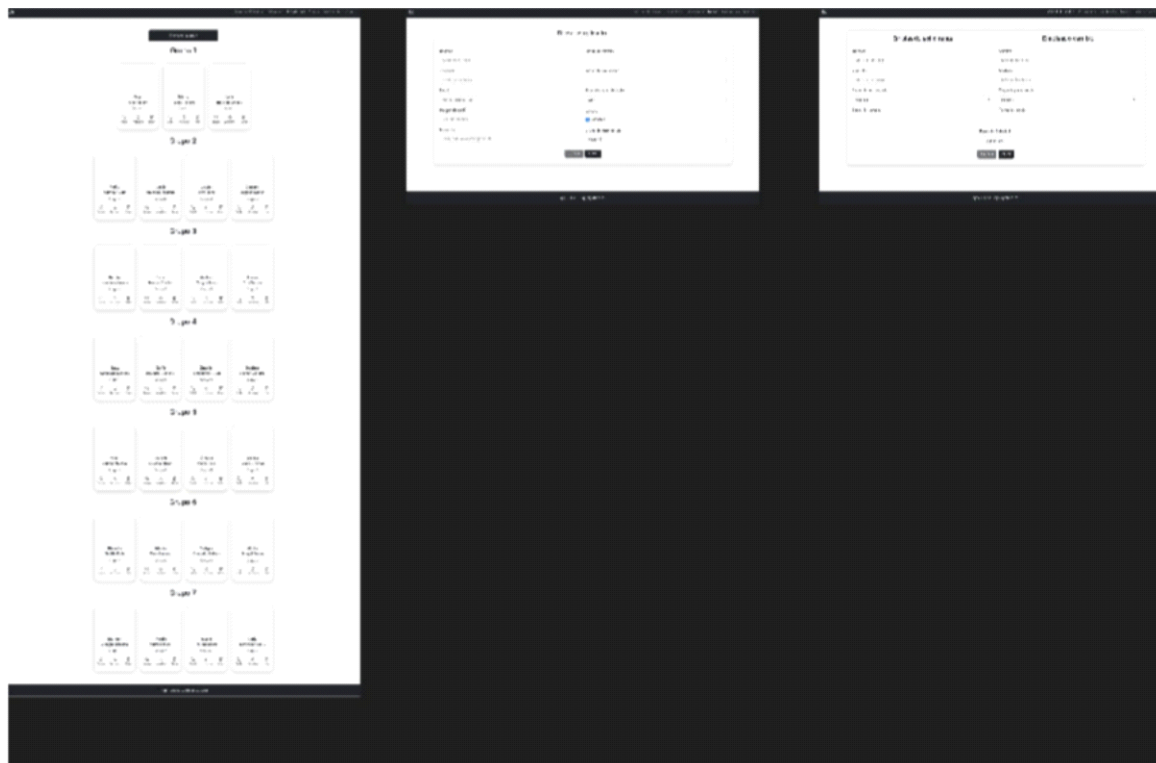
- Si el empleado está **inactivo**, el sistema impide su asignación a un grupo.



- **Diseño de interfaces:**
- **Wireframes:**



- Prototipo de interfaz de alta definición:



- **Logotipo:**

Elegimos este logotipo porque creemos que representa idea que tenemos sobre la gestion de tiempo y su importancia, teniendo en cuenta el sentido de urgencia con el

que un trabajador quiere obtener respuesta sobre sus peticiones de cambios en la jornada laboral.



- Planificación temporal y trabajo en equipo:

- Presupuesto temporal de tareas:

Tareas	Subtareas	Duración (horas)	Persona/s asignada/s
Planificación del Proyecto	Definir objetivos y alcance	2h	GRUPO
	Identificar recursos	1h	GRUPO
	Crear plan de proyecto	1h	GRUPO

	...		
Diseño de la Aplicación	Crear wireframes y maquetas	4h	CARLOS
	Diseñar arquitectura de BD	2h	ANA
	Definir experiencia de UX	1h	DANI
	...		
Desarrollo Frontend	Implementar interfaz de usuario	3h	CARLOS
	Integrar diseño responsivo	2h	ANA
	Realizar pruebas de usabilidad	3h	DANI
	...		
Desarrollo Backend	Configurar servidor y BD	8h	DANI
	Implementar lógica de negocio	4h	CARLOS
	Crear APIs y servicios web	4h	ANA
	...		
Integración y Pruebas	Integrar frontend y backend	4h	DANI
	Realizar pruebas funcionales	2h	CARLOS
	Corregir errores y optimizar	3h	ANA
	...		
Despliegue y Lanzamiento	Configurar entorno de producción	2h	ANA
	Desplegar la aplicación	2h	DANI
	Realizar pruebas finales	2h	CARLOS
	...		

- Organización de tareas y tiempos finales:

Tareas	Subtareas	Duración (horas)	Persona/s asignada/s
Planificación del Proyecto	Definir objetivos y alcance	2h	GRUPO
	Identificar recursos	1h	GRUPO
	Crear plan de proyecto	1h	GRUPO

	...		
Diseño de la Aplicación	Crear wireframes y maquetas	8h	CARLOS
	Diseñar arquitectura de BD	4h	ANA
	Definir experiencia de UX	2h	DANI
	...		
Desarrollo Frontend	Implementar interfaz de usuario	3h	CARLOS
	Integrar diseño responsivo	4h	ANA
	Realizar pruebas de usabilidad	3h	DANI
	...		
Desarrollo Backend	Configurar servidor y BD	10h	DANI
	Implementar lógica de negocio	4h	CARLOS
	Crear APIs y servicios web	3h	ANA
	...		
Integración y Pruebas	Integrar frontend y backend	6h	DANI
	Realizar pruebas funcionales	3h	CARLOS
	Corregir errores y optimizar	3h	ANA
	...		
Despliegue y Lanzamiento	Configurar entorno de producción	0h	ANA
	Desplegar la aplicación	0h	DANI
	Realizar pruebas finales	0h	CARLOS
	...		

- Conclusiones:

- **Análisis de desviaciones temporales y de tareas:**

En cuanto a la desviación temporal al realizar las tareas, el problema principal fue la subestimación de algunas áreas a desarrollar, causando así que tareas que en principio tenían un tiempo estipulado de finalización, acabaron alargándose más de la cuenta, quitando así, tiempo a otras áreas.

Este caso se presenta en el área de despliegue, en la cual, tanto por falta de conocimientos, como por falta de tiempo, se nos hizo imposible completar el despliegue a tiempo para la entrega. También en general, al desarrollar es común encontrar errores y refactorizar código, procesos que aumentan el tiempo necesario para completar algunas tareas.

- **Conclusiones generales del proyecto:**

De este proyecto podemos concluir que llevamos a cabo el desarrollo exitoso de una app cuyo valor reside en la consistencia y facilidad de uso que presenta a sus usuarios para gestionar sus turnos de manera eficaz.

Utilizando las tecnologías y conocimientos adquiridos en el grado, fuimos capaces de realizar el correcto desarrollo del gestor de turnos, siguiendo las pautas y metodologías impartidas, finalizando con un producto con buena estructura y potencial escalabilidad.

Para el futuro podemos implementar metodologías de planteamiento más eficaces y precisas, ya con un mejor entendimiento del tiempo que nos pueden llegar a tomar ciertas tareas, consiguiendo así una mejor estimación del trabajo necesario.

- **Limitaciones y prospectiva:**

Nuestro proyecto aún está en las fases iniciales del desarrollo, tiene mucho lugar a mejora y funcionalidades adicionales que nos gustaría implementar en el futuro.

Por ejemplo, la funcionalidad para permitir a los trabajadores consultar el detalle de su ficha, gestionar sus datos, y visualizar las horas trabajadas hasta la fecha.

Siguiendo la línea de visualización de horas trabajadas, queremos implementar un registro que guarde las horas trabajadas mensuales y anuales en función de la fecha o el userID.

Consideramos que nuestro servicio de gestor de turnos puede ser implementado tanto en PYMES cuyo propósito sea el de facilitar a los trabajadores una forma de consultar y modificar sus turnos, como para las grandes empresas en búsqueda de automatizar la gestión de turnos. Para realizar esto, debemos establecer un buen sistema de escalabilidad para poder soportar el mayor volumen de tráfico que recibiría la aplicación en esta circunstancia.

- **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>

ANEXO B.CÓDIGO FUENTE DE LA SOLUCIÓN Y PRUEBAS:

No hemos contado con el tiempo necesario para realizar las pruebas.

ANEXO C.MANUAL DE INSTALACIÓN - DESPLIEGUE:

Debido a la falta de los conocimientos adecuados para realizar el despliegue, así como la falta de tiempo para realizarlo, no hemos implementado esta función.