

**UNIR FP**

**Informática y Comunicaciones**

Técnico Superior en DAM

Título del Trabajo Fin de Estudios

|  |  |
| --- | --- |
| Trabajo fin de estudio presentado por: | García Marcos, Miguel Ángel |
| Tipo de trabajo: | Gestor de Proyectos |
| Tutor/a: | Soy Sualdea, Damián |
| Fecha: | 17/03/2024 |

Resumen

(Máximo 150 - 200 palabras)

La aplicación desarrollada tiene como objetivo la finalidad de centralizar y facilitar la gestión y supervisión de proyectos dentro de una organización. Permite a los usuarios supervisar en tiempo real el estado de cada proyecto, identificando en que fase se encuentra según las fechas establecidas. Además, proporciona una vista clara de los recursos destinados a cada proyecto, permitiendo una distribución eficiente del equipo de trabajo. Entre sus funcionalidades se incluyen la creación, edición o borrado de proyectos, fases y usuarios, así como el cálculo automático de los costes detallados y totales de cada proyecto, por separado y en conjunto. Ofrece herramientas de filtrado y búsqueda para mejorar el análisis de datos. En resumen, se trata de una plataforma adaptada a las necesidades de gestión de cualquier organización.

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

* Gestión de proyectos
* Usuarios
* Fases
* Costes
* Organización

Abstract

The developed application aims to centralize and facilitate the management and supervision of projects within an organization. It allows users to monitor the status of each project in real-time, identifying at what stage it is according to the established dates. Furthermore, it provides a clear view of the resources allocated to each project, enabling efficient distribution of the work team. Its functionalities include creating, editing, or deleting projects, phases, and users, as well as the automatic calculation of detailed and total costs for each project, separately and collectively. It offers filtering and search tools to enhance data analysis. In summary, it is a platform tailored to the management needs of any organization.

**Keywords:** (5 words max)

* Project management
* Users
* Phases
* Costs
* Organization

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 13](#_Toc180827992)

[5. Componentes del equipo y aportaciones realizadas por cada alumno 14](#_Toc180827993)

[5.1. Estudio de mercado 14](#_Toc180827994)

[5.1.1. Tabla comparativa de aplicaciones actualmente en el mercado 14](#_Toc180827995)

[5.1.2. Análisis DAFO de nuestra solución 16](#_Toc180827996)

[5.1.3. Subapartado 1.2 17](#_Toc180827997)

[5.2. Modelo de datos 17](#_Toc180827998)

[5.3. Diagramas UML 20](#_Toc180827999)

[5.3.1. Diagrama de clases 20](#_Toc180828000)

[5.3.2. Clasificación de usuarios 21](#_Toc180828001)

[5.3.3. Caso de uso 01 22](#_Toc180828002)

[5.4. Diseño de interfaces 24](#_Toc180828003)

[5.4.1. Wireframes 24](#_Toc180828004)

[5.4.2. Prototipo de interfaz de alta definición 25](#_Toc180828005)

[5.4.3. Paleta de colores 26](#_Toc180828006)

[5.4.4. Logotipo 27](#_Toc180828007)

[5.5. Planificación temporal y trabajo en equipo 28](#_Toc180828008)

[5.5.1. Presupuesto temporal de tareas 28](#_Toc180828009)

[5.5.2. Organización de tareas y tiempos finales 29](#_Toc180828010)

[5.5.3. Trabajo en equipo 30](#_Toc180828011)

[6. Conclusiones 31](#_Toc180828012)

[6.1. Análisis de desviaciones temporales y de tareas 31](#_Toc180828013)

[6.2. Conclusiones generales del proyecto 31](#_Toc180828014)

[6.2.1. Evaluación global del proyecto. 31](#_Toc180828015)

[6.2.2. Reflexión sobre el proceso de aprendizaje y desarrollo. 31](#_Toc180828016)

[6.2.3. Recomendaciones para futuros proyectos similares. 31](#_Toc180828017)

[6.3. Limitaciones y prospectiva 32](#_Toc180828018)

[6.3.1. Posibles mejoras y ampliaciones del proyecto. 32](#_Toc180828019)

[6.3.2. Nuevas líneas de investigación o desarrollo que podrían derivarse del proyecto. 32](#_Toc180828020)

[6.3.3. Sugerencias para la implementación en entornos reales. 32](#_Toc180828021)

[7. Referencias bibliográficas 33](#_Toc180828022)

[Anexo A. Diagramas de GANTT 34](#_Toc180828023)

[Anexo B. Código fuente de la solución y pruebas 35](#_Toc180828024)

[Anexo C. Manual de instalación - despliegue 36](#_Toc180828025)

[Anexo D. Documentación de la API 40](#_Toc180828026)

[Anexo E. Otros anexos de interés **¡Error! Marcador no definido.**](#_Toc180828027)

Índice de figuras

[Tabla 1: Herramientas, lenguajes, frameworks y APIs utilizadas 11](#_Toc180824791)

[Tabla 2 Presupuesto temporal de tareas **¡Error! Marcador no definido.**](#_Toc180824792)

[Tabla 3 Comparativa de aplicaciones actualmente en el mercado 15](#_Toc180824793)

[Ilustración 1 Análisis DAFO 16](#_Toc180824794)

[Ilustración 2 Diagrama E/R **¡Error! Marcador no definido.**](#_Toc180824795)

[Ilustración 3 Diagrama de clases 21](#_Toc180824796)

[Ilustración 4 Caso de uso "Recepción de pedido" **¡Error! Marcador no definido.**](#_Toc180824797)

[Ilustración 5 Wireframes 24](#_Toc180824798)

[Ilustración 6 Prototipo de interfaz de alta definición **¡Error! Marcador no definido.**](#_Toc180824799)

[Ilustración 7 Paleta de colores 26](#_Toc180824800)

[Ilustración 8 Logotipo en positivo 27](#_Toc180824801)

[Ilustración 9 Logotipo en negativo 27](#_Toc180824802)

Índice de tablas

[Tabla 1: Herramientas, lenguajes, frameworks y APIs utilizadas 11](#_Toc180824803)

[Tabla 2 Presupuesto temporal de tareas **¡Error! Marcador no definido.**](#_Toc180824804)

[Tabla 3 Comparativa de aplicaciones actualmente en el mercado 15](#_Toc180824805)

# Introducción

## Justificación

Es un gestor de proyectos que sirve para mejorar, ayudar y controlar mejor cada uno de los proyectos a nivel interno de una compañía.

Con esto podremos mejorar en lo siguiente:

* Mejora en los tiempos de gestión del proyecto
* Mayor control sobre los proyectos
* Al disminuir el tiempo en la gestión de proyectos, aumenta el tiempo de la persona que controle los proyectos, para dedicarlo a otras causas de la compañía.
* Mayor control en la pérdida y seguridad de los datos.
* Mayor control sobre los empleados y en que dedican su tiempo.

## Objetivos

Como comentaba en el apartado anterior al final los objetivos serán los siguientes:

* Mejora en los tiempos de gestión del proyecto
* Mayor control sobre los proyectos
* Al disminuir el tiempo en la gestión de proyectos, aumenta el tiempo de la persona que controle los proyectos, para dedicarlo a otras causas de la compañía.
* Mayor control en la pérdida y seguridad de los datos.
* Mayor control sobre los empleados y en que dedican su tiempo.

# Módulos formativos aplicados en el trabajo

Identifica los módulos formativos y resultados de aprendizaje aplicados en el presente trabajo (consultar legislación – puedes pedir ayuda a tu tutor)

**Programación**

* Desarrollo de lógica de aplicación (CRUD, validaciones, control de errores).
* Uso de estructuras de datos y algoritmos básicos.

**Bases de Datos**

* Diseño de bases de datos relacionales (tablas, claves foráneas).
* Consulta y manipulación de datos con SQL (JOIN, INSERT, UPDATE...).

**Entornos de desarrollo**

* Uso de entornos como Visual Studio Code y control de versiones (Git).
* Pruebas y depuración del software.

**Desarrollo de Interfaces**

* Implementación de interfaces gráficas web usando React.
* Diseño adaptado a la experiencia de usuario (UX/UI).

**Acceso a Datos**

* Interacción entre la lógica de negocio y la base de datos a través del backend (API REST con Node.js).

**Sistemas de gestión empresarial (opcional)**

* Relación con la gestión de tareas y equipos en entornos reales.

# Herramientas y lenguajes utilizados

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

Tabla : Herramientas, lenguajes, frameworks y APIs utilizadas

|  |  |
| --- | --- |
| **Git**  Git, social media icon - Free download on Iconfinder | 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). |
| **IntelliJ Idea** | IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para Java, desarrollado por JetBrains. Ofrece soporte para el desarrollo en Spring Boot, ayudando en la programación backend gracias a su sistema de autocompletado, depuración avanzada y herramientas de gestión de proyectos. |
| Visual Studio Code (VSCode) | VSCode es un editor de código fuente ligero desarrollado por Microsoft. Fue utilizado para el desarrollo de la parte Frontend en React.js, permitiendo una rápida edición, instalación de extensiones y control de versiones. |
| React.js | React es una biblioteca de JavaScript de código abierto para construir interfaces de usuario. Desarrollada por Facebook, facilita la creación de componentes reutilizables y la gestión eficiente del DOM virtual. |
| Spring Boot | Spring Boot es un framework de Java que simplifica la creación de aplicaciones backend. Facilita la configuración automática, el manejo de dependencias y la creación de APIs REST de forma rápida y robusta. |
| MySQL | MySQL es un sistema de gestión de bases de datos relacional popular en entornos de desarrollo y producción. Se utilizó para almacenar toda la información de los proyectos y fases de manera estructurada. |
| DBeaver | DBeaver es una herramienta de administración de bases de datos que permite gestionar, consultar y visualizar bases de datos como MySQL y PostgreSQL de forma gráfica. Fue utilizado para la gestión de tablas, consultas SQL y depuración de datos. |
| npm (Node Package Manager) | npm es el sistema de gestión de paquetes de Node.js. Se utilizó para instalar las dependencias necesarias para el proyecto Frontend, como React Icons, React Router, Bootstrap, etc. |
| Bootstrap | Bootstrap es un framework de CSS que permite el diseño de aplicaciones web responsivas y estilizadas de forma rápida. Se utilizó para estructurar el Frontend de manera ordenada. |
| Spring Data JPA | Es una parte de Spring que facilita el acceso a bases de datos mediante la abstracción de repositorios. Permite realizar operaciones CRUD sin necesidad de implementar consultas SQL manuales. |
| Postman | Postman es una herramienta que permite realizar pruebas de APIs REST. Se usó para probar manualmente los endpoints creados en el backend, enviando solicitudes GET, POST, PUT y DELETE. |

# Metodologías utilizadas

Dado que este proyecto ha sido realizado de forma individual y en paralelo con responsabilidades laborales, no se ha seguido una metodología de desarrollo en específico como Scrum o Kanban. Sin embargo, se ha trabajado de manera iterativa y adaptativa, resolviendo cada parte del sistema por fases según el tiempo disponible y la complejidad de cada componente.

El enfoque fue modular, desarrollando primero la base de datos, luego una base de la API backend y finalmente combinando el Front con API backend. Se priorizó tener funcionalidad completas y probadas antes de avanzar.

Aunque no se emplearon herramientas de gestión de proyectos, se utilizaron listas de tareas personales y controles de versiones con Git para mantener el orden y coherencia del desarrollo.

Este enfoque mas flexible ha permitido compatibilizar el trabajo profesional con el avance del TFC, logrando una solución funcional y estructurada dentro de estas limitaciones.

# Componentes del equipo y aportaciones realizadas por cada alumno

Este proyecto ha sido desarrollado íntegramente de forma individual.Esto ha implicado asumir todos los roles del proyecto como análisis, diseño, desarrollo y pruebas.

| **Rol asumido** | **Tareas realizadas** |
| --- | --- |
| Analista | Estudio de necesidades, definición de funcionalidades, modelado de datos |
| Diseñador | Diseño de interfaces, elección de paleta de colores y estructura visual |
| Desarrollador Backend | Programación de la API REST, conexión con la base de datos, lógica de negocio |
| Desarrollador Frontend | Implementación en React, interactividad, gestión de estados y validaciones |
| Tester | Pruebas funcionales, revisión de errores, mejoras iterativas |

## 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 Comparativa de aplicaciones actualmente en el mercado

| **Característica** | **Trello** | **Asana** | **Taskmanager** |
| --- | --- | --- | --- |
| **Descripción** | App de gestión visual de tareas | App de planificación de proyectos | Web app para gestionar proyectos |
| **Funcionalidades** | Tableros, listas, tarjetas | Tareas, cronogramas, dependencias | CRUD proyectos, fases y usuarios |
| **Público Objetivo** | Equipos pequeños / medianos | Empresas de tamaño medio | PYMEs, proyectos académicos |
| **Plataformas** | Web, iOS, Android | Web, iOS, Android | |  | | --- | |  |  |  | | --- | | Web | |
| **Precio** | Freemium | |  | | --- | |  |  |  | | --- | | Gratuito limitado / premium | | Gratuita |
| **Puntuación** | 4.5 / 5.0 | 4.6 / 5.0 | 4.7 / 5.0 (estimada) |

### Análisis DAFO de nuestra solución

Ilustración Análisis DAFO

**Debilidades**

* Requiere conexión a Internet para su funcionamiento.
* No cuenta con una aplicación móvil nativa que permita acceder desde smartphones.
* La interfaz es funcional pero menos pulida que otras soluciones comerciales más maduras.

**Amenazas**

* Alta competencia en el mercado con herramientas consolidadas como Trello, Asana o Microsoft Project.
* Posibles vulnerabilidades de seguridad si la aplicación no se actualiza regularmente.
* Evolución constante de las tecnologías y frameworks, lo que exige mantenimiento continuo.

**Fortalezas**

* Interfaz sencilla e intuitiva, fácilmente personalizable según las necesidades del usuario.
* Ofrece control detallado sobre usuarios, fases del proyecto y costes asociados.
* Adecuada para pequeñas empresas, centros educativos o equipos reducidos dentro de organizaciones.

**Oportunidades**

* Potencial para desarrollar una versión móvil multiplataforma.
* Posibilidad de implementar gráficos interactivos para visualización de datos.
* Capacidad de evolucionar hacia un producto comercial escalable o modelo SaaS.

### Subapartado 1.2

Texto Normal del menú de estilos.

## Modelo de datos

Diagrama E/R:

Interfaz de usuario gráfica, Diagrama

El contenido generado por IA puede ser incorrecto.

**Tabla Usuarios:**

Contiene la información de cada usuario del sistema.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Tabla proyectos:**

Representa cada proyecto registrado en el sistema.

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

**Tabla usuarios\_proyectos:**

Asigna uno o varios usuarios a cada proyecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

**Tabla fases\_proyecto:**

Define las fases que componen un proyecto, con sus fechas y la duración de cada una de ellas.

Captura de pantalla con la imagen de una pantalla

El contenido generado por IA puede ser incorrecto.

**Tabla usuarios\_roles:**

Tabla intermedia para gestionar la relación entre usuarios y roles. Ya que un usuario puede tener varios roles y varios roles pueden ser gestionados por un usuario.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Tabla roles:**

Esta tabla almacena los diferentes tipos de roles que puede tener un usuario del sistema.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Diagramas UML

Los diagramas UML (Unified Modeling Language) son una herramienta visual que permite representar el diseño y estructura de un sistema informático. Son esenciales para documentar cómo interactúan las distintas partes del software, tanto desde el punto de vista estructural (diagrama de clases) como funcional (casos de uso).

En este proyecto, se han aportado los siguientes diagramas:

* Diagrama de clases
* Diagrama de casos de uso

### Diagrama de clases

El diagrama de clases representa la estructura de las entidades del sistema, sus atributos y relaciones. Las clases son:

* Usuario: contiene atributos como id, nombre, email y password.
* Rol: define los distintos tipos de roles que puede tener un usuario.
* Proyecto: agrupa la información relacionada con los proyectos, como su nombre, costes, fase\_actual y su relación con el usuario creador del proyecto.
* Fases Proyecto: representa cada etapa del proyecto, con fechas de inicio y fin de cada una de las fases, y su duración.

**Relaciones:**

* Usuario y Proyecto: Muchos a muchos (a través de tabla intermedia).
* Usuario y Rol: Muchos a muchos
* Proyecto y Fase Proyecto: Uno a muchos.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ilustración Diagrama de clases

### Clasificación de usuarios

En nuestra aplicación se contemplan distintos tipos de usuarios, cada uno definido con un rol.

Para comprender mejor estos perfiles, se ha aplicado la metodología Persona del enfoque Design Thinking, la cual permite representar a los usuarios mediante estereotipos realistas basados en sus comportamientos.

Jefe de Proyecto:

* **Objetivo**: Tener una visión clara del avance de cada proyecto.
* **Funciones clave**: Crear proyectos, asignar fases y usuarios, controlar costes.
* **Necesidades**: Informes rápidos, interfaz clara, control del equipo.

Desarrollador:

* **Objetivo**: Saber en que proyectos está trabajando y en qué fases.
* **Funciones clave**: Consultar proyectos asignados y tareas en función de las fases.
* **Necesidades**: Acceso sencillo y rápido, sin mucha información.

Administrador del sistema:

* **Objetivo**: Gestionar la aplicación, nuevos usuarios, roles y mantener la seguridad.
* **Funciones clave**: Control de accesos y creación de usuarios, roles.
* **Necesidades**: Seguridad y trazabilidad del sistema.

### Caso de uso 01

**Gestión de usuarios**

* **Login**: autenticación mediante correo y contraseña.
* **Recuperación de contraseña**: si un usuario olvida la contraseña, el sistema genera una nueva automáticamente y la envía por correo. Esta contraseña se actualiza directamente en la base de datos.
* **Crear / Editar / Borrar usuarios**.
* **Asignar roles** a cada usuario (por ejemplo: Jefe de Proyecto, Desarrollador, QA, etc.).

Diagrama

El contenido generado por IA puede ser incorrecto.

**Gestión de proyectos**

* **Crear / Editar / Borrar proyectos**.
* **Asignar usuarios** a un proyecto.
* **Visualizar usuarios asignados**.
* **Consultar el coste total** (calculado automáticamente como suma de coste interno y externo).

Diagrama

El contenido generado por IA puede ser incorrecto.

**Gestión de fases**

* **Crear / Editar / Borrar fases** para cada proyecto.
* **Control automático de la fase actual** según la fecha actual.
* **Visualización por proyecto** de todas las fases asociadas.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Filtrado y consulta**

* Filtrar proyectos por:
  + Nombre, presupuesto, costes, fase actual, etc.
* Filtrar usuarios por:
  + Nombre, correo electrónico, roles, etc.
* Consulta agregada de **costes totales** de todos los proyectos.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Sesiones**

* **Logout**: cierre seguro de sesión.

## Diseño de interfaces

### Wireframes

Pizarrón blanco con texto en letras negras sobre fondo blanco

El contenido generado por IA puede ser incorrecto.

Ilustración Wireframes

### Prototipo de interfaz de alta definición

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

### Paleta de colores

He utilizado esta paleta de colores, ya que, en su conjunto, ofrece equilibrio visual, usabilidad y una apariencia profesional, facilitando así la experiencia de usuario.

Aplicación

El contenido generado por IA puede ser incorrecto.

Ilustración Paleta de colores

### Logotipo

Icono

El contenido generado por IA puede ser incorrecto.

Ilustración Logotipo en positivo

Logotipo en negativo

Logotipo

El contenido generado por IA puede ser incorrecto.

Ilustración Logotipo en negativo

## 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 | 4h | Miguel |
| |  | | --- | |  |  |  | | --- | | Estudio de mercado y análisis de competencia | | 3h | Miguel |
| Análisis DAFO y casos de uso | 2h | Miguel |
| **Diseño de la Aplicación** | Diseño arquitectura de BD | 6h | Miguel |
| Creación de diagramas UML | 3h | Miguel |
| Bocetos y diseño UI | 4h | Miguel |
| **Desarrollo Frontend** | Implementar interfaz de usuario | 18h | Miguel |
| Gestión de estado y eventos | 5h | Miguel |
| Realizar pruebas de usabilidad | 1h | Miguel |
| **Desarrollo Backend** | Configurar servidor y BD | 8h | Miguel |
| Implementar lógica de negocio | 8h | Miguel |
| Crear APIs y servicios web | 8h | Miguel |
| **Integración y Pruebas** | Integrar frontend y backend | 8h | Miguel |
| Realizar pruebas funcionales | 4h | Miguel |
| Corregir errores y optimizar | 4h | Miguel |
| **Despliegue y Lanzamiento** | Configurar entorno de producción | 1h | Miguel |
| Desplegar la aplicación | 1h | Miguel |
| Realizar pruebas finales | 1h | Miguel |

### Organización de tareas y tiempos finales

|  |  |  |  |
| --- | --- | --- | --- |
| **Tareas** | **Subtareas** | **Duración (horas)** | **Persona/s asignada/s** |
| **Planificación del Proyecto** | Definir objetivos y alcance | 4h | Miguel |
| |  | | --- | |  |  |  | | --- | | Estudio de mercado y análisis de competencia | | 2h | Miguel |
| Análisis DAFO y casos de uso | 2h | Miguel |
| **Diseño de la Aplicación** | Diseño arquitectura de BD | 5h | Miguel |
| Creación de diagramas UML | 3h | Miguel |
| Bocetos y diseño UI | 4h | Miguel |
| **Desarrollo Frontend** | Implementar interfaz de usuario | 16h | Miguel |
| Gestión de estado y eventos | 5h | Miguel |
| Realizar pruebas de usabilidad | 2h | Miguel |
| **Desarrollo Backend** | Configurar servidor y BD | 8h | Miguel |
| Implementar lógica de negocio | 8h | Miguel |
| Crear APIs y servicios web | 6h | Miguel |
| **Integración y Pruebas** | Integrar frontend y backend | 3h | Miguel |
| Realizar pruebas funcionales | 4h | Miguel |
| Corregir errores y optimizar | 4h | Miguel |
| **Despliegue y Lanzamiento** | Configurar entorno de producción | 1h | Miguel |
| Desplegar la aplicación | 1h | Miguel |
| Realizar pruebas finales | 1h | Miguel |

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

### Trabajo en equipo

Gráfico de aportaciones:

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

# 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.

La evaluación general es bastante positiva. A pesar de que el desarrollo se ha realizado de forma individual y con limitaciones de tiempo por motivos laborales, se ha conseguido completar los objetivos planteados inicialmente. Cumple con los criterios básicos de una aplicación CRUD moderna. El sistema es estable y escalable para seguir con su futuro desarrollo.

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

Me ha permitido consolidar conocimientos y también aprender nuevas tecnologías como React, ya que hemos podido ver muy poco de esta parte en clase. He aprendido a diseñar una aplicación desde cero, juntando todos los conocimientos del ciclo formativo.

También he aprendido a valorar el esfuerzo que conlleva realizar un desarrollo completo de forma autónoma.

### Recomendaciones para futuros proyectos similares.

Para futuros proyectos similares, recomendaría realizar una planificación de todo por horas de desarrollo frente a las horas disponibles. Documentar a la misma vez que se va realizando la aplicación y se va desarrollando, tanto en código como realizando la documentación oportuna del proyecto. Es muy recomendable, mantener una estructura clara desde el principio tanto a nivel de carpetas como de la lógica de código, para facilitar enormemente el mantenimiento y futuras ampliaciones.

## Limitaciones y prospectiva

### Posibles mejoras y ampliaciones del proyecto.

* Desarrollo de una aplicación móvil nativa para Android/iOS.
* Gestión de permisos según el rol de usuario.
* Exportación de informes en Excel, útiles para reportes de documentación.
* Dashboard visual con gráficas dinámicas, para representar costes y progreso por fase de manera intuitiva.
* Notificaciones por correo ante cambios en fases o asignaciones de usuario.
* Nuevo estado, sobre en que estado se encuentra el proyecto.
* Comentarios en el dashboard, con fechas para hacer seguimiento.
* Mejora de la seguridad con contraseñas cifradas.
* Añadir nuevo módulo para guardar la documentación de la realización de pruebas, tanto las de desarrollo como las de usuarios.

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

* Análisis de rendimiento de equipos: A través de métricas de tiempo invertido por fase y usuario.
* Integración con servicios cloud (Google Drive, Dropbox) para almacenar documentación del proyecto directamente desde la aplicación.

### Sugerencias para la implementación en entornos reales.

* Alojamiento seguro en servidores profesionales con acceso restringido y políticas de backup automático.
* Cifrado robusto de contraseñas y comunicaciones (HTTPS, JWT, Bcrypt) para garantizar la protección de datos personales.
* Documentación técnica completa para facilitar el mantenimiento y posibles integraciones futuras.
* Formación básica para los usuarios finales sobre el uso del sistema y flujos de trabajo.
* Sistema de roles bien definido con validaciones en backend para evitar accesos indebidos.

# 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

MySQL. (2024). *MySQL Reference Manual*. Recuperado de <https://dev.mysql.com/doc/>

**React.js. (2024). *Documentación oficial*. Recuperado de** [**https://reactjs.org**](https://reactjs.org)

Oracle. (2024). *Documentación oficial de Java SE*. Recuperado de <https://docs.oracle.com/javase/>

Microsoft. (2024). *Visual Studio Code Documentation*. Recuperado de <https://code.visualstudio.com/docs>

1. Diagramas de GANTT

**Gráfico Temporal:**

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

**Gráfico Final:**

Gráfico

El contenido generado por IA puede ser incorrecto.

1. Código fuente de la solución y pruebas

Pruebas realizadas desde el Front y desde Postman:

* Comprobación del login, tanto OK, como FAIL.
* Recuperación de contraseña.
* Alta usuario.
* Modificar usuario.
* Eliminar usuario.
* Alta proyecto.
* Modificar proyecto.
* Eliminar proyecto. (Probar también con uno que tenga fases para que se elimine todo, incluso si hay usuarios asignados a ese proyecto).
* Crear fase.
* Modificar fase.
* Eliminar fase.
* Comprobación del dashboard. (Ver que coinciden los proyectos, activos y sus fases).

1. Manual de instalación – despliegue
2. Añadir archivo Dockerfile en el back.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

1. Añadir archivo Dockerfile en el front.

Captura de pantalla con la imagen de una pantalla

El contenido generado por IA puede ser incorrecto.

1. Construir de nuevo el proyecto taskmanager del back, para generar el .jar.

Texto

El contenido generado por IA puede ser incorrecto.

1. Crear archivo .yml

Texto

El contenido generado por IA puede ser incorrecto.

1. Construir y levantar el proyecto completo

Texto

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

6. Ejecutamos el script de la creación de la base de datos, en DBeaver.

1. Documentación de la API

**Usuarios:**

* GET 🡪 (/usuario/getAll): Listar todos los usuarios.
* POST 🡪 (/usuario/add): Añadir un nuevo usuario.
* POST 🡪 (/usuario/addWithRol): Añade un usuario y le asigna un rol.
* POST 🡪 (/usuario/login): Autentica un usuario.
* POST 🡪 (/usuario/recuperar): Envía nueva contraseña por correo.
* PUT 🡪 (/usuario/update/{id}): Actualiza el usuario.
* DELETE 🡪 (/usuario/delete/{id}): Elimina un usuario.

**Proyectos:**

* GET 🡪 (/proyecto/dashboard): Listar todos los proyectos.
* POST 🡪 (/proyecto /add?usuarioId=1): Crea un proyecto asociado a un usuario.
* POST 🡪 (/proyecto /asignarUsuario): Asigna un usuario a un proyecto.
* POST 🡪 (/proyecto /eliminarUsuario): Elimina un usuario de un proyecto.
* PUT 🡪 (/proyecto /update/{id}): Actualiza el proyecto.
* DELETE 🡪 (/proyecto /delete/{id}): Elimina un usuario.

**Fases del Proyecto:**

* GET 🡪 (/fases /nombres): Obtiene los nombres de las fases.
* POST 🡪 (/fases /add): Crea una nueva fase.
* PUT 🡪 (/fases /update/{id}): Actualiza una fase existente.
* DELETE 🡪 (/fases /delete/{id}): Elimina una fase.

**Roles:**

* GET 🡪 (/rol/getAll): Listar todos los roles.