

**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 tiene como objetivo facilitar la supervisión y gestión de proyectos dentro de una organización. Permite llevar el control de:

* En qué **fase** se encuentra cada proyecto.
* Qué **usuarios** están asignados a cada proyecto.
* Qué proyectos están siendo o han sido **realizados**.
* Cuáles son los **costes** asociados a cada proyecto.
* Cuáles son la suma total de los **costes** de los proyectos.

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

Abstract

(150 - 200 words max)

**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

## 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)

# Herramientas y lenguajes utilizados

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

Tabla 1: 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

Describe la metodología de desarrollo de proyecto aplicada

# Componentes del equipo y aportaciones realizadas por cada alumno

Lo he realizado solo.

## 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** | **FinTech Pro** | **MoneyMaster** | **FinanceGuru** |
| --- | --- | --- | --- |
| **Descripción** | Una app para gestionar inversiones y portafolios. | Una app para el seguimiento de gastos y presupuestos. | Una app para asesoramiento financiero personalizado. |
| **Funcionalidades** | Análisis de mercado, alertas de inversión, informes de rendimiento. | Registro de gastos, creación de presupuestos, gráficos de gastos. | Asesoramiento financiero, simulaciones de inversión, análisis de riesgo. |
| **Público Objetivo** | Inversores y gestores de portafolios. | Personas que desean controlar sus finanzas diarias. | Personas que buscan asesoramiento financiero profesional. |
| **Plataformas** | iOS, Android. | iOS, Android. | iOS, Android. |
| **Precio** | Suscripción mensual. | Gratis con compras dentro de la app. | Suscripción anual. |
| **Puntuación** | 4.6/5.0 | 4.4/5.0 | 4.7/5.0 |

### Análisis DAFO de nuestra solución

Ilustración 1 Análisis DAFO

Describe las diferentes debilidades, amenazas, fortalezas y oportunidades

### Subapartado 1.2

Texto Normal del menú de estilos.

## Modelo de datos

Diagrama E/R:

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 roles:**

Define los distintos tipos de roles disponibles en el sistema.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Tabla usuarios\_roles:**

Relaciona usuarios con uno o varios roles.

Interfaz de usuario gráfica, Aplicación

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.

## 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

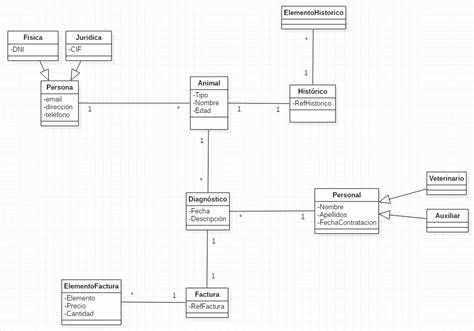


Ilustración 3 Diagrama de clases

### Clasificación de usuarios

Explicación de los tipos de usuario y perfiles del sistema desarrollado

Utilizar el método Persona (UX) (Design Thinking Services, 2023)

### 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.).

**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).

**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.

**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.

**Sesiones**

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

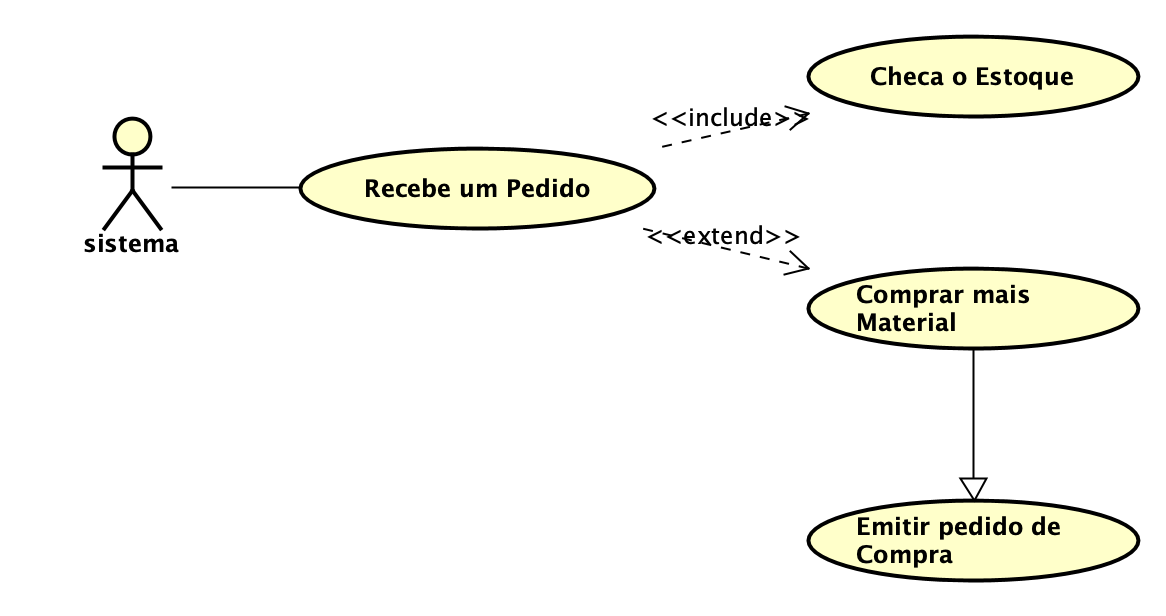


Ilustración 4 Caso de uso "Recepción de pedido"

## Diseño de interfaces

### Wireframes

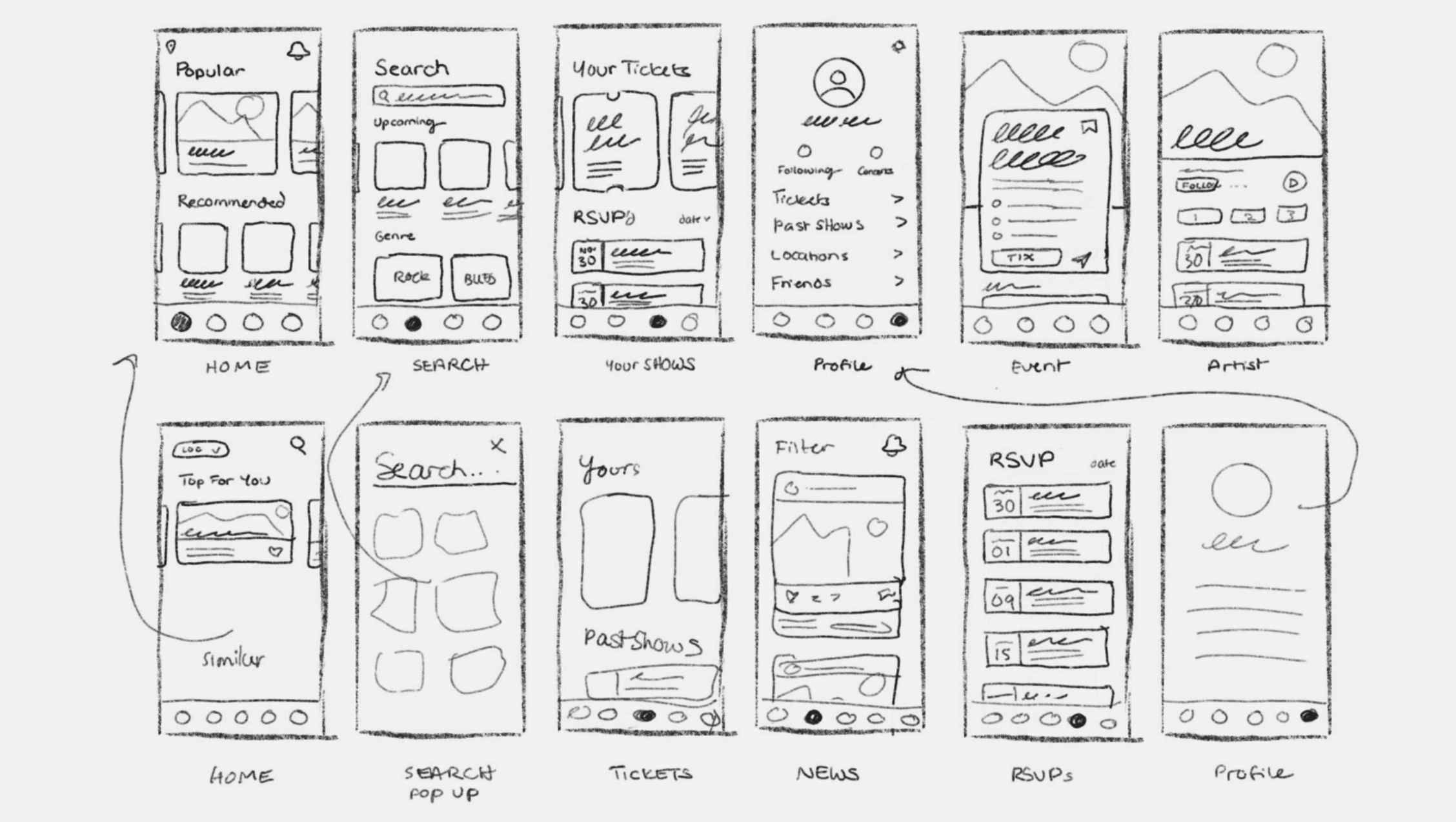


Ilustración 5 Wireframes

### Prototipo de interfaz de alta definición

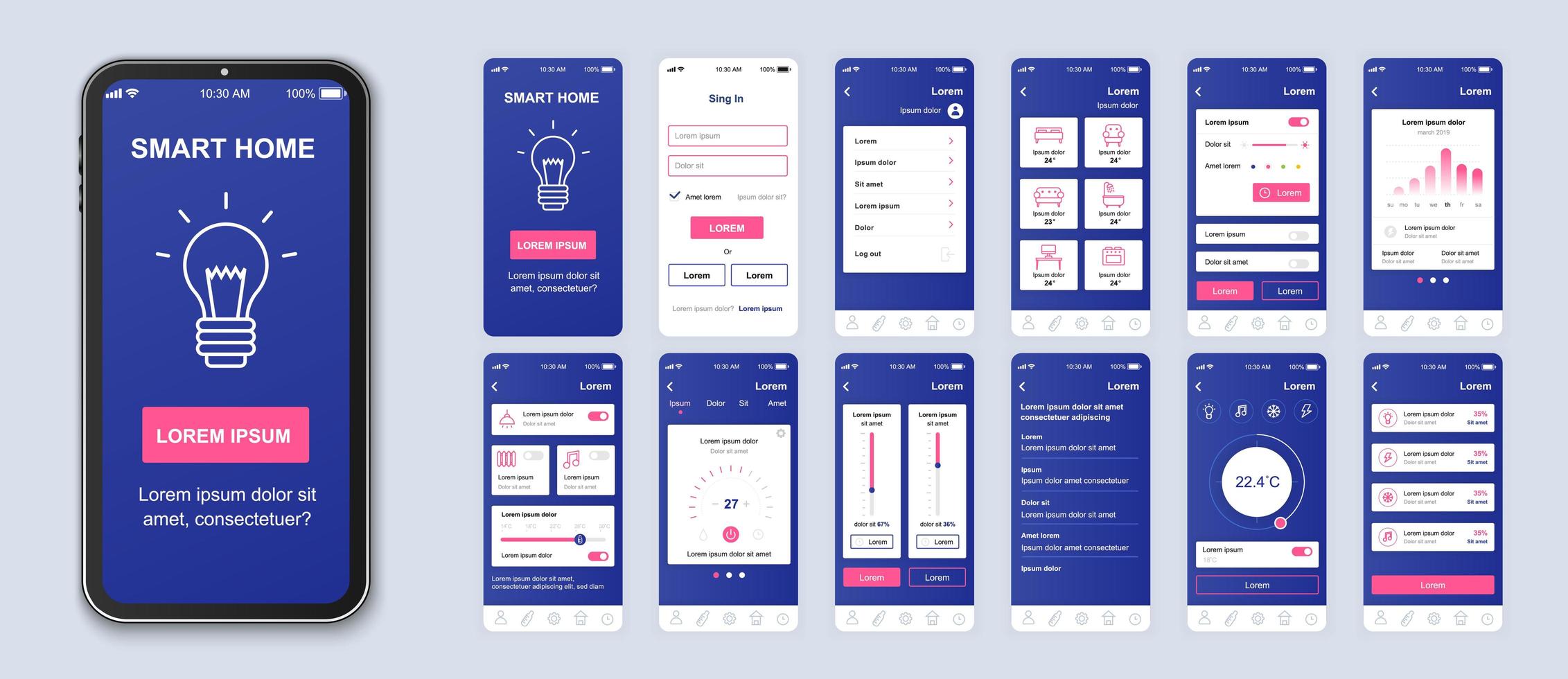


Ilustración 6 Prototipo de interfaz de alta definición

### 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.

### 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

Proyectos(/proyecto):

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.