

# UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software – Sección 30

Ing. Cristian Muralles



## Proyecto: MoneyFlow

Osman Emanuel de León García – 23428

Milton Giovanni Polanco Serrano - 23471

Gadiel Amir Ocaña Véliz - 231270

Guatemala, 20 de noviembre de 2024

## Introducción

El proyecto está dirigido a personas que buscan mejorar la gestión de sus finanzas personales. No está vinculado a una entidad específica, sino que se enfoca en usuarios individuales con diferentes perfiles, como jóvenes profesionales, emprendedores, amas de casa y estudiantes. Este grupo de usuarios enfrenta desafíos comunes, como la falta de control sobre sus gastos, la dificultad para planificar ahorros y la necesidad de herramientas que les permitan visualizar mejor su situación financiera. La aplicación se desarrollará con una estructura flexible y escalable, permitiendo adaptaciones futuras según las necesidades de los usuarios.

## Descripción de la idea

La idea central del proyecto es la creación de una aplicación web que permita a los usuarios registrar sus ingresos y gastos, categorizarlos y analizar patrones de consumo mediante gráficos interactivos y reportes financieros. Esta solución se basa en la problemática de la falta de educación financiera y herramientas accesibles para la gestión del dinero, lo que afecta la estabilidad económica de muchas personas.

La aplicación contará con funcionalidades clave, como la exportación de datos en CSV/PDF, la posibilidad de establecer presupuestos y metas de ahorro, y la evaluación de integraciones con APIs bancarias para la importación automática de transacciones. Además, el desarrollo se realizará utilizando tecnologías modernas como React para el frontend y PostgreSQL para la gestión de datos.

El enfoque metodológico se basará en Design Thinking, asegurando que la solución esté alineada con las necesidades reales de los usuarios, y en Scrum, permitiendo una implementación iterativa y eficiente.

## Objetivos Generales y Específicos

### Objetivo General

Desarrollar una aplicación web intuitiva y eficiente para la gestión de finanzas personales, que permita a los usuarios registrar, analizar y optimizar sus ingresos y gastos, fomentando la toma de decisiones financieras más informadas.

### Objetivos Específicos

1. Diseñar una interfaz responsiva e intuitiva con React, que permita una experiencia de usuario fluida en diferentes dispositivos.
2. Implementar una base de datos en PostgreSQL para almacenar y gestionar la información financiera de manera segura y confiable.
3. Permitir la clasificación de transacciones en categorías personalizadas, facilitando el análisis de ingresos y gastos.
4. Desarrollar módulos para la creación y seguimiento de presupuestos y metas de ahorro.
5. Integrar gráficos y reportes financieros interactivos que ayuden a los usuarios a identificar patrones de consumo.

6. Evaluar la posible integración de APIs bancarias para la importación automática de transacciones.
7. Implementar la opción de exportar reportes en formatos CSV y PDF, permitiendo su uso en análisis externos.
8. Aplicar metodologías ágiles como Scrum para garantizar el desarrollo iterativo y adaptativo del proyecto.
9. Documentar todo el proceso de desarrollo y validar la solución con pruebas de usuario, asegurando que cumple con los requerimientos identificados.

## Brief

El proyecto consiste en el desarrollo de una aplicación web para la gestión de finanzas personales, diseñada para ayudar a los usuarios a registrar, analizar y optimizar su administración financiera. La aplicación busca atender a estudiantes universitarios y trabajadores, permitiéndoles gestionar sus ingresos, gastos y cuentas de forma eficiente. El problema central es la falta de herramientas simples y efectivas para la organización de finanzas en personas que están en diferentes etapas de independencia económica. La solución debe ser intuitiva, adaptada a las necesidades específicas de cada usuario y permitir la consolidación de información financiera en un solo lugar.

## Selección de Usuarios

### *Usuarios Externos*

Estos usuarios son el público objetivo principal y beneficiarios directos del sistema. Los perfiles de manera más detallada incluyen:

1. *Adultos jóvenes y profesionales independientes:*
  - Personas entre 20 y 35 años que buscan mejorar el control de sus ingresos y gastos.
  - Necesitan herramientas para presupuestar, ahorrar y optimizar sus hábitos de consumo.
2. *Familias y amas de casa:*
  - Encargados de gestionar los gastos del hogar.
  - Usarán la aplicación para clasificar transacciones familiares, establecer límites de gasto y planificar ahorros a largo plazo.
3. *Pequeños emprendedores o freelancers:*
  - Personas que combinan sus finanzas personales con las de sus actividades comerciales.
  - Requieren funcionalidades para separar y analizar sus ingresos y egresos en diferentes categorías.
4. *Estudiantes:*
  - Jóvenes que están aprendiendo a manejar su dinero.

- Buscan herramientas simples y didácticas para ahorrar y establecer metas financieras.

La selección de estos usuarios responde a la necesidad de abordar diferentes niveles de independencia financiera y problemas comunes en la gestión del dinero. Se han elegido estos usuarios porque:

- Representan a una gran parte de la población que enfrenta dificultades en la administración financiera.
- Cada perfil tiene necesidades específicas que pueden resolverse con la aplicación.
  - Desde aprender a manejar un presupuesto hasta consolidar múltiples cuentas y tarjetas.
- Permiten desarrollar una solución adaptable a distintos niveles de experiencia financiera.
  - Interfaz sencilla para principiantes y herramientas avanzadas para quienes manejan varias cuentas.
- La aplicación puede escalar y expandirse en función de estos perfiles, permitiendo mejoras futuras.

Este enfoque garantiza que la aplicación sea útil para diferentes tipos de usuarios, desde quienes buscan aprender a administrar su dinero hasta aquellos que necesitan automatización y análisis financiero detallado.

### **Selección de Usuarios Extremos**

Dentro de los usuarios seleccionados, se incluyen casos extremos que pueden presentar desafíos adicionales en el uso de la aplicación, pero cuya experiencia puede aportar información valiosa sobre la flexibilidad y utilidad del sistema:

1. Usuarios con ingresos irregulares (por trabajos informales o autónomos).
2. Usuarios con múltiples tarjetas o cuentas bancarias que requieren consolidación de información.
3. Usuarios sin conocimientos financieros que buscan una interfaz sencilla y educativa.

### **Evidencia de la Técnica "Qué, Cómo y Por Qué"**

La técnica "Qué, Cómo y Por Qué" nos permite descomponer y analizar las actividades principales relacionadas con el uso de la aplicación web para la gestión de finanzas personales. Se ha aplicado la técnica "Qué, Cómo y Por Qué" a cada uno de los perfiles definidos para comprender mejor sus necesidades y comportamientos.

### **Perfil 1: Estudiantes Universitarios Dependientes Económicamente**

- **Qué:**
  - Reciben dinero de sus padres mensualmente o en fechas específicas.
  - Gastan en comida, transporte, entretenimiento y educación.
  - No tienen ingresos propios y dependen de la asignación familiar.

- **Cómo:**
  - Registran manualmente sus ingresos y gastos en notas o aplicaciones básicas.
  - Tienen dificultades para prever cuánto dinero les quedará a fin de mes.
  - Priorizan gastos a corto plazo sin planificación financiera.
- **Por Qué:**
  - Necesitan evitar quedarse sin dinero antes de recibir la siguiente asignación.
  - Buscan herramientas que les ayuden a visualizar su presupuesto sin complejidad.
  - Quieren mejorar su control financiero sin sentir que están restringidos.

### **Perfil 2: Estudiantes Universitarios que Trabajan**

- **Qué:**
  - Manejan ingresos provenientes de su empleo y, en algunos casos, aún reciben apoyo de sus padres.
  - Gastan en transporte, renta, alimentación y estudio.
  - Intentan equilibrar sus ingresos con sus estudios, pero tienen gastos imprevistos.
- **Cómo:**
  - Llevan un registro mental o en notas de cuánto ganan y gastan.
  - A veces dividen su dinero en cuentas separadas o lo retiran en efectivo.
  - Intentan ahorrar, pero les cuesta debido a gastos inesperados.
- **Por Qué:**
  - Necesitan saber cuánto dinero pueden destinar al ahorro sin afectar sus gastos esenciales.
  - Quieren visualizar cómo su salario cubre sus necesidades y cuánto pueden gastar en ocio.
  - Buscan herramientas que los ayuden a planificar mejor sin que el proceso sea tedioso.

### **Perfil 3: Personas que Trabajan y Manejan Múltiples Cuentas**

- **Qué:**
  - Reciben ingresos de distintas fuentes (salarios, pagos por servicios, inversiones).
  - Usan varias cuentas bancarias y tarjetas de crédito.
  - Buscan administrar su dinero sin perder el control de los movimientos financieros.
- **Cómo:**
  - Revisan manualmente cada cuenta y tarjeta para saber cuánto tienen disponible.
  - Hacen transferencias frecuentes entre cuentas para equilibrar su flujo de dinero.

- Usan aplicaciones bancarias, pero sin integración entre ellas.
- **Por Qué:**
  - Necesitan consolidar toda su información financiera en un solo lugar.
  - Buscan evitar errores al calcular cuánto dinero tienen disponible.
  - Quieren herramientas que automatizan el seguimiento de gastos y pagos.

En el desarrollo de la aplicación web para la gestión de finanzas personales, se han identificado los siguientes perfiles de usuarios, quienes representan distintos grupos de personas con necesidades específicas en la administración de su dinero.

### **Perfiles Identificados**

#### **1. Estudiantes universitarios dependientes económicamente**

- Son jóvenes que aún no generan ingresos propios y dependen económicamente de sus padres o familiares.
- Reciben una cantidad fija de dinero en fechas específicas, por lo general mensual o semanalmente.
- Suelen gastar en comida, transporte, entretenimiento y material académico.
- Buscan administrar mejor su asignación para no quedarse sin dinero antes del siguiente depósito.

#### **2. Estudiantes universitarios que trabajan**

- Son personas que combinan sus estudios con un empleo parcial o de tiempo completo.
- Reciben ingresos propios y, en algunos casos, pueden seguir recibiendo apoyo económico de sus familias.
- Manejan gastos adicionales como renta, transporte, comida y estudios.
- Buscan equilibrar sus ingresos y gastos, estableciendo presupuestos y ahorros de manera más organizada.

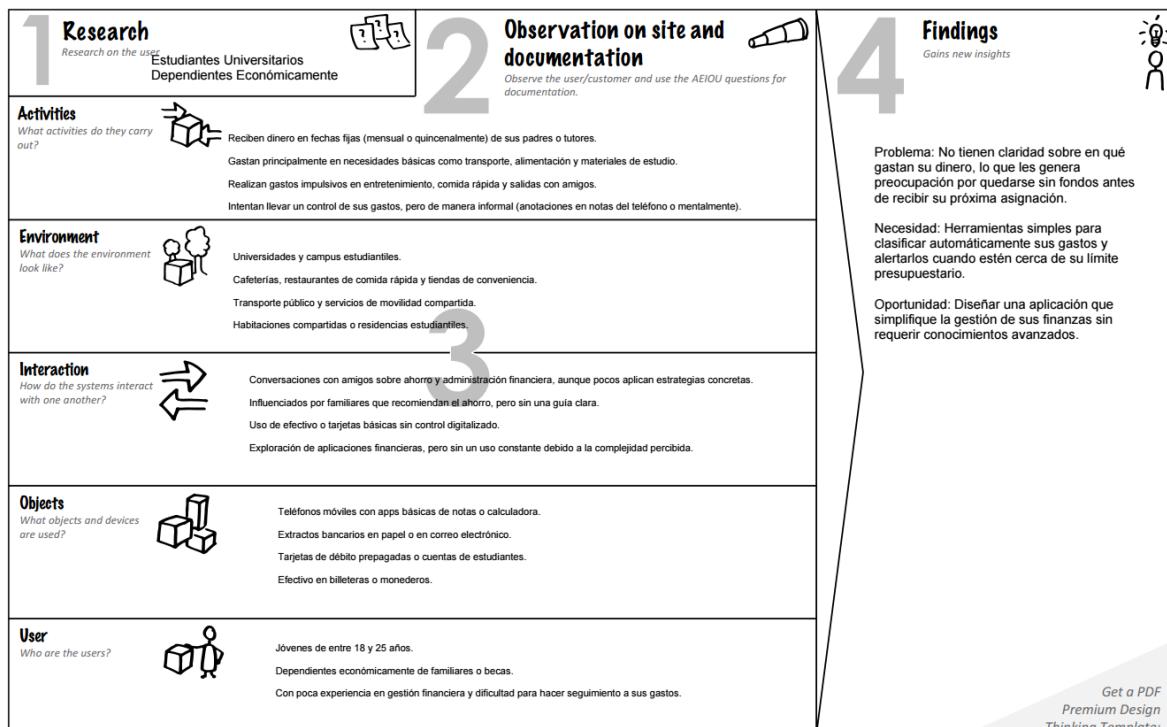
#### **3. Personas que trabajan y manejan múltiples cuentas o tarjetas**

- Son trabajadores que tienen independencia financiera y administran diferentes fuentes de ingreso.
- Pueden manejar varias cuentas bancarias, tarjetas de crédito y pagos recurrentes.
- Necesitan consolidar su información financiera en un solo lugar para facilitar la toma de decisiones.
- Buscan automatizar el control de sus transacciones, optimizar sus presupuestos y mejorar su planificación financiera.

Cada uno de estos perfiles tiene necesidades y comportamientos distintos en la gestión de sus finanzas, lo que permite diseñar funcionalidades que realmente resuelvan sus problemas.

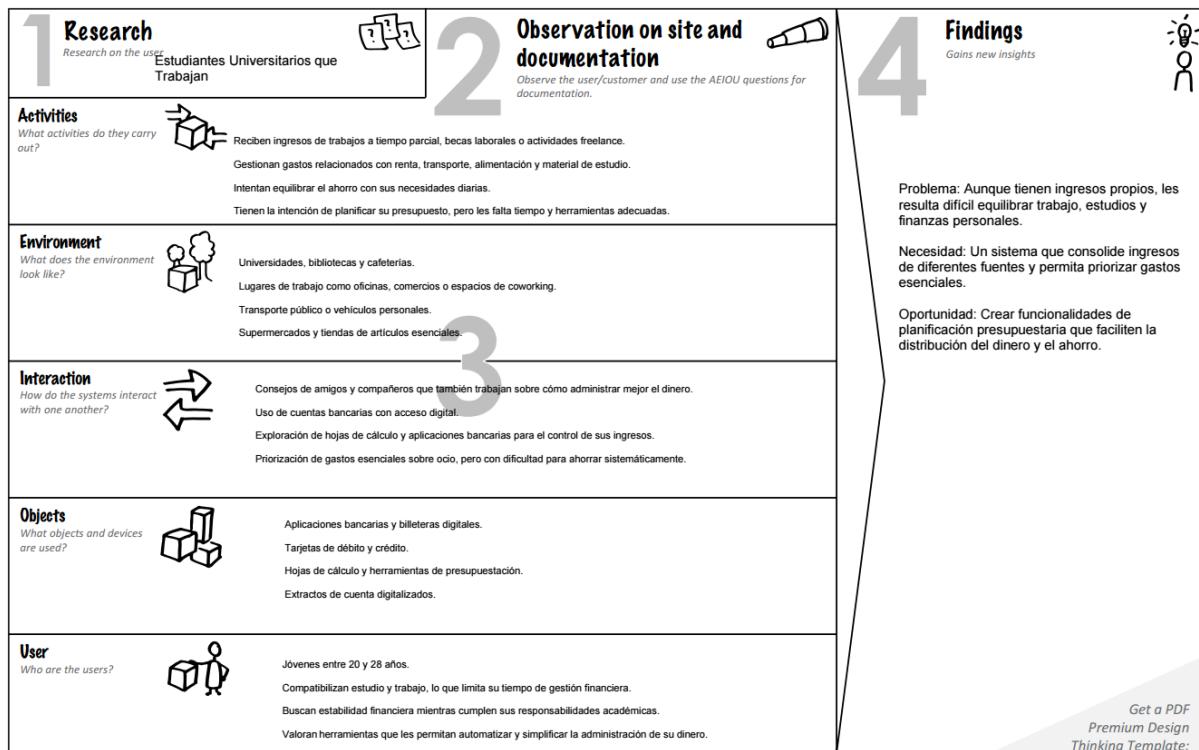
### **Framework AEIOU**

#### **1. Estudiantes universitarios dependientes económicamente**



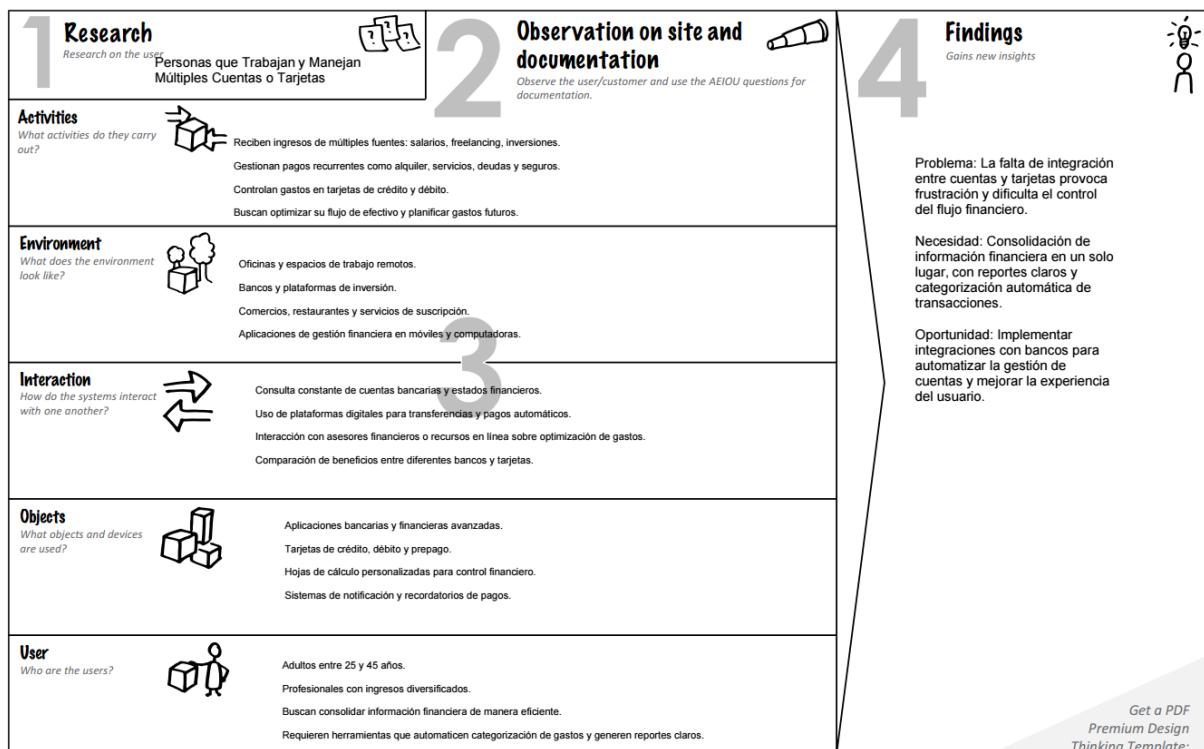
Get a PDF  
Premium Design  
Thinking Template:

## 2. Estudiantes universitarios que trabajan



Get a PDF  
Premium Design  
Thinking Template:

## 3. Personas que trabajan y manejan múltiples cuentas o tarjetas



Get a PDF  
Premium Design  
Thinking Template:

## Entrevistas

### Guion para las entrevistas

- ¿Qué es lo que más te preocupa cuando administras tus finanzas?
- ¿Qué herramientas o métodos utilizas actualmente para registrar tus ingresos y gastos?
- ¿Qué es lo más frustrante o difícil cuando intentas controlar tus finanzas personales?
- ¿Qué tipo de consejos financieros recibes de amigos, familia o compañeros?
- ¿Qué resultados esperas lograr al usar una herramienta de gestión financiera?
- ¿Qué comportamientos crees que deberías cambiar para mejorar tu situación financiera?
- ¿Cuáles son tus metas financieras principales a corto o mediano plazo?
- ¿Qué funcionalidades o características crees que serían imprescindibles en una aplicación para que te ayude con tus finanzas?

### Perfil 1: Estudiantes Universitarios Dependientes Económicamente

Los estudiantes dependientes mencionaron que su principal preocupación es quedarse sin dinero antes de que sus padres les den la siguiente asignación. Expresaron que a menudo no saben en qué gastan su dinero y, al final del mes, sienten que el dinero "desaparece". Aunque algunos intentan registrar sus gastos, lo hacen de manera muy básica, como en notas en su teléfono o mentalmente, lo que les dificulta tener un control real. Escuchan consejos de sus padres para ser más responsables y aprender a ahorrar, pero no tienen herramientas claras para hacerlo. En cuanto a lo que esperan de una solución, quieren algo simple que les permita visualizar cuánto gastan en diferentes categorías y cuánto les queda disponible para no quedarse sin dinero. Sus metas principales incluyen gestionar mejor su asignación mensual

para cubrir necesidades básicas y ahorrar para actividades sociales o imprevistos. Mencionaron que funcionalidades como la clasificación automática de gastos y alertas al acercarse al límite de su presupuesto serían de gran utilidad.

### **Perfil 2: Estudiantes Universitarios que Trabajan**

Los estudiantes que trabajan se sienten desafíados por la necesidad de equilibrar sus ingresos con los gastos relacionados con sus estudios y vida personal. Su principal preocupación es no tener suficiente dinero para cubrir todos sus compromisos, especialmente en meses con gastos inesperados. La mayoría utiliza hojas de cálculo básicas o simplemente consulta el saldo en sus aplicaciones bancarias, pero sienten que les falta un sistema unificado que les permita planificar mejor. Expresaron frustración porque, a pesar de trabajar, no siempre saben cómo distribuir su dinero para poder ahorrar y cubrir todas sus necesidades. Aunque reciben consejos de familiares o compañeros de trabajo, a menudo encuentran que no tienen tiempo para investigar sobre finanzas personales. Esperan una herramienta que les facilite organizar sus ingresos y gastos sin requerir demasiado tiempo de gestión. Sus metas incluyen ahorrar para emergencias y mantener un mejor control sobre sus finanzas mientras estudian. Las funcionalidades más deseadas son la consolidación de ingresos de diferentes fuentes y la posibilidad de crear presupuestos claros para estudios, renta y ocio.

### **Perfil 3: Personas que Trabajan y Manejan Múltiples Cuentas o Tarjetas**

Las personas que trabajan y manejan varias cuentas indicaron que su mayor preocupación es perder el control de sus finanzas al manejar múltiples fuentes de ingreso, cuentas bancarias y tarjetas. Se sienten frustradas por la falta de integración entre las diferentes plataformas que utilizan, ya que actualmente combinan aplicaciones bancarias, hojas de cálculo y notas manuales para rastrear sus gastos. Este proceso les resulta fragmentado y poco eficiente. Expresaron que también les preocupa olvidar pagos importantes o realizar transferencias tardías que puedan generar penalizaciones. Aunque escuchan recomendaciones de colegas o asesores financieros, consideran que las herramientas disponibles son muy complicadas o están diseñadas para empresas, no para personas individuales con múltiples cuentas. Esperan una solución que consolide toda su información financiera en una sola plataforma, les permita ver su saldo total y clasificar automáticamente sus transacciones. Sus metas incluyen optimizar su flujo de efectivo y ahorrar para metas específicas como la compra de bienes importantes o inversiones futuras. Las funcionalidades que más valoran son la integración con bancos, la categorización automática de gastos y la generación de reportes detallados por cuenta.

## Insights Detectados

### 4. Estudiantes universitarios dependientes económicamente:

- **Insight clave:** Muchos estudiantes no tienen claridad sobre en qué gastan su dinero. Les preocupa quedarse sin fondos antes de recibir su próxima asignación.
- **Necesidad:** Herramientas simples que clasifiquen automáticamente sus gastos y les avisen cuando estén cerca de su límite presupuestario.
- **Oportunidad:** Diseñar una aplicación que simplifique la gestión de sus finanzas, adaptándose a su falta de experiencia financiera.

### 5. Estudiantes universitarios que trabajan:

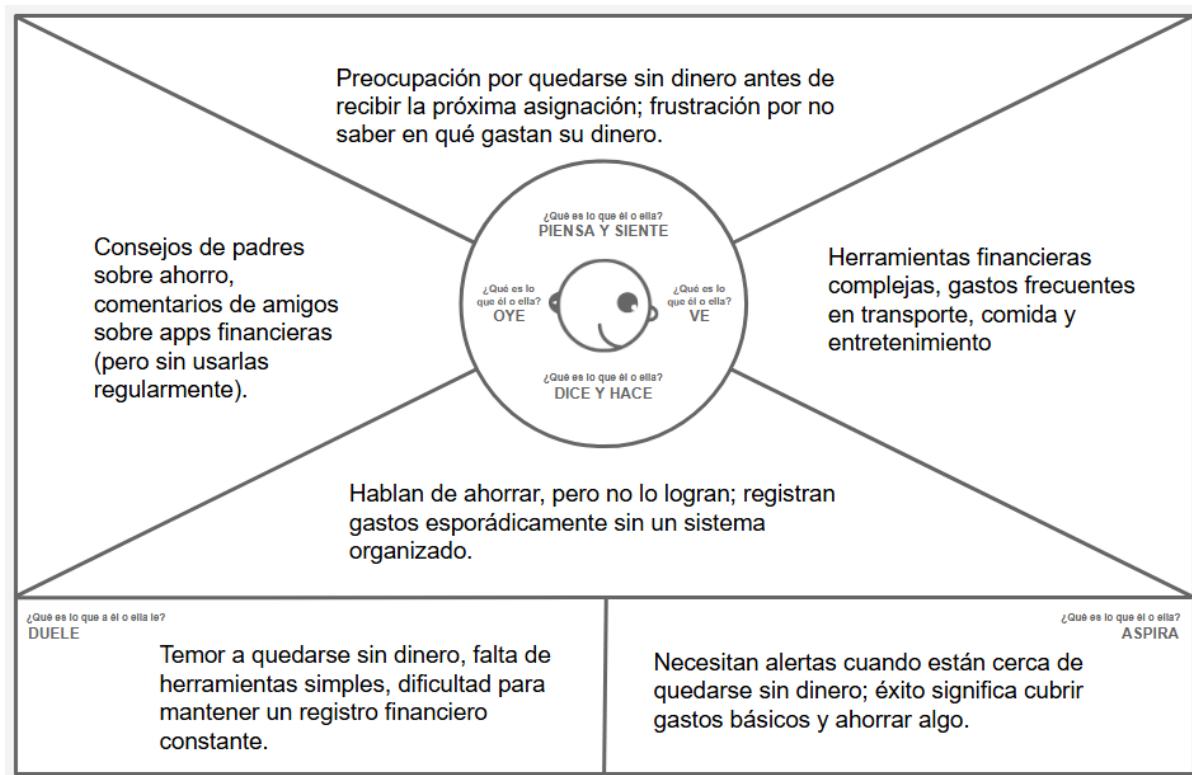
- **Insight clave:** A pesar de tener ingresos propios, enfrentan dificultades para equilibrar trabajo, estudios y finanzas personales.
- **Necesidad:** Un sistema que consolide sus ingresos de diferentes fuentes y les permita priorizar gastos esenciales.
- **Oportunidad:** Crear funcionalidades de planificación presupuestaria que ayuden a estos usuarios a distribuir mejor su dinero y ahorrar.

### 6. Personas que trabajan y manejan múltiples cuentas o tarjetas:

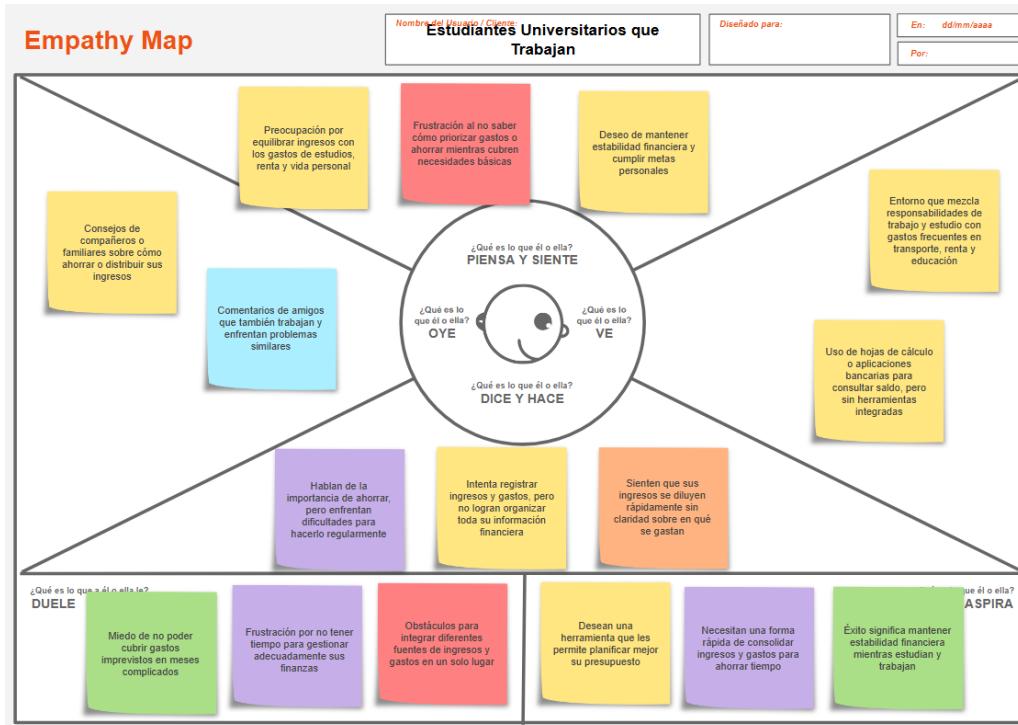
- **Insight clave:** La falta de integración entre cuentas y tarjetas provoca frustración y dificulta el control del flujo financiero.
- **Necesidad:** Consolidación de información financiera en un solo lugar, con reportes claros y categorización automática de transacciones.
- **Oportunidad:** Implementar integraciones con bancos para automatizar la gestión de cuentas y mejorar la experiencia del usuario.

## Mapas de Empatía

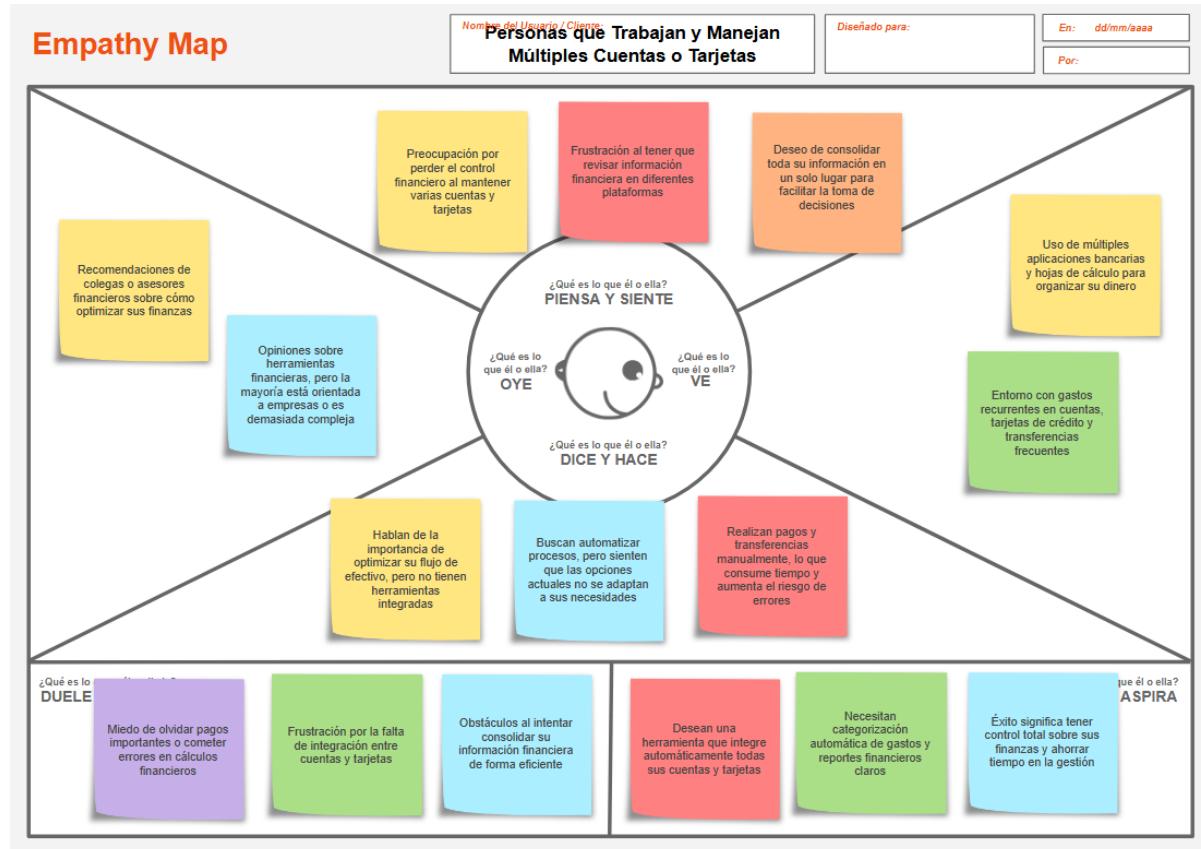
### Estudiantes Universitarios Dependientes Económicamente



### Estudiantes Universitarios que Trabajan



## Personas que Trabajan y Manejan Múltiples Cuentas o Tarjetas



### Oportunidades Descubiertas con la Técnica "¿Cómo podríamos?"

#### 1. Estudiantes dependientes económicamente:

- ¿Cómo podríamos ayudar a los estudiantes a tener claridad sobre en qué gastan su dinero?
- ¿Cómo podríamos alertarlos antes de que se queden sin fondos?
- ¿Cómo podríamos diseñar una herramienta fácil de usar que no requiera conocimientos financieros?

#### 2. Estudiantes que trabajan:

- ¿Cómo podríamos facilitar la planificación financiera de quienes tienen ingresos limitados y múltiples prioridades?
- ¿Cómo podríamos ayudarles a ahorrar sin que sientan que están sacrificando demasiado?
- ¿Cómo podríamos consolidar sus ingresos y gastos para darles una visión clara de su presupuesto?

#### 3. Trabajadores con múltiples cuentas o tarjetas:

- ¿Cómo podríamos integrar múltiples cuentas y tarjetas en una sola plataforma de manera eficiente?
- ¿Cómo podríamos categorizar automáticamente las transacciones para ahorrar tiempo?
- ¿Cómo podríamos reducir la frustración de revisar información en varias plataformas?

## Definición del Proyecto

### Problema que Resolverá el Proyecto:

La falta de herramientas integradas, intuitivas y adaptadas para diferentes tipos de usuarios dificulta la gestión financiera personal. Esto afecta a estudiantes y trabajadores que necesitan una manera eficiente de registrar, analizar y planificar sus ingresos y gastos, generando frustración y pérdida de control financiero.

### Enunciado del Problema:

¿Cómo podemos desarrollar una aplicación web que permita a los usuarios, independientemente de su nivel de independencia financiera, gestionar eficientemente sus ingresos y gastos, consolidar múltiples fuentes de información y obtener reportes claros que les ayuden a tomar mejores decisiones económicas?

### Objetivo del Proyecto:

Diseñar e implementar una aplicación web intuitiva que facilite la gestión financiera personal mediante funcionalidades como consolidación de cuentas, clasificación automática de gastos y generación de reportes financieros, adaptada a las necesidades específicas de estudiantes dependientes, estudiantes que trabajan y personas que manejan múltiples cuentas o tarjetas.

## CORTE #2

Design Thinking y Aplicación de Design Studio

### 1. Proceso de Design Studio

#### *Fase 1: Generación de Ideas*

En esta fase, realizamos una sesión de lluvia de ideas para identificar soluciones innovadoras a los problemas detectados en la gestión de finanzas personales. Utilizamos la técnica "¿Cómo podríamos...?" para definir preguntas clave basadas en los insights obtenidos de entrevistas y observaciones.

#### Preguntas clave:

*¿Cómo podríamos ayudar a los usuarios a visualizar sus gastos de manera intuitiva?*

*¿Cómo podríamos facilitar la creación de presupuestos personalizados?*

*¿Cómo podríamos mejorar la integración con cuentas bancarias?*

*¿Cómo podríamos alertar a los usuarios cuando se acercan a su límite de gasto?*

Cada integrante del equipo propuso soluciones y creó bocetos de interfaces y funcionalidades. Se recopilaron todas las ideas en notas adhesivas y se votaron las más viables.

## Fase 2: Selección de Ideas y Sketches

Después de analizar y discutir las ideas generadas, seleccionamos las más repetidas y votadas:

- **Dashboard financiero visual:** Un panel que muestre ingresos, gastos y presupuestos con gráficos intuitivos.
- **Clasificación automática de gastos:** Utilizar etiquetas inteligentes para categorizar transacciones sin intervención manual.
- **Alertas de presupuesto:** Notificaciones cuando el usuario se acerque al límite de un presupuesto establecido.
- **Integración con APIs bancarias:** Posibilidad de importar movimientos automáticamente.

Realizamos bocetos rápidos de cada una de estas ideas para visualizar su implementación.

## Fase 3: Creación de Prototipos

A partir de los bocetos, elaboramos prototipos en papel con el flujo de interacción del usuario. Los prototipos iniciales incluyeron:

- Pantalla principal con resumen financiero.
- Sección de presupuestos para establecer metas de ahorro y control de gastos.
- Gráficos interactivos para el análisis de gastos.
- Notificaciones de advertencia sobre gastos excesivos.

Estos prototipos se refinaron a través de tres iteraciones con base en la retroalimentación obtenida de usuarios potenciales.

## Fase 4: Testeo de Prototipos

Para validar nuestras soluciones, realizamos consultas con usuarios que representaban nuestros perfiles objetivo. Registramos sus comentarios sobre la usabilidad y funcionalidades, lo que nos permitió ajustar el diseño.

### Principales aprendizajes del testeo:

- La pantalla de resumen financiero debe incluir un saldo proyectado.
- Las categorías de gasto deben ser editables y personalizables.
- Las notificaciones deben ser configurables para evitar alertas molestas.

Con base a esto se irá realizando los ajustes y refinando los prototipos finales para el futuro.

## 2. Evidencias Gráficas

[BrainStorming - Grupo #9](#)

## Modelación del Sistema

### 1. Lista de Historias de Usuario

Para definir el comportamiento del sistema, identificamos las siguientes historias de usuario basadas en las necesidades detectadas:

- **Registro de usuario:** Como usuario, quiero registrarme con mi correo electrónico y una contraseña para acceder a la aplicación.
- Inicio de sesión: Como usuario, quiero iniciar sesión de forma segura para acceder a mis datos financieros.
- **Registro de transacciones:** Como usuario, quiero ingresar mis ingresos y gastos manualmente para llevar un control de mis finanzas.
- **Clasificación de transacciones:** Como usuario, quiero que mis transacciones se clasifiquen automáticamente en categorías para facilitar el análisis de mis gastos.
- **Visualización de informes:** Como usuario, quiero ver gráficos interactivos que me muestren patrones de consumo.
- **Definición de presupuestos:** Como usuario, quiero establecer límites de gasto para distintas categorías y recibir alertas cuando me acerque a esos límites.
- **Integración con bancos:** Como usuario, quiero importar mis transacciones automáticamente desde mi cuenta bancaria.
- **Exportación de reportes:** Como usuario, quiero exportar mis reportes financieros en formato CSV o PDF.

### 2. Descripción de Actores

Los siguientes actores son los que se han ido identificando y con los que se ha ido trabajando dentro del sistema:

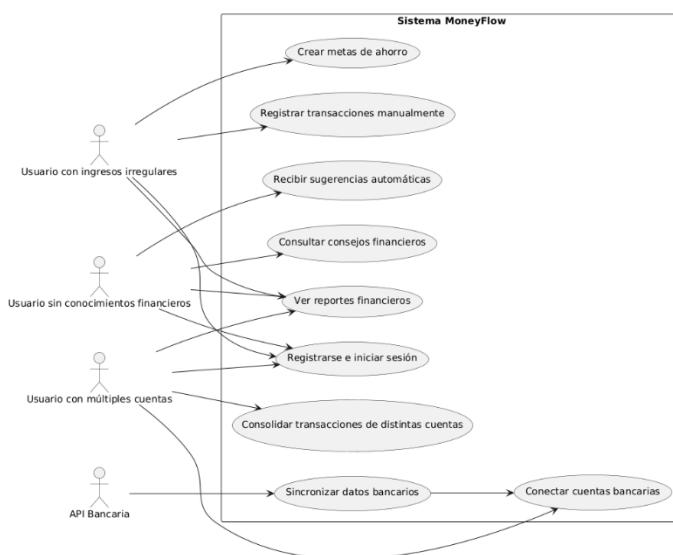
- **Usuario con ingresos irregulares:** Persona que tiene trabajos informales o autónomos y necesita gestionar su flujo de ingresos variable.
- **Usuario con múltiples cuentas bancarias:** Persona que maneja diversas cuentas y tarjetas y requiere consolidar su información financiera en una sola plataforma.
- **Usuario sin conocimientos financieros:** Persona que busca una interfaz sencilla y educativa para mejorar su comprensión sobre sus finanzas personales.

### 3. Mapa de Historias de Usuario

Etapa	Usuario con ingresos irregulares	Usuario con múltiples cuentas	Usuario sin conocimientos financieros
Registro e inicio	Registrarse con email o Google	Registrarse con email o Google	Registrarse con email o Google
	Iniciar sesión con credenciales	Iniciar sesión con credenciales	Iniciar sesión con credenciales
Gestión de transacciones	Registrar ingresos y gastos manualmente	Conectar múltiples cuentas bancarias	Recibir sugerencias de categorías de gasto

	Clasificar transacciones en categorías	Consolidar transacciones de distintas cuentas	Ingreso automático de datos con preguntas guiadas
<b>Análisis y reportes</b>	Ver balance mensual y flujo de caja	Generar reportes financieros	Dashboard educativo sobre hábitos financieros
<b>Extras</b>	Establecer metas de ahorro	Sincronizar datos automáticamente	Consejos financieros personalizados

#### 4. Diagrama UML de Casos de Uso



#### 5. Priorización de Historias de Usuario

Historia de Usuario	Usuario	Prioridad
<b>Registrarse e iniciar sesión</b>	Todos	Alta
<b>Registrar transacciones manualmente</b>	Usuario con ingresos irregulares	Alta
<b>Conectar cuentas bancarias</b>	Usuario con múltiples cuentas	Alta
<b>Ver reportes financieros</b>	Todos	Media
<b>Establecer metas de ahorro</b>	Usuario con ingresos irregulares	Media
<b>Recibir sugerencias de categorías</b>	Usuario sin conocimientos	Media
<b>Sincronizar datos bancarios</b>	API Bancaria	Baja
<b>Consejos financieros personalizados</b>	Usuario sin conocimientos	Baja

#### Requisitos No Funcionales

##### 1. Apariencia o Interfaz Externa

- Diseño minimalista y moderno con colores neutros y accesibles.

- Uso de iconografía clara para mejorar la navegación.

## 2. Usabilidad

- Interfaces intuitivas con accesibilidad para usuarios sin experiencia financiera.
- Implementación de guías interactivas para usuarios nuevos.

## 3. Rendimiento

- El tiempo de carga de la aplicación no debe superar los 2 segundos.
- Capacidad de procesar hasta 10,000 transacciones simultáneas.

## 4. Soporte

- Actualizaciones periódicas para garantizar compatibilidad con nuevas versiones de sistemas operativos.
- Servicio de asistencia en línea disponible 24/7.

## 5. Portabilidad

- Compatible con dispositivos Android e iOS.
- Accesible a través de navegadores web modernos.

## 6. Seguridad y Privacidad

- Implementación de autenticación segura.
- Cifrado de datos sensibles como contraseñas y datos de las cuentas.
- Protección contra ataques de fuerza bruta.

## 7. Políticos y Culturales

- Cumplimiento de normativas locales sobre privacidad de datos financieros.
- Traducción y adaptación cultural para distintos mercados.

## 8. Legales

- Cumplimiento con GDPR y otras regulaciones de protección de datos.
- Contratos de términos y condiciones claros para los usuarios.

## 9. Confiabilidad

- Garantizar un tiempo de actividad del 99%.
- Implementación de copias de seguridad automáticas cada 24 horas.

## 10. Interfaz Interna

- API bien documentada para futuras integraciones.
- Modularidad en el desarrollo para facilitar mantenimiento.

## 11. Ayudas y Documentación en Línea

- Base de conocimientos accesible desde la aplicación.
- Videos tutoriales y documentación técnica en el sitio web.

## 12. Hardware

- Optimización para dispositivos con al menos 2GB de RAM y procesador quad-core.

## 13. Software

- Requiere Android 9+ y iOS 13+ para su instalación.
- Compatible con navegadores Chrome, Firefox y Edge.

## 14. Restricciones en el Diseño y la Implementación

- Uso de frameworks de código abierto siempre que sea posible.
- Estandarización de componentes UI para garantizar consistencia.

**Informe de Gestión****1. Bitácora de Interacción con el Product Owner**

<b>Fecha</b>	<b>Participantes</b>	<b>Temas Tratados</b>	<b>Observaciones</b>
<b>07/02/2025</b>	Equipo de desarrollo, Product Owner	Revisión inicial de necesidades y funcionalidades esperadas.	Se solicitó mayor énfasis en la categorización automática de gastos.
<b>14/02/2025</b>	Equipo de desarrollo, Product Owner	Presentación de lluvia de ideas y recolección de feedback.	Se sugirió las ideas a tomar de la lluvia de ideas realizada.
<b>21/02/2025</b>	Equipo de desarrollo, Product Owner	Implementación de ajustes y revisión.	Se validó la integración con APIs bancarias y ajustes en las notificaciones.

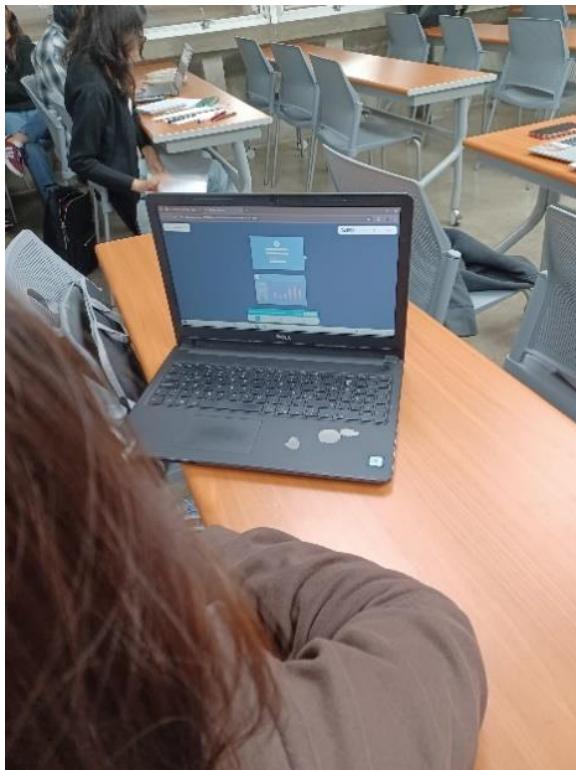
## TERCER CORTE DEL PROYECTO

### DISIGN THINKING

Prototipos:

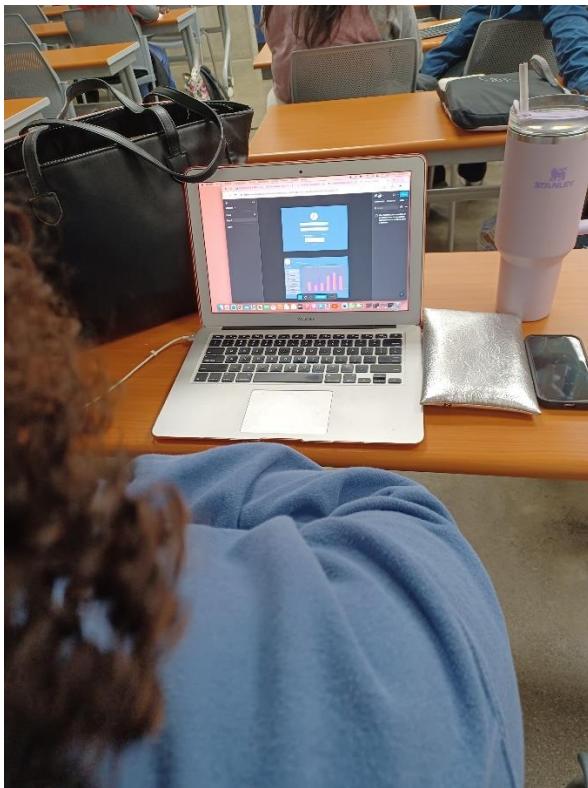
<https://www.figma.com/design/ZfjZ4YXlenkeUE9rli6IZz/Untitled?node-id=0-1&t=B86PCkBmxSZhgGFh-1>

Pruebas:



### Comentarios:

1. "El diseño se ve muy simple, falta un ícono o algo visualmente atractivo en la pantalla de inicio de sesión."  
Mejora aplicada: Se agregó un ícono de usuario en la parte superior para mejorar la identidad visual.
2. "Sería más conveniente poder registrarse rápidamente con una cuenta de Google."  
Mejora aplicada: Se añadió el botón *Sign up with Google* para un registro más rápido.
3. "El campo de usuario y contraseña no tienen bordes visibles o resaltados que ayuden a diferenciarlos del fondo."  
Mejora aplicada: Se mejoró el contraste y el tamaño de los campos de texto para mayor visibilidad.
4. "El botón de 'Registrarse' no se ve claramente, parece más un enlace que un botón importante."  
Mejora aplicada: Se hizo un diseño más llamativo para la opción de registro.



Comentarios:

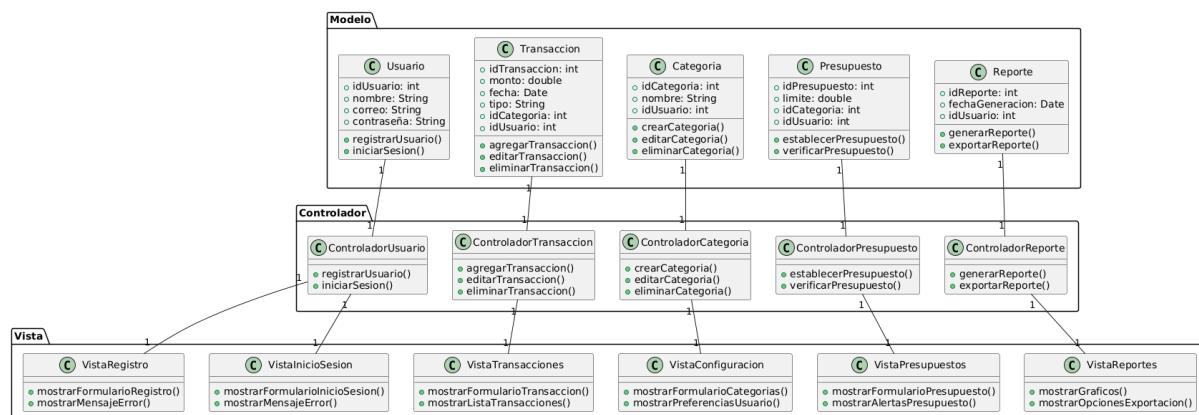
1. "El sistema de navegación no es intuitivo, cuesta encontrar las opciones para ver diferentes secciones."  
Mejora aplicada: Se incorporó un menú lateral con botones claros y etiquetas descriptivas.
2. "Los gráficos se ven bien, pero deberían tener un fondo diferenciado para destacar más."  
Mejora aplicada: Se añadió un fondo diferenciado detrás del gráfico para mejorar la visibilidad.
3. "No hay suficiente información sobre qué representan los datos en el gráfico."  
Mejora aplicada: Se agregaron etiquetas y nombres a los ejes del gráfico.
4. "Las opciones de navegación deberían estar más organizadas y agrupar funciones similares."  
Mejora aplicada: Se ordenaron las opciones en categorías como *Financial, Product, Transfer*, etc

## ANÁLISIS

### Requisitos Funcionales del Proyecto

R. F.	Requisito Funcional	Historia de Usuario
RF1	Registro con correo electrónico y contraseña.	Registrarse con email o Google.
RF2	Registro mediante autenticación de terceros (Google).	Registrarse con email o Google.
RF3	Inicio de sesión con credenciales.	Iniciar sesión con credenciales.
RF4	Inicio de sesión mediante autenticación de terceros (Google).	Iniciar sesión con credenciales.
RF5	Ingreso manual de ingresos y gastos.	Registrar transacciones manualmente.
RF6	Edición y eliminación de transacciones.	Registrar transacciones manualmente.
RF7	Clasificación automática de transacciones en categorías predefinidas.	Clasificar transacciones en categorías.
RF8	Creación y personalización de categorías de gastos.	Clasificar transacciones en categorías.
RF9	Generación de gráficos interactivos de patrones de consumo.	Ver gráficos interactivos de patrones de consumo.
RF10	Resumen financiero mensual (ingresos, gastos, saldo disponible).	Ver gráficos interactivos de patrones de consumo.
RF11	Establecimiento de límites de gasto por categoría.	Establecer límites de gasto y recibir alertas.
RF12	Notificaciones al acercarse o superar límites de gasto.	Establecer límites de gasto y recibir alertas.
RF15	Exportación de reportes en formato CSV.	Exportar reportes financieros en formato CSV o PDF.
RF16	Exportación de reportes en formato PDF.	Exportar reportes financieros en formato CSV o PDF.
RF17	Sugerencias de categorías de gasto basadas en transacciones anteriores.	Recibir sugerencias de categorías de gasto.
RF18	Consejos financieros personalizados basados en hábitos de gasto.	Recibir consejos financieros personalizados.

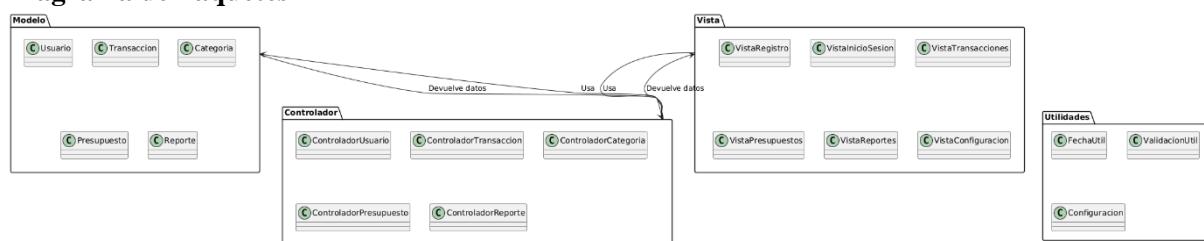
## Modelo Vista – Controlador



### Explicación del modelo

Capa	Clase	Descripción
<b>Modelo</b>	Usuario	Gestiona el registro e inicio de sesión de usuarios.
	Transaccion	Representa las transacciones financieras (ingresos y gastos).
	Categoría	Representa las categorías de gastos (por ejemplo, comida, transporte).
	Presupuesto	Representa los límites de gasto establecidos por el usuario.
	Reporte	Representa los reportes financieros generados por el sistema.
	Banco	Gestiona la integración con APIs bancarias para importar transacciones.
<b>Vista</b>	VistaRegistro	Interfaz para el registro de usuarios.
	VistaInicioSesion	Interfaz para el inicio de sesión.
	VistaTransacciones	Interfaz para registrar y visualizar transacciones.
	VistaPresupuestos	Interfaz para establecer y visualizar presupuestos.
	VistaReportes	Interfaz para visualizar gráficos y reportes financieros.
	VistaConfiguracion	Interfaz para configurar categorías y preferencias del usuario.
<b>Controlador</b>	ControladorUsuario	Gestiona el registro e inicio de sesión de usuarios.
	ControladorTransaccion	Gestiona las transacciones financieras.
	ControladorCategoria	Gestiona las categorías de gastos.
	ControladorPresupuesto	Gestiona los presupuestos de los usuarios.
	ControladorReporte	Gestiona la generación de reportes financieros.
	ControladorBanco	Gestiona la integración con APIs bancarias.

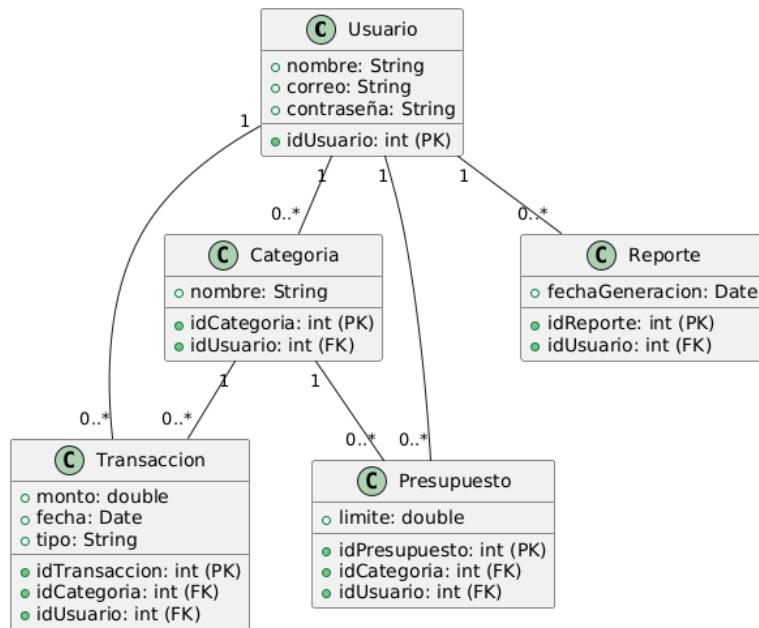
## Diagrama de Paquetes



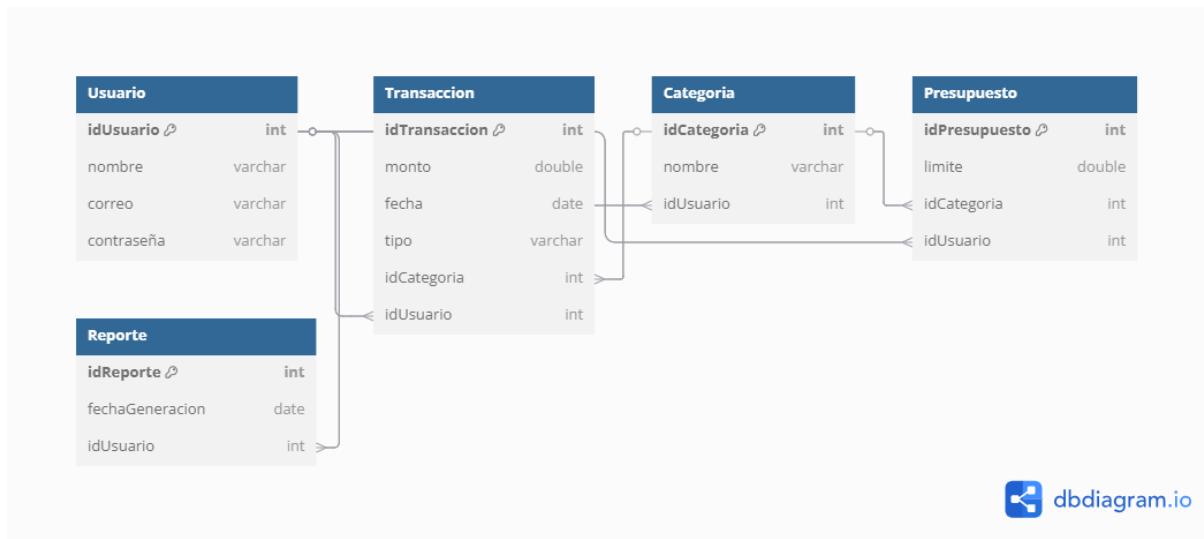
## Descripción de las Clases

Paquete	¿Qué contiene?	Ejemplos de Clases
<b>Modelo</b>	Las clases que guardan y manejan los datos de la aplicación.	Usuario, Transaccion, Categoria, Presupuesto, Reporte, Banco.
<b>Vista</b>	Las clases que muestran la interfaz de usuario (lo que el usuario ve).	VistaRegistro, VistaInicioSesion, VistaTransacciones, VistaPresupuestos, etc.
<b>Controlador</b>	Las clases que gestionan la lógica de la aplicación (el "intermediario").	ControladorUsuario, ControladorTransaccion, ControladorCategoria, etc.
<b>Utilidades</b>	Las clases que ayudan en tareas generales (opcional).	FechaUtil, ValidacionUtil, Configuracion.

### Diagrama de Clases Persistentes



### Diagrama Entidad Relación



## Propuesta 1: React + Node.js + Express + PostgreSQL + JWT

### 1. Selección de Tecnologías

#### Frontend: React

- **Justificación:**
  - React fue seleccionado debido a su eficiencia en la construcción de interfaces de usuario dinámicas y responsivas. Su capacidad para crear aplicaciones de una sola página (SPA) permite una experiencia de usuario fluida, lo cual es esencial para una aplicación de gestión financiera donde la interactividad y el rendimiento son críticos.
- **Ventajas:**
  - Alto rendimiento y eficiencia en la renderización de componentes.
  - Amplia comunidad y disponibilidad de librerías complementarias (React Router, Axios, Chart.js).
  - Facilidad de aprendizaje para desarrolladores con experiencia en JavaScript.
- **Desventajas:**
  - Requiere configuración adicional para la gestión de estado en aplicaciones grandes.
  - Puede presentar una curva de aprendizaje para desarrolladores sin experiencia previa en JavaScript.

#### Backend: Node.js + Express

- **Justificación:**
  - Node.js, junto con el framework Express, fue seleccionado por su capacidad para manejar un alto número de conexiones simultáneas gracias a su modelo de E/S no bloqueante. Esto lo hace ideal para aplicaciones en tiempo real, como una plataforma de gestión financiera que requiere respuestas rápidas y escalabilidad.
- **Ventajas:**
  - Alto rendimiento y escalabilidad, especialmente en aplicaciones con un gran número de usuarios concurrentes.
  - Uso de JavaScript tanto en el frontend como en el backend, lo que simplifica el desarrollo y mantenimiento.
  - Amplia comunidad y disponibilidad de módulos en npm.
- **Desventajas:**
  - Menos robusto que otros frameworks (como Django) para aplicaciones con lógica de negocio compleja.
  - Requiere configuración manual para funcionalidades avanzadas, como autenticación y validación.

#### Base de Datos: PostgreSQL

- **Justificación:**
  - PostgreSQL fue seleccionado por su robustez y capacidad para manejar transacciones complejas y relaciones entre datos. Es una base de datos relacional altamente escalable, lo que la hace adecuada para una aplicación que gestiona información financiera sensible y requiere integridad de datos.
- **Ventajas:**

- Soporte avanzado para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
- Escalabilidad y confiabilidad, con capacidad para manejar grandes volúmenes de datos.
- Compatibilidad con ORMs como Sequelize, que facilitan la interacción con la base de datos.
- **Desventajas:**
  - Mayor complejidad en la configuración y administración en comparación con bases de datos más ligeras (como SQLite).
  - Requiere más recursos de hardware para operar de manera óptima.

### **Autenticación: JWT (JSON Web Tokens)**

- **Justificación:**
  - JWT fue seleccionado por su simplicidad y eficiencia en la autenticación de usuarios. Al ser un estándar ampliamente adoptado, permite implementar un sistema de autenticación seguro sin necesidad de almacenar sesiones en el servidor.
- **Ventajas:**
  - Ligero y fácil de implementar.
  - No requiere almacenamiento de sesiones en el servidor, lo que reduce la carga en el backend.
  - Fácil integración con Node.js y React.
- **Desventajas:**
  - Los tokens JWT pueden ser inseguros si no se implementan correctamente (por ejemplo, si no se invalidan adecuadamente).
  - No es ideal para aplicaciones que requieren autenticación con terceros (como Google o Facebook).

## **2. Requisitos No Funcionales**

### **Cantidad de Usuarios Probables**

- **Estimación:** El sistema está diseñado para manejar entre **1,000 y 10,000 usuarios activos simultáneamente**.
- **Justificación:** Node.js, con su modelo de E/S no bloqueante, es altamente eficiente en el manejo de un gran número de conexiones simultáneas, lo que garantiza un buen rendimiento incluso con un alto volumen de usuarios.

### **Tiempo de Respuesta**

- **Estimación:** El tiempo promedio de respuesta del sistema será de **menos de 2 segundos**.
- **Justificación:** Node.js y Express son conocidos por su bajo tiempo de latencia y alto rendimiento, lo que asegura respuestas rápidas a las solicitudes de los usuarios.

### **Escalabilidad**

- **Estimación:** El sistema está diseñado para escalar horizontalmente, permitiendo manejar un crecimiento en el número de usuarios y transacciones.
- **Justificación:** PostgreSQL es altamente escalable y puede manejar grandes volúmenes de datos. Además, Node.js permite escalar horizontalmente añadiendo más instancias del servidor.

## Disponibilidad y Confiabilidad

- **Estimación:** El sistema tendrá un tiempo de actividad del **99.9%**.
- **Justificación:** PostgreSQL ofrece replicación y copias de seguridad automáticas, lo que garantiza alta disponibilidad y confiabilidad en la gestión de datos.

## Tamaño en Disco

- **Estimación:** Se espera que la información almacenada ocupe entre **10 GB y 100 GB** en los primeros años.
- **Justificación:** PostgreSQL es eficiente en el manejo de grandes volúmenes de datos y permite optimizar el almacenamiento mediante técnicas de particionamiento y compresión.

## Propuesta 2: React + Django + OAuth + MySQL

### 1. Selección de Tecnologías

#### Frontend: React

- **Justificación:**
  - React fue seleccionado por su capacidad para construir interfaces de usuario dinámicas y responsivas. Su enfoque en componentes reutilizables y su eficiencia en la renderización lo hacen ideal para una aplicación que requiere una experiencia de usuario fluida.
- **Ventajas:**
  - Alto rendimiento y eficiencia en la renderización de componentes.
  - Amplia comunidad y disponibilidad de librerías complementarias (React Router, Axios, Chart.js).
  - Facilidad de aprendizaje para desarrolladores con experiencia en JavaScript.
- **Desventajas:**
  - Requiere configuración adicional para la gestión de estado en aplicaciones grandes.
  - Puede presentar una curva de aprendizaje para desarrolladores sin experiencia previa en JavaScript.

#### Backend: Django

- **Justificación:**
  - Django fue seleccionado por ser un framework robusto y completo que incluye muchas funcionalidades listas para usar, como un ORM integrado, autenticación y un panel de administración. Esto lo hace ideal para aplicaciones que requieren un alto nivel de seguridad y escalabilidad.
- **Ventajas:**
  - Alto nivel de seguridad y escalabilidad, ideal para aplicaciones medianas y grandes.
  - ORM integrado que facilita la interacción con la base de datos.
  - Amplia comunidad y disponibilidad de recursos y documentación.
- **Desventajas:**
  - Mayor complejidad en comparación con frameworks más ligeros (como Express).
  - Requiere más configuración inicial y recursos de hardware.

#### Base de Datos: MySQL

- **Justificación:**
  - MySQL fue seleccionado por ser una base de datos relacional popular y fácil de usar, con un amplio soporte comunitario. Es compatible con Django a través de su ORM, lo que facilita la gestión de datos.

- **Ventajas:**
  - Fácil de usar y administrar, con una curva de aprendizaje suave.
  - Ampliamente utilizado y con gran soporte comunitario.
  - Escalable y confiable para aplicaciones medianas y grandes.
- **Desventajas:**
  - Menos funcionalidades avanzadas en comparación con PostgreSQL (por ejemplo, soporte limitado para tipos de datos complejos).
  - Puede tener limitaciones en el manejo de grandes volúmenes de datos en comparación con bases de datos más robustas.

#### **Autenticación: OAuth**

- **Justificación:**
  - OAuth fue seleccionado por ser un estándar ampliamente utilizado para la autenticación con terceros (como Google o Facebook). Esto permite a los usuarios registrarse e iniciar sesión utilizando sus cuentas existentes, lo que mejora la experiencia de usuario.
- **Ventajas:**
  - Permite autenticación con terceros de manera sencilla y segura.
  - Altamente confiable y ampliamente adoptado.
  - Fácil integración con Django mediante librerías como django-allauth.
- **Desventajas:**
  - Mayor complejidad en la implementación en comparación con JWT.
  - Requiere configuración adicional para manejar tokens de acceso y permisos.

## **2. Requisitos No Funcionales**

### **Cantidad de Usuarios Probables**

- **Estimación:** El sistema está diseñado para manejar entre **10,000 y 100,000 usuarios activos simultáneamente**.
- **Justificación:** Django, con su arquitectura robusta y escalable, es capaz de manejar un gran número de conexiones simultáneas, lo que lo hace ideal para aplicaciones con un alto volumen de usuarios.

### **Tiempo de Respuesta**

- **Estimación:** El tiempo promedio de respuesta del sistema será de **menos de 2 segundos**.
- **Justificación:** Django, combinado con un servidor eficiente como **Gunicorn**, ofrece un alto rendimiento y baja latencia, asegurando respuestas rápidas a las solicitudes de los usuarios.

### **Escalabilidad**

- **Estimación:** El sistema está diseñado para escalar horizontalmente, permitiendo manejar un crecimiento en el número de usuarios y transacciones.
- **Justificación:** MySQL es altamente escalable y puede manejar grandes volúmenes de datos. Además, Django permite escalar horizontalmente añadiendo más instancias del servidor.

### **Disponibilidad y Confiabilidad**

- **Estimación:** El sistema tendrá un tiempo de actividad del **99.9%**.
- **Justificación:** MySQL ofrece replicación y copias de seguridad automáticas, lo que garantiza alta disponibilidad y confiabilidad en la gestión de datos.

### **Tamaño en Disco**

- **Estimación:** Se espera que la información almacenada ocupe entre **10 GB y 100 GB** en los primeros años.

- **Justificación:** MySQL es eficiente en el manejo de grandes volúmenes de datos y permite optimizar el almacenamiento mediante técnicas de particionamiento y compresión.

### Gestión de Tiempo

Formulario LOGT – Milton Polanco 23471

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
27/02/2025	09:00	12:00	0:15	2:45	Prototipos	Revisión de requerimientos y diseño inicial de prototipos.
28/02/2025	10:00	13:00	0:30	2:30	Prototipos	Creación de dos versiones de prototipos y recopilación de feedback.
01/03/2025	14:00	17:00	0:20	2:40	Prototipos	Refinamiento de prototipos basado en feedback de usuarios.
04/03/2025	09:00	11:00	0:10	1:50	Selección de Tecnologías	Investigación de tecnologías para el frontend y autenticación.
06/03/2025	10:00	12:00	0:15	1:45	Selección de Tecnologías	Discusión en equipo sobre la selección de tecnologías.
08/03/2025	14:00	16:00	0:20	1:40	Gestión del Tiempo	Revisión y ajuste del cronograma del proyecto.
10/03/2025	09:00	11:00	0:10	1:50	Prototipos	Finalización de prototipos y preparación de evidencias (fotos/videos).
12/03/2025	10:00	13:00	0:30	2:30	Selección de Tecnologías	Justificación final de la selección de tecnologías.
14/03/2025	14:00	16:00	0:15	1:45	Gestión del Tiempo	Elaboración del informe final de gestión del tiempo.

**Formulario LOGT – Gadiel Ocaña 231270**

<b>Fecha</b>	<b>Inicio</b>	<b>Fin</b>	<b>Tiempo Interrupción</b>	<b>Delta Tiempo</b>	<b>Fase</b>	<b>Comentarios</b>
27/02/2025	09:00	12:00	0:15	2:45	Prototipos	Revisión de requerimientos y diseño inicial de prototipos.
28/02/2025	10:00	13:00	0:30	2:30	Prototipos	Creación de dos versiones de prototipos y recopilación de feedback.
02/03/2025	14:00	17:00	0:20	2:40	Prototipos	Refinamiento de prototipos basado en feedback de usuarios.
05/03/2025	09:00	11:00	0:10	1:50	Selección de Tecnologías	Investigación de tecnologías para el backend y base de datos.
07/03/2025	10:00	12:00	0:15	1:45	Selección de Tecnologías	Discusión en equipo sobre la selección de tecnologías.
09/03/2025	14:00	16:00	0:20	1:40	Gestión del Tiempo	Revisión y ajuste del cronograma del proyecto.
11/03/2025	09:00	11:00	0:10	1:50	Prototipos	Finalización de prototipos y preparación de evidencias (fotos/videos).
13/03/2025	10:00	13:00	0:30	2:30	Selección de Tecnologías	Justificación final de la selección de tecnologías.
15/03/2025	14:00	16:00	0:15	1:45	Gestión del Tiempo	Elaboración del informe final de gestión del tiempo.

**Formulario LOGT – Osman Emanuel de León García 23428**

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
27/02/2025	09:00	12:00	0:15	2:45	Diagramas de Clases	Revisión de requerimientos y diseño inicial del diagrama de clases.
28/02/2025	10:00	13:00	0:30	2:30	Diagramas de Clases	Elaboración del diagrama de clases preliminar.
03/03/2025	14:00	17:00	0:20	2:40	Diagramas de Clases	Revisión y ajuste del diagrama de clases.
05/03/2025	09:00	11:00	0:10	1:50	Persistencia	Diseño del diagrama de clases persistentes y diagrama entidad-relación.
07/03/2025	10:00	12:00	0:15	1:45	Selección de Tecnologías	Investigación de tecnologías para el backend y base de datos.
09/03/2025	14:00	16:00	0:20	1:40	Gestión del Tiempo	Revisión y ajuste del cronograma del proyecto.
11/03/2025	09:00	11:00	0:10	1:50	Persistencia	Validación del diagrama entidad-relación con el equipo.
13/03/2025	10:00	13:00	0:30	2:30	Selección de Tecnologías	Justificación final de la selección de tecnologías.
15/03/2025	14:00	16:00	0:15	1:45	Gestión del Tiempo	Elaboración del informe final de gestión del tiempo.