

# Instituto de Profesorado Sedes Sapientiae



Práctica Profesionalizante II



**FINTEK** es una herramienta diseñada para optimizar la administración de ingresos, egresos, ahorros e inversiones. Su objetivo es brindar a los usuarios una visión clara y en tiempo real de su situación financiera, facilitando la toma de decisiones estratégicas.

Equipo formado por:

Figún, Valentín

Morales, Matías Valentino

Haunau, Paula Rocío

## Introducción

El presente trabajo corresponde a la materia **Práctica Profesionalizante** del segundo año de la **Tecnicatura Superior en Análisis y Desarrollo de Software**. En este proyecto, se aplicarán los principios de la **programación orientada a objetos (POO)** para desarrollar una solución informática que responda a una problemática del mundo real. El objetivo es no solo consolidar los conocimientos adquiridos en el diseño y desarrollo de software, sino también fomentar el trabajo en equipo, la planificación y la implementación de soluciones eficientes y escalables.

El proyecto desarrollado se denomina **Fintek** y consiste en un **gestor de finanzas personales**. Su finalidad es brindar a los usuarios una herramienta intuitiva y funcional para administrar sus ingresos, egresos, ahorros e inversiones. En un contexto donde la planificación financiera se ha vuelto fundamental, **Fintek** busca facilitar la toma de decisiones económicas mediante el registro y análisis de movimientos financieros.

A lo largo del desarrollo del proyecto, se implementarán conceptos clave de la programación orientada a objetos, tales como **clases, objetos y herencia**, asegurando así un diseño modular y reutilizable. **Fintek** no solo será una herramienta de gestión financiera, sino también una aplicación práctica de los fundamentos de la ingeniería de software en un entorno real.

El sistema gestionará distintos tipos de activos e inversiones como bonos, acciones, criptomonedas y negocios, ingresos y egresos mensuales, además de controlar ahorros fijos. El objetivo es que los usuarios puedan visualizar su situación financiera en tiempo real, evaluar sus hábitos de gasto e identificar oportunidades de inversión.

Para concluir, el desarrollo de este proyecto representa una oportunidad para aplicar en un caso práctico los conocimientos adquiridos en la carrera. A través de la implementación de **Fintek**, se pondrán en práctica habilidades de análisis, diseño y programación, abordando un problema real. Esto permitirá no solo reforzar la teoría, sino también enfrentar desafíos propios del desarrollo de software en un entorno profesional.

## Marco Conceptual

El desarrollo de software se sustenta en una serie de fundamentos teóricos que permiten comprender, estructurar y llevar a cabo un proyecto informático con solidez metodológica.

### **Programación Orientada a Objetos (POO)**

La programación orientada a objetos es un paradigma que estructura el software en torno a objetos, los cuales representan entidades del sistema y agrupan datos (atributos) y comportamientos (métodos). Este enfoque promueve la modularidad, reutilización de código y escalabilidad del sistema.

Entre sus conceptos clave se destacan:

- Clase: modelo que define las características y comportamientos de un conjunto de objetos.
- Objeto: instancia concreta de una clase.
- Encapsulamiento: control del acceso a los datos internos del objeto.
- Herencia: permite derivar nuevas clases a partir de otras ya existentes.

Este enfoque permite diseñar sistemas mantenibles, seguros y fácilmente escalables.

### **Proyecto de Software**

Un proyecto de software es un conjunto de actividades planificadas con el objetivo de desarrollar una solución tecnológica para resolver una necesidad específica. Abarca desde la recolección de requisitos hasta el mantenimiento posterior a su implementación. Requiere de planificación, coordinación de recursos y gestión de tiempos y tareas.

### **UML (Lenguaje Unificado de Modelado)**

UML es un lenguaje gráfico estandarizado utilizado para representar distintos aspectos de un sistema de software. Permite documentar de forma clara la estructura y comportamiento de un sistema antes de su implementación.

Algunos de sus diagramas más comunes son:

- Diagrama de clases: muestra las entidades del sistema, sus atributos, métodos y relaciones.
- Diagrama de casos de uso: representa las funcionalidades del sistema y su interacción con los usuarios.
- Diagrama de secuencia: detalla cómo se comunican los objetos en el tiempo.

## **Metodologías Ágiles**

Son enfoques de trabajo que se basan en la entrega iterativa de software funcional, priorizando la colaboración, la adaptabilidad y la mejora continua. Estas metodologías dividen el trabajo en ciclos cortos (sprints) que permiten evaluar y ajustar el producto frecuentemente.

Sus principios clave incluyen:

- Comunicación continua con el usuario.
- Equipos autoorganizados.
- Flexibilidad ante cambios en los requerimientos.

## **Ciclo de Vida del Desarrollo de Sistemas**

Este ciclo describe las etapas que atraviesa un sistema informático desde que se detecta una necesidad hasta su mantenimiento final. Sus fases son:

1. Relevamiento: investigación del problema a resolver.
2. Análisis: definición detallada de requerimientos.
3. Diseño: planificación de la arquitectura del sistema.
4. Implementación: construcción y programación del sistema.
5. Pruebas: verificación del funcionamiento.
6. Mantenimiento: actualización y soporte posterior.

## Descripción

En la actualidad, muchas personas enfrentan dificultades para llevar un control ordenado de sus **ingresos, egresos, ahorros e inversiones**. La falta de planificación financiera puede derivar en decisiones impulsivas, gastos innecesarios y dificultades para alcanzar objetivos económicos a largo plazo. Además, la creciente diversidad de opciones de inversión, como **bonos, acciones y criptomonedas**, hace que la gestión del dinero sea cada vez más compleja.

Ante esta problemática, surgen preguntas clave: **¿Cómo puedo organizar mis finanzas de manera efectiva? ¿Cuál es el estado real de mis ingresos y gastos? ¿Estoy invirtiendo de forma adecuada? ¿Cuánto puedo ahorrar mensualmente sin comprometer mis gastos esenciales?** Estas dudas reflejan la necesidad de contar con una herramienta que centralice y facilite la administración del dinero de manera clara y accesible.

**Fintek** ofrece una solución a este problema mediante un **gestor de finanzas personales** que permite a los usuarios registrar y visualizar sus movimientos económicos en tiempo real. A través de una interfaz intuitiva, podrán clasificar sus ingresos y egresos, gestionar ahorros y monitorear sus inversiones. Con funciones de análisis y reportes personalizados, la aplicación busca proporcionar una visión clara de la situación financiera del usuario, ayudándolo a tomar **decisiones informadas y estratégicas** para optimizar sus recursos y alcanzar sus metas económicas.

## Objetivos del Proyecto

**Objetivos Generales:** Gestionar operaciones financieras y generar reportes.

**Objetivos Específicos:**

- Visualización de balances entre cuentas.
- Generación de reportes con los movimientos realizados por el usuario.
- Creación y edición de metas de ahorro.

**Alcance:**

- Incluye:
  - Registro, modificación y eliminación de usuarios.
  - Gestión completa de ingresos, egresos, ahorros e inversiones.
  - Visualización gráfica de balances.
  - Reportes personalizados por período o categoría.
- No incluye :
  - Operaciones de compra/venta de activos en línea.

**Usuarios Destinatarios:**

- Personas adultas con ingresos fijos o variables.
- Usuarios sin conocimientos técnicos, pero que requieren una herramienta intuitiva para el control de su economía personal.

# Requerimientos

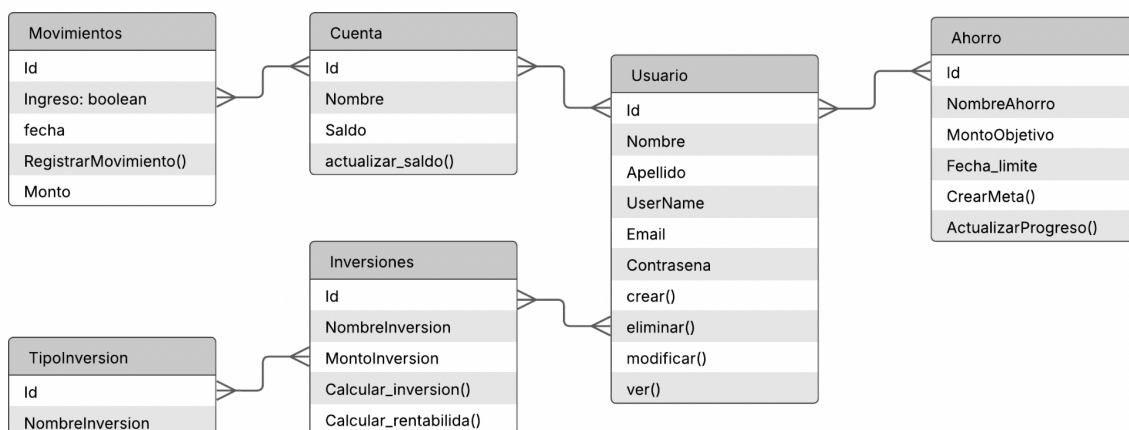
## **Funcionales**

1. **Gestión de Usuarios:** Permitir que los usuarios se registren en el sistema y accedan mediante autenticación.
  - a. Registro.
  - b. Autenticación.
  - c. Modificación.
  - d. Eliminar.
2. **Gestión de Movimientos:**
  - a. Registrar. Categorizarlos y almacenarlos en el sistema.
  - b. Modificar
  - c. Eliminar movimientos.
3. **Administración de Ahorros:**
  - a. Crear metas de ahorro.
  - b. Modificar metas de ahorro.
  - c. Eliminar metas de ahorro.
4. **Gestión de Inversiones:**
  - a. Crear.
  - b. Modificar.
  - c. Eliminar.
5. **Visualización de Balances:** Mostrar el balance general de cuentas del usuario en tiempo real.
6. **Generación de Reportes:** Crear reportes con los movimientos financieros del usuario.

## No Funcionales

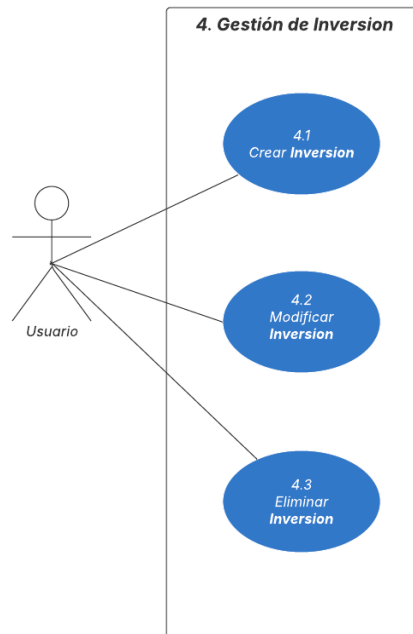
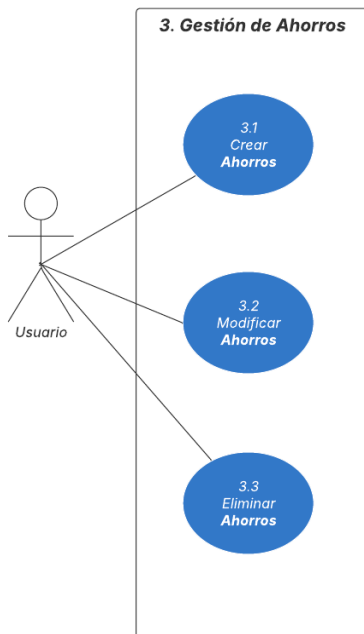
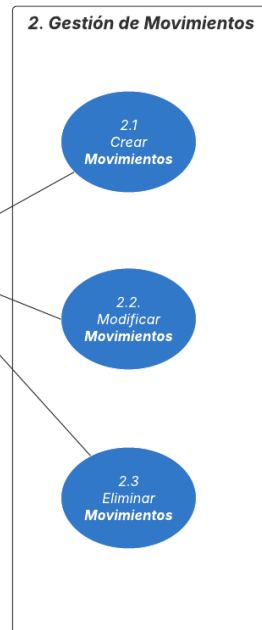
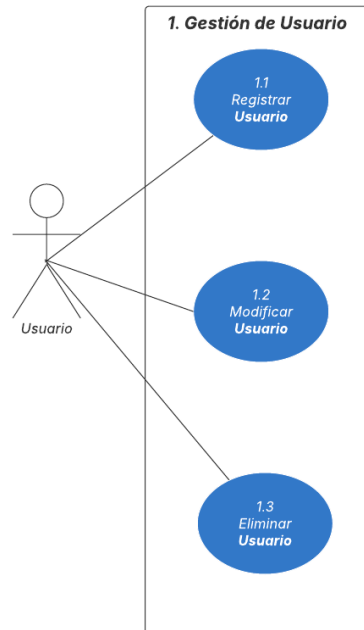
1. **Seguridad:** Implementación de mecanismos de autenticación segura y cifrado de datos.
2. **Escalabilidad:** Capacidad de expandir la aplicación sin afectar el rendimiento.
3. **Disponibilidad:** Garantizar acceso continuo al sistema con mínima interrupción.
4. **Rendimiento:** Respuesta rápida a las interacciones del usuario.
5. **Compatibilidad:** Accesibilidad desde dispositivos móviles y computadoras.
6. **Usabilidad:** Diseño amigable que facilite la experiencia del usuario.
7. **Mantenimiento:** Código modular para facilitar la actualización y corrección de errores.
8. **Interfaz Intuitiva:** Facilitar la navegación y el uso del sistema para todos los usuarios.
9. **Alertas y Notificaciones:** Notificar a los usuarios sobre movimientos importantes y vencimientos de metas.

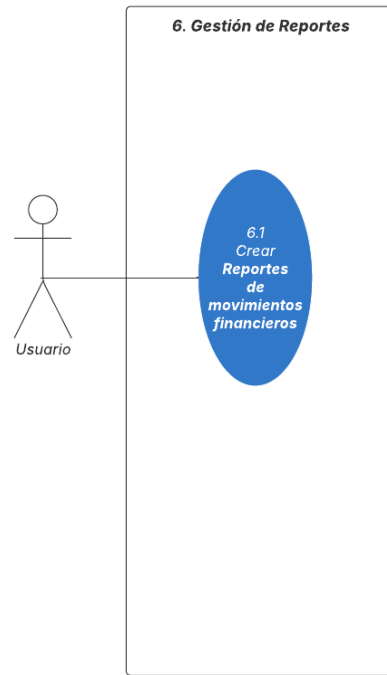
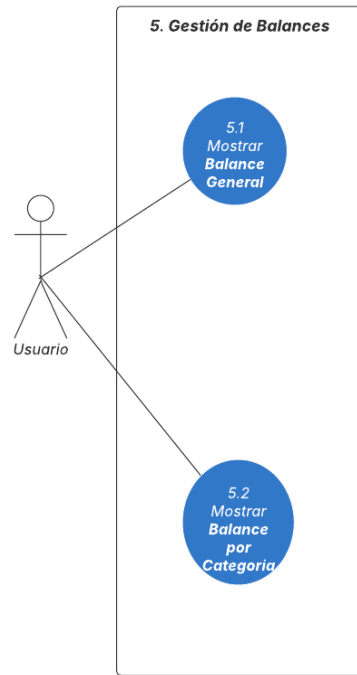
## Diagrama de Clases



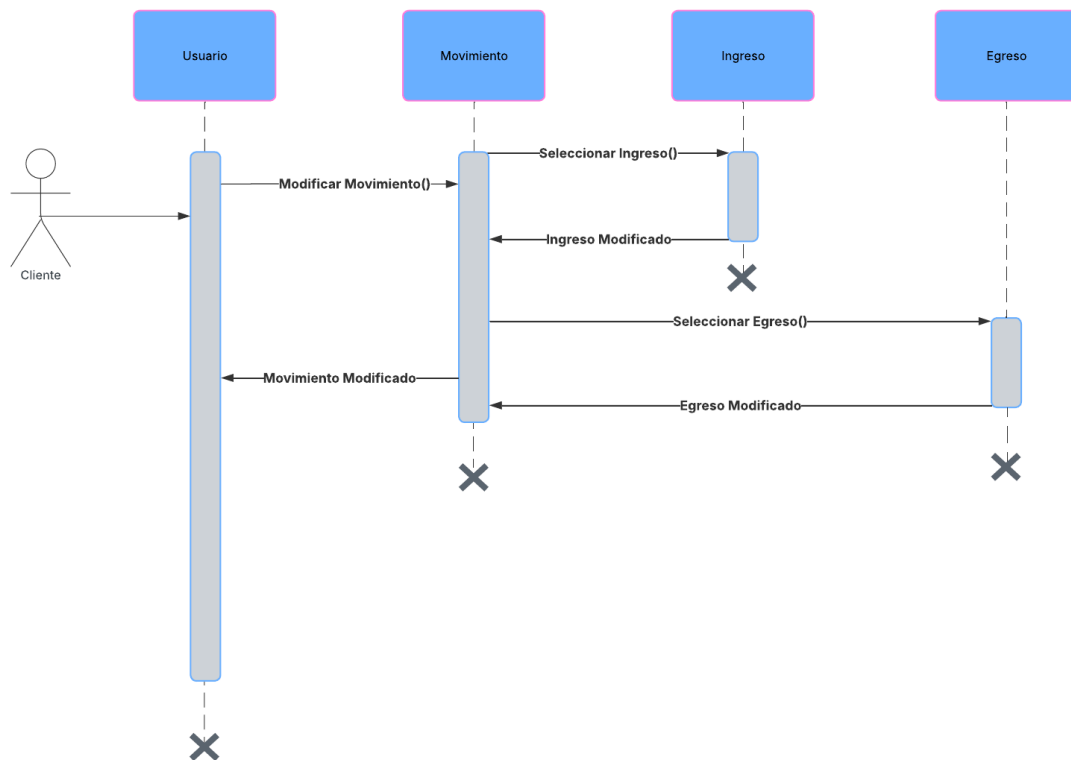
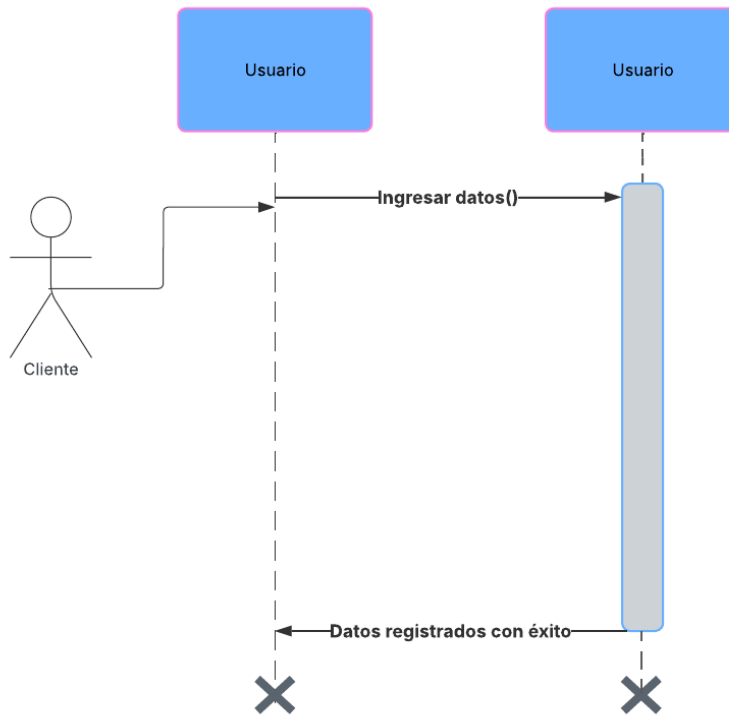


# Diagramas de Casos de Uso



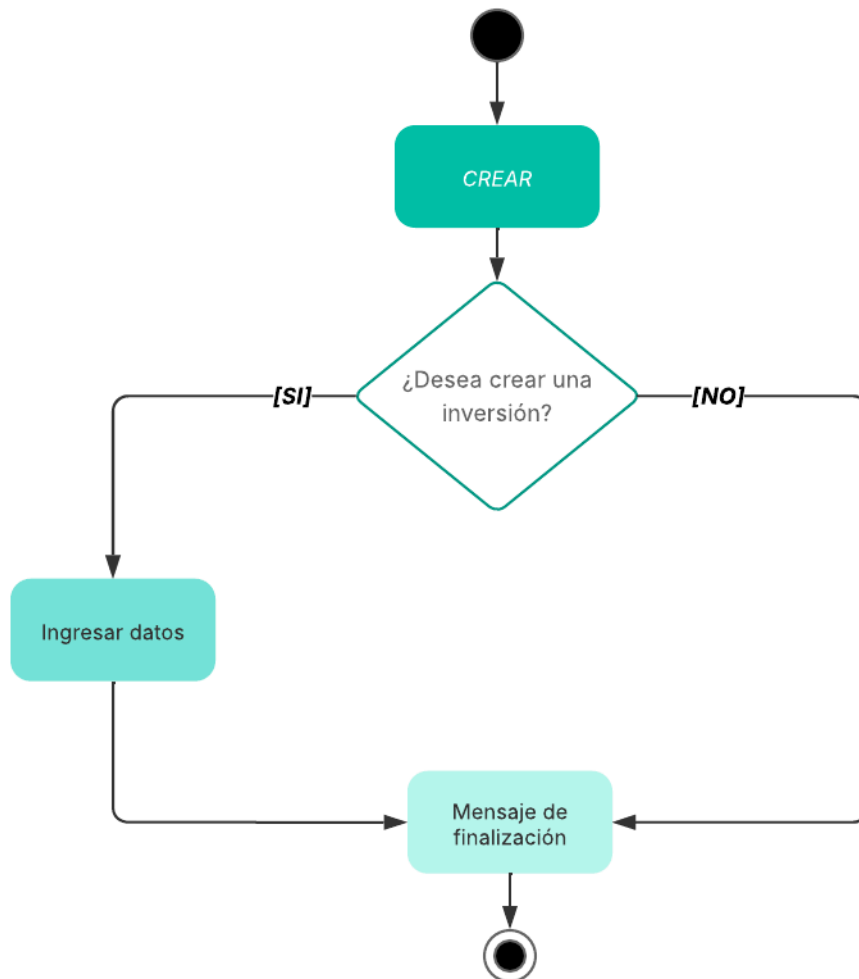


## Diagramas de Secuencia

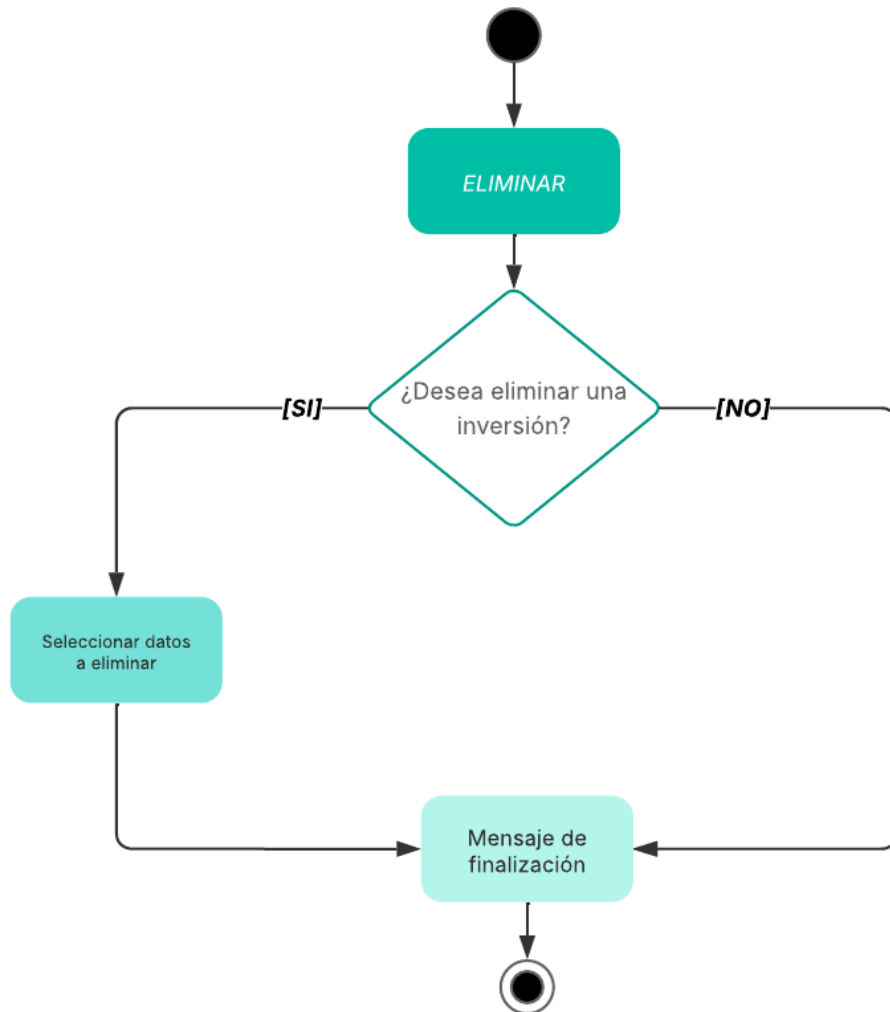


# Diagramas de Actividades

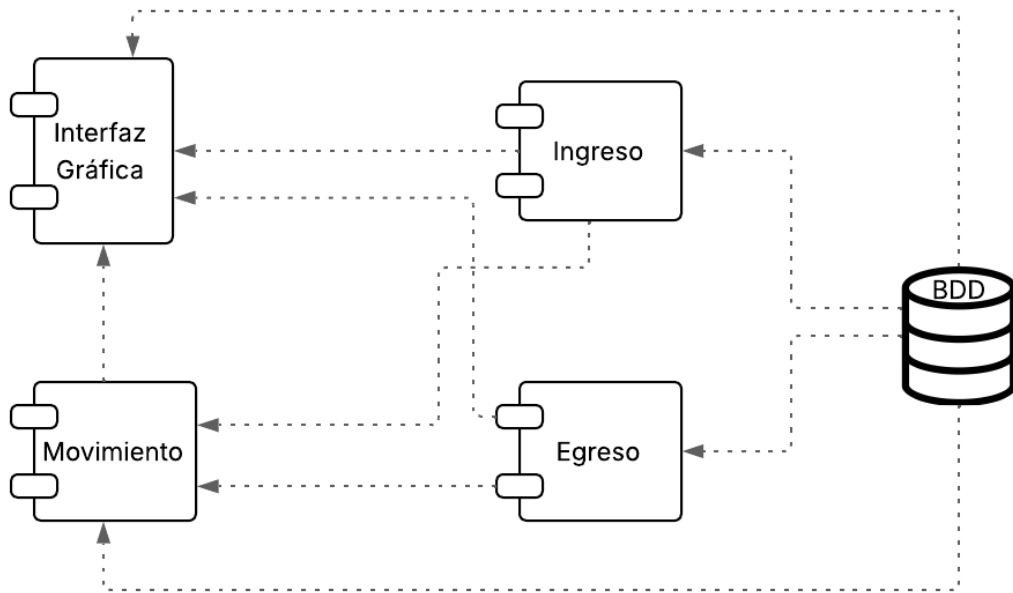
## Gestión de Inversiones



## Gestión de Inversiones



## Diagramas de Componentes



## Escenarios y Subescenarios

1. Gestión de Usuarios
  - 1.1. Registrar Usuario:
    - 1.1.1. Verificar existencia de Usuario. Si el Usuario NO existe, crear usuario completando los datos.
    - 1.1.2. Verificar existencia de Usuario. Si el Usuario existe, mensaje: 'Usuario ya registrado'
  - 1.2. Modificación de usuario:
    - 1.2.1. Verificar existencia de Usuario. Si el Usuario NO existe, no se puede modificar un usuario no existente.
    - 1.2.2. Verificar existencia de Usuario. Si el Usuario existe, modificar los datos solicitados.
  - 1.3. Eliminación de usuario:
    - 1.3.1. Verificar existencia de Usuario. Si el Usuario NO existe, no se puede eliminar un usuario no existente.
    - 1.3.2. Verificar existencia de Usuario. Si el Usuario existe, eliminar sus datos.
2. Gestión de Movimientos
  - 2.1. Registrar movimiento:

- 2.1.1. Completar los datos requeridos. Si se completan los datos obligatorios, 'Movimiento registrado con éxito'.
    - 2.1.2. Completar los datos requeridos. Si no se completan los datos obligatorios, 'Completar todos los datos necesarios'.
  - 2.2. Modificar movimiento
    - 2.2.1. El usuario selecciona el movimiento que desea modificar.
    - 2.2.2. El sistema permite cambiar los datos del movimiento seleccionado (por ejemplo: monto, categoría, fecha, etc.).
    - 2.2.3. Si la modificación se realiza correctamente, se muestra el mensaje: "Movimiento modificado con éxito."
    - 2.2.4. Si ocurre un error durante la modificación, se muestra el mensaje: 'Error al modificar el movimiento.'
  - 2.3. Eliminar movimiento:
    - 2.3.1. El usuario selecciona el movimiento que desea eliminar.
    - 2.3.2. Si el movimiento se elimina con éxito, se muestra el mensaje: 'Movimiento eliminado con éxito'.
    - 2.3.3. Si el movimiento se ve interrumpido, ya sea por falla en el sistema o elección del usuario, se muestra el mensaje: 'Movimiento no ha sido eliminado'.
- 3. Gestión de Ahorros
  - 3.1. Crear ahorros
    - 3.1.1. El usuario fija un monto de ahorro.
  - 3.2. Modificar ahorros
    - 3.2.1. El usuario selecciona el monto a modificar.
    - 3.2.2. Si hay cambios en el monto, se muestra un mensaje: 'El monto ha sido modificado'.
  - 3.3. Eliminar ahorros
    - 3.3.1. El usuario selecciona el monto a eliminar.
    - 3.3.2. Si se elimina el monto seleccionado, se muestra el mensaje: 'Eliminado con éxito'.
- 4. Gestión de Inversiones
  - 4.1. Crear inversión
    - 4.1.1. El usuario selecciona el tipo de inversión que quiere generar.
    - 4.1.2. El usuario establece el monto que quiere invertir.
    - 4.1.3. Si se crea correctamente, se muestra el mensaje: 'Creado con éxito'.
  - 4.2. Modificar inversión
    - 4.2.1. El usuario selecciona la inversión que quiere modificar.
    - 4.2.2. Si hay cambios en la inversión, se muestra un mensaje: 'La inversión ha sido modificada'.
  - 4.3. Eliminar inversión

- 4.3.1. El usuario selecciona el monto a eliminar.
  - 4.3.2. Si se elimina el monto seleccionado, se muestra el mensaje: 'Eliminado con éxito'.
- 5. Gestión de Balances
  - 5.1. Mostrar balance general
    - 5.1.1. El usuario podrá visualizar el balance total proveniente de ingresos y egresos.
  - 5.2. Mostrar balance por categoría
    - 5.2.1. El usuario podrá visualizar el balance filtrando por tipo de activos.
- 6. Generación de Reportes
  - 6.1. Crear reportes de movimientos financieros
    - 6.1.1. El sistema crea automáticamente reportes de cada usuario.
    - 6.1.2. El usuario selecciona el período de tiempo que quiere visualizar.
    - 6.1.3. El usuario tendrá la posibilidad de descargar un documento en donde se puede visualizar todos sus movimientos financieros.

## Desarrollo

La **metodología de desarrollo** adoptada para este proyecto se basó en un enfoque colaborativo y autodirigido, centrado en el trabajo grupal fuera del horario de clases. Si bien no se siguió de forma estricta una metodología ágil formal como Scrum, se tomaron elementos fundamentales de este tipo de enfoques, tales como la planificación por etapas, la distribución de tareas, y la coordinación constante entre los integrantes del equipo.

El grupo decidió organizar encuentros extracurriculares con el objetivo de avanzar progresivamente en el desarrollo del proyecto. Estas reuniones permitieron establecer objetivos semanales, revisar avances, resolver dudas en conjunto y tomar decisiones consensuadas respecto a la dirección del trabajo. Esta modalidad también fomentó la responsabilidad compartida y la autogestión, aspectos claves en contextos donde no existe una supervisión constante.

Durante estos espacios de trabajo, cada integrante asumió un rol activo en distintas fases del proyecto: análisis, documentación, diseño de diagramas, redacción del marco conceptual, organización de los entregables, entre otros. La flexibilidad de esta metodología permitió adaptarse a los tiempos y disponibilidades de los



participantes, respetando los plazos establecidos para la entrega sin comprometer la calidad del producto final.

Durante el proceso de diseño y documentación del sistema se utilizaron **diversas herramientas** que facilitaron la organización del proyecto, el modelado del sistema y la presentación de los entregables. Las principales herramientas seleccionadas fueron:

- **Google Docs / Google Drive:** Se utilizó para la redacción colaborativa de documentos, permitiendo trabajar en tiempo real entre los integrantes del equipo, organizar las entregas y compartir archivos relevantes.
- **Lucidchart:** Herramienta utilizada para el diseño de diagramas de clases y casos de uso. Su interfaz intuitiva y sus funciones específicas para diagramas UML facilitaron la representación visual del sistema.

## Conclusiones Finales

### **Conclusión del trabajo**

A lo largo del desarrollo del presente proyecto, se logró aplicar de manera efectiva los conocimientos adquiridos. El sistema diseñado responde a una problemática concreta del ámbito cotidiano: la dificultad de llevar un control financiero personal ordenado y eficiente.

La elaboración del proyecto permitió recorrer todas las etapas del ciclo de vida del desarrollo de software: desde el relevamiento, análisis y diseño, hasta la especificación de requerimientos y modelado. Cada fase fue abordada con responsabilidad y compromiso, permitiendo no solo consolidar saberes técnicos, sino también fortalecer la capacidad de pensar soluciones prácticas a problemas reales.

Asimismo, el trabajo aportó una visión integral del proceso de creación de software, desde su planificación hasta la documentación, demostrando la importancia de un enfoque sistemático y organizado para garantizar la calidad del producto final.

## **Conclusión del proceso grupal**

El proceso grupal fue una experiencia enriquecedora tanto en lo académico como en lo humano. La distribución de tareas, el respeto por los tiempos y la comunicación constante fueron pilares fundamentales para el avance ordenado del proyecto.

Cada integrante del equipo pudo aportar sus fortalezas personales, generando un entorno de colaboración que favoreció el aprendizaje conjunto. Las decisiones se tomaron de forma consensuada, priorizando el cumplimiento de los objetivos y el respeto por las ideas de todos los participantes.

El resultado final refleja no solo el esfuerzo técnico, sino también el crecimiento colectivo que se dio a lo largo del proceso.

## **Bibliografía**

- "Ingeniería del software. Un enfoque práctico. - Roger S. Pressman, Ph.D."
- <https://www.computerweekly.com/es/definicion/Programacion-orientada-a-objetos-OOP/>
- <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
- <https://www.timecamp.com/es/planner/glossary/metodologias-agiles/>