



ugr | Universidad
de **Granada**

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Plataforma web para la gestión de contabilidad personal

Autor

Jesús Leyva Fernández

Directores

Manuel Jesús Cobo Martín



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

—
Granada, Junio de 2024



Plataforma web para la gestión de contabilidad personal

Autor

Jesús Leyva Fernández

Directores

Manuel Jesús Cobo Martín

Plataforma web para la gestión de contabilidad personal

Jesús Leyva Fernández

Palabras clave: API Rest, finanzas personales, gastos compartidos, aplicación web, bases de datos, estadísticas, Java, Spring, Vue.js, MySQL.

Resumen

La gestión financiera de las finanzas personales y compartidas es esencial para mantener un control efectivo sobre los recursos económicos y tomar decisiones informadas. En la actualidad, existen múltiples aplicaciones móviles que permiten gestionar estos aspectos, pero pocas plataformas accesibles desde PC ofrecen una solución integrada que combine ambas funcionalidades.

Este proyecto tiene como objetivo desarrollar una plataforma web que facilite la gestión de finanzas personales y compartidas, proporcionando una herramienta eficiente y accesible. La plataforma permitirá a los usuarios registrar y seguir sus gastos e ingresos, gestionar perfiles de usuario y colaborar en contabilidades compartidas. Al consolidar estas funciones en una única plataforma, se espera mejorar la organización y transparencia financiera de los usuarios, permitiéndoles tomar decisiones económicas más acertadas y optimizar el uso de sus recursos.

La implementación de esta plataforma busca ofrecer una visión integrada y completa de las finanzas, ayudando a los usuarios a mantener un control detallado de sus operaciones económicas y fomentar una gestión más eficiente y colaborativa de sus finanzas personales y compartidas.

Web Platform for Personal Accounting Management

Jesús Leyva Fernández

Keywords: REST API, personal finances, shared expenses, web application, databases, statistics, Java, Spring, Vue.js, MySQL.

Abstract

Effective financial management of personal and shared finances is essential for maintaining control over economic resources and making informed decisions. Currently, there are numerous mobile applications that facilitate these aspects, but few PC-accessible platforms offer an integrated solution that combines both functionalities.

This project aims to develop a web platform that simplifies the management of personal and shared finances, providing an efficient and accessible tool. The platform will allow users to record and track their expenses and income, manage user profiles, and collaborate on shared accounts. By consolidating these functions into a single platform, the goal is to enhance the organization and transparency of users' finances, enabling them to make better financial decisions and optimize the use of their resources.

The implementation of this platform seeks to provide an integrated and comprehensive view of finances, helping users maintain detailed control of their financial operations and promoting more efficient and collaborative management of their personal and shared finances.

Yo, **Jesús Leyva Fernández**, alumno de la titulación Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77389915-K, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Jesús Leyva Fernández

Granada a 20 de Junio de 2024 .

D. Manuel Jesús Cobo Martín,
Profesor del Departamento de Ciencias
de la Computación e Inteligencia Artificial
de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Plataforma web para la gestión de contabilidad personal*** ha sido realizado bajo su supervisión por **Jesús Leyva Fernández (alumno)**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada
a 20 de Junio de 2024.

El director:

Manuel Jesús Cobo Martín

Agradecimientos

A mis familiares y especialmente a mi madre, por su amor incondicional, comprensión y apoyo constante me ha llevado a ser la persona que soy hoy. A mis amigos por acompañarme a lo largo de este camino. A Paula, por su paciencia y cariño durante esta etapa.

A todos ellos, mi más sincero agradecimiento.

ÍNDICE

Capítulo 1: Introducción.....	1
1.1. Motivación.....	1
1.2. Objetivos.....	1
1.3. Herramientas similares.....	2
1.4. Estructura de memoria.....	3
Capítulo 2: Planificación.....	4
2.1. Metodología de desarrollo.....	4
2.2. Planificación del proyecto.....	5
2.2.1. Diagram Gantt.....	5
2.3. Organización.....	6
2.3.1. Roles y responsabilidades.....	6
2.3.2. Recursos.....	7
2.4. Costes.....	8
2.5. Riesgos.....	9
Capítulo 3: Análisis del sistema.....	10
3.1. Catálogo de requisitos.....	10
3.1.1. Requisitos funcionales.....	10
3.1.2. Requisitos no funcionales.....	16
3.1.3. Requisitos de información.....	18
3.1.4. Reglas de negocio.....	20
3.2. Casos de uso.....	22
3.2.1. Actores.....	22
3.2.2. Diagramas de los casos de uso.....	24
3.2.3. Descripción de los casos de uso.....	25
3.3. Modelo conceptual de datos.....	32
3.3.1. Entidades del modelo conceptual de datos.....	32
Capítulo 4: Diseño del sistema.....	34
4.1. Arquitectura del sistema.....	34
4.1.1. Arquitectura física.....	34
4.1.2. Arquitectura lógica.....	34
4.2. Diseño físico de datos.....	36
4.3. Patrones de diseño.....	38
4.4. Diseño de los servicios.....	39
4.5. Diseño API Rest.....	40
4.6. Diseño de la interfaz de usuario.....	46
4.6.1. Pantalla inicial.....	46
4.6.2. Pantalla principal.....	48
4.6.3. Vistas.....	49
Capítulo 5: Construcción del sistema.....	59
5.1. Implementación.....	59
5.1.1. Java y Spring Boot.....	59
5.1.2. Vue.....	59
5.1.3. MySQL.....	60
5.2. Entorno de desarrollo.....	60

5.2.1. IntelliJ IDEA.....	60
5.3. Herramientas adicionales.....	61
5.3.1. Google Drive.....	61
5.3.2. Draw.io.....	61
5.3.3. Github.....	61
5.3.4. Postman.....	62
Capítulo 6: Pruebas del sistema.....	63
6.1. Estrategia.....	63
6.2. Entorno de pruebas.....	63
6.3. Pruebas del sistema.....	63
6.3.1. Pruebas funcionales.....	63
6.3.2. Pruebas no funcionales.....	70
Capítulo 7: Manual de instalación e implementación.....	73
7.1. Requisito previos.....	73
7.2. Inventario de componentes.....	73
7.3. Procedimiento de instalación.....	74
Capítulo 8: Manual de usuario.....	75
8.1. Características principales.....	75
8.2. Requisitos previos.....	75
8.3. Uso del sistema.....	75
8.3.1. Registro e inicio de sesión en la aplicación.....	76
8.3.2. Área de usuario.....	79
8.3.3. Barra lateral de navegación.....	81
8.3.4. Vistas.....	82
8.3.5. Home.....	85
8.3.6. Gastos.....	86
8.3.7. Ingresos.....	88
8.3.8. Evolución.....	88
8.3.9. Operaciones.....	89
8.3.10. Contabilidad compartida.....	90
Capítulo 9: Conclusiones.....	93
9.1. Análisis de objetivos.....	93
9.2. Lecciones aprendidas.....	94
9.3. Trabajo futuro.....	94
Bibliografía.....	96

ÍNDICE DE FIGURAS

Figura 2.1: Diagrama Gantt.....	6
Figura 3.1: Actores del sistema.....	22
Figura 3.2: Casos de uso usuario/visualizador.....	24
Figura 3.3: Casos de uso editor.....	25
Figura 3.3: Casos de uso creador.....	25
Figura 3.4: Modelo conceptual de datos.....	32
Figura 4.1: Arquitectura lógica.....	35
Figura 4.2: Diagrama entidad-relación.....	37
Figura 4.3: Paso a tablas.....	37
Figura 4.4: Diagrama de clases.....	39
Figura 4.5: Diseño de pantalla inicial econoMe.....	46
Figura 4.6: Diseño de formulario registro de usuario.....	47
Figura 4.7: Diseño de pantalla principal.....	48
Figura 4.8: Diseño de home.....	49
Figura 4.9: Diseño de gastos.....	50
Figura 4.10: Diseño de ingresos.....	51
Figura 4.11: Diseño de evolución.....	52
Figura 4.12: Diseño de operaciones.....	53
Figura 4.13: Diseño de contabilidades compartidas.....	54
Figura 4.14: Diseño de formulario operación.....	55
Figura 4.15: Diseño de formulario contabilidad.....	56
Figura 4.16: Diseño de formulario modificar detalles de usuario.....	57
Figura 4.17: Diseño de formulario modificar contraseña.....	58
Figura 8.1: Pantalla principal.....	76
Figura 8.2: Registro de usuario.....	77
Figura 8.3: Inicio de sesión.....	78
Figura 8.4: Página home por defecto.....	79
Figura 8.5: Área de usuario.....	79
Figura 8.6: Datos usuario.....	80
Figura 8.7: Cambiar contraseña.....	80
Figura 8.8: Barra lateral de navegación.....	81
Figura 8.9: Botón de información de contabilidad.....	82
Figura 8.10: Modal de información de contabilidad.....	82
Figura 8.11: Botón para añadir operación.....	83
Figura 8.12: Modal para añadir operación.....	84
Figura 8.13: Intervalo para filtrar datos y estadísticas.....	85
Figura 8.14: Vista Home.....	86
Figura 8.15: Vista Gastos.....	87
Figura 8.16: Vista Ingresos.....	88
Figura 8.17: Vista Evolución.....	89
Figura 8.18: Vista Operaciones.....	90
Figura 8.19: Filtro aplicado en Vista Operaciones.....	90
Figura 8.20: Vista Contabilidad Compartida.....	91

ÍNDICE DE TABLAS

Tabla 2.1: Coste humano del proyecto.....	8
Tabla 2.2: Tabla de riesgos asociados al proyecto.....	9
Tabla 3.1: RF-01 (Registro de un usuario).....	10
Tabla 3.2: RF-02 (Inicio de sesión de un usuario).....	10
Tabla 3.3: RF-03 (Cerrar sesión).....	11
Tabla 3.4: RF-04 (Visualizar perfil de usuario).....	11
Tabla 3.5: RF-05 (Modificar datos de usuario).....	11
Tabla 3.6: RF-06 (Aregar una contabilidad).....	12
Tabla 3.7: RF-07 (Modificar una contabilidad).....	12
Tabla 3.8: RF-08 (Visualizar una contabilidad).....	12
Tabla 3.9: RF-09 (Eliminar una contabilidad).....	13
Tabla 3.10: RF-10 (Aregar un gasto/ingreso).....	13
Tabla 3.11: RF-11 (Visualizar un gasto/ingreso).....	13
Tabla 3.12: RF-12 (Editar un gasto/ingreso).....	14
Tabla 3.13: RF-13 (Eliminar un gasto/ingreso).....	14
Tabla 3.14: RF-14 (Aregar un editor).....	14
Tabla 3.15: RF-15 (Dar de baja un editor).....	15
Tabla 3.16: RF-16 (Aregar un visualizador).....	15
Tabla 3.17: RF-17 (Dar de baja un visualizador).....	15
Tabla 3.18: RF-18 (Visualizar estadísticas).....	16
Tabla 3.19: RNF-01 (Seguridad).....	16
Tabla 3.20: RNF-02 (Usabilidad).....	17
Tabla 3.21: RNF-03 (Rendimiento).....	17
Tabla 3.22: RNF-04 (Mantenibilidad).....	18
Tabla 3.23: RNF-06 (Portabilidad).....	18
Tabla 3.24: RINF-01 (Identificación del usuario).....	18
Tabla 3.25: RINF-02 (Datos de sesión).....	19
Tabla 3.26: RINF-03 (Detalles de contabilidad).....	19
Tabla 3.27: RINF-04 (Detalles de operaciones).....	19
Tabla 3.28: RRN-01 (Autenticación de usuarios).....	20
Tabla 3.29: RRN-02 (Unicidad de correo electrónico y nombre de usuario).....	20
Tabla 3.30: RRN-03 (Acceso y permisos).....	20
Tabla 3.31: RRN-04 (Validación de datos y operaciones).....	21
Tabla 3.32: RRN-05 (Integridad de la contabilidad).....	21
Tabla 3.33: RRN-06 (Seguridad y privacidad).....	21
Tabla 3.34: RRN-07 (Reportes y estadísticas).....	22
Tabla 3.35: CU-01 (Registro de usuario).....	25
Tabla 3.36: CU-02 (Inicio de sesión).....	26
Tabla 3.37: CU-03 (Cerrar sesión).....	26
Tabla 3.38: CU-04A (Crear contabilidad).....	27
Tabla 3.39: CU-05B (Modificar contabilidad).....	27

Tabla 3.40: CU-04C (Visualizar contabilidad).....	28
Tabla 3.41: CU-04D (Eliminar contabilidad).....	28
Tabla 3.42: CU-05A (Agregar Gastos/Ingresos).....	29
Tabla 3.43: CU-05B (Visualizar Gastos/Ingresos).....	29
Tabla 3.44: CU-05C (Modificar Gasto/Ingreso).....	30
Tabla 3.45: CU-05D (Eliminar Gastos/Ingresos).....	30
Tabla 3.46: CU-06 (Gestionar Roles).....	31
Tabla 3.47: CU-07 (Visualizar estadísticas).....	31
Tabla 4.1: Endpoints de AccountingController.....	44
Tabla 4.2: Endpoints de UsersController.....	45
Tabla 4.3: Endpoints de AuthController.....	45
Tabla 6.1: PF-01 (Prueba registro de usuario).....	64
Tabla 6.2: PF-02 (Prueba inicio de sesión).....	64
Tabla 6.3: PF-03 (Prueba cierre de sesión).....	64
Tabla 6.4: PF-04 (Prueba consultar datos de usuario).....	65
Tabla 6.5: PF-05 (Prueba modificar datos de usuario).....	65
Tabla 6.6: PF-06 (Prueba modificar contraseña de usuario).....	65
Tabla 6.7: PF-07 (Prueba consultar datos de contabilidad).....	66
Tabla 6.8: PF-08 (Prueba modificar datos de contabilidad).....	66
Tabla 6.9: PF-09 (Prueba visualizar una contabilidad).....	66
Tabla 6.10: PF-10 (Eliminar contabilidad compartida).....	67
Tabla 6.11: PF-11 (Prueba agregar gasto/ingreso a una contabilidad).....	67
Tabla 6.12: PF-12 (Prueba visualizar gasto/ingreso a una contabilidad).....	67
Tabla 6.13: PF-13 (Prueba editar gasto/ingreso de una contabilidad).....	68
Tabla 6.14: PF-14 (Prueba eliminar gasto/ingreso de una contabilidad).....	68
Tabla 6.15: PF-15 (Prueba filtrar operaciones).....	68
Tabla 6.16: PF-16 (Aregar usuario editor a una contabilidad compartida).....	69
Tabla 6.17: PF-17 (Eliminar usuario editor de una contabilidad compartida).....	69
Tabla 6.18: PF-18 (Aregar usuario visualizador a una contabilidad compartida).....	69
Tabla 6.19: PF-19 (Eliminar usuario visualizador de una contabilidad compartida).....	70
Tabla 6.20: PF-20 (Visualizar estadísticas).....	70

Capítulo 1: Introducción

En este primer capítulo se proporciona una visión general del proyecto, explicando la motivación detrás de su desarrollo, los objetivos planteados y un análisis de herramientas similares disponibles en el mercado. Se pretende situar al lector en el contexto del problema a resolver y los beneficios esperados de la solución propuesta.

1.1. Motivación

En la era digital actual, la gestión de las finanzas personales se ha vuelto una necesidad vital para muchas personas. Sin embargo, a pesar del crecimiento exponencial de aplicaciones móviles dedicadas a esta tarea, existe una carencia notable en el ámbito de las plataformas de gestión de contabilidad accesibles desde PC. Este proyecto nace de la motivación por llenar ese vacío, ofreciendo una solución completa y accesible para usuarios que prefieren o necesitan gestionar sus finanzas personales y compartidas desde un ordenador.

La mayoría de las aplicaciones existentes en el mercado están enfocadas exclusivamente en dispositivos móviles, limitando el acceso y la comodidad para usuarios que trabajan con PCs en su vida diaria. Según un artículo de Kantar Media, el uso de aplicaciones de gestión financiera ha aumentado significativamente debido a la facilidad de acceso y el bajo coste, pero este crecimiento ha sido predominantemente en el ámbito móvil, dejando una brecha para los usuarios de PC [1]. Además, estas aplicaciones suelen especializarse en la gestión de gastos personales o compartidos, pero rara vez integran ambas funcionalidades de manera eficiente. Esto obliga a los usuarios a utilizar múltiples herramientas para satisfacer sus necesidades financieras, complicando la gestión y seguimiento de sus finanzas [2].

En resumen, este proyecto surge de la necesidad de llenar un vacío en el mercado de plataformas de gestión de contabilidad accesibles desde PCs, ofreciendo una solución integral que combine la gestión de gastos personales y compartidos. Al abordar estas carencias, se espera proporcionar una herramienta innovadora y útil que simplifique la vida financiera de los usuarios y les permita llevar un control más eficiente y colaborativo de sus finanzas.

1.2. Objetivos

En este apartado se exponen los objetivos primarios que se pretenden lograr a lo largo del desarrollo de este proyecto. Estos son:

- Desarrollar una plataforma web para la gestión de finanzas personales y compartidas.
- Mejorar la eficiencia en el registro y seguimiento de gastos.

Como objetivos más específicos:

- Diseñar un modelo de datos apropiado para la aplicación y sencillo para no dotar de gran complejidad al sistema.
- Implementar un sistema de autenticación eficaz y sencillo, que permita acceder a usuarios solo registrados y proteger sus datos.
- Implementar una API Rest que facilite la interacción entre el servidor y la interfaz de usuario.

1.3. Herramientas similares

Tras una búsqueda en el mercado de aplicaciones con propósitos similares, se han identificado dos aplicaciones con características parecidas:

- **Splitwise**: Splitwise [\[3\]](#) está diseñada específicamente para la gestión de gastos compartidos. Permite a los usuarios registrar gastos compartidos, calcular quién debe a quién, y enviar recordatorios de pagos. Sin embargo, no ofrece una integración completa para la gestión de finanzas personales.
- **Spendee**: Spendee [\[4\]](#) permite a los usuarios crear carteras compartidas, rastrear ingresos y gastos, y generar resúmenes financieros visuales. Ofrece diferentes planes, incluido un plan gratuito con funciones básicas y planes de pago con funcionalidades avanzadas como la categorización automática de transacciones y la sincronización bancaria.

Splitwise y Spendee son aplicaciones populares que se centran en la gestión de gastos compartidos y personales, respectivamente. Splitwise se especializa en la gestión de gastos compartidos, proporcionando una manera fácil de dividir y recordar pagos, pero carece de funcionalidades para la gestión integral de finanzas personales así como de estadísticas concretas de ingresos y gastos compartidos. Por otro lado, Spendee ofrece una solución más completa para la gestión de finanzas, incluyendo la sincronización bancaria y la categorización automática de transacciones, pero también carece de una integración eficiente de las funcionalidades para la gestión de gastos personales y compartidos en una misma plataforma, así como de una versión web. Además la visualización de operaciones por ingresos y gastos así como sus estadísticas no es sencilla y no se hace en una misma vista.

La plataforma web desarrollada en este proyecto pretende suprir las carencias identificadas en las aplicaciones existentes. Al combinar la gestión de finanzas personales y compartidas en una única solución accesible desde PC, se busca ofrecer una herramienta integral y eficiente que permita a los usuarios mantener un control detallado y colaborativo de sus finanzas. Esta plataforma incluirá estadísticas concretas de ingresos y gastos, una visualización sencilla y unificada de operaciones.

1.4. Estructura de memoria

En este apartado se presenta la estructura de la memoria del proyecto, detallando los capítulos y secciones que la componen. Cada capítulo aborda un aspecto específico del desarrollo, implementación y evaluación del sistema, proporcionando una visión completa y organizada del trabajo realizado.

Capítulo 1: Introducción

Este capítulo introduce el proyecto, presenta la motivación detrás de su realización, los objetivos planteados y una revisión de herramientas similares disponibles en el mercado.

Capítulo 2: Planificación

Describe el plan de trabajo seguido durante el proyecto. Se detalla la metodología de gestión de proyectos empleada para asegurar el cumplimiento de los plazos y objetivos.

Capítulo 3: Análisis del sistema

Se realiza un análisis exhaustivo del sistema requerido, especificando los requisitos funcionales, no funcionales, de información y reglas de negocio. Además se detallan los actores implicados y los casos de uso de estos mismos.

Capítulo 4: Diseño del sistema

Describe el diseño del sistema, incluyendo la arquitectura del software, el modelo de datos y las interfaces de usuario. Se presentan los diagramas correspondientes para ilustrar la estructura y funcionamiento del sistema.

Capítulo 5: Construcción del sistema

Explica el proceso de desarrollo del sistema, destacando las tecnologías utilizadas, los componentes clave del software y los desafíos encontrados durante la implementación.

Capítulo 6: Pruebas del sistema

Presenta las pruebas realizadas para asegurar la calidad del sistema. Incluye pruebas utilizadas para comprobar que se han satisfecho todos los requisitos, tanto funcionales como no funcionales.

Capítulo 7: Manual de instalación e implementación

Proporciona instrucciones detalladas para la instalación y configuración del sistema. Incluye los requisitos previos, el procedimiento paso a paso y las configuraciones necesarias para poner en marcha el sistema.

Capítulo 8: Manual de usuario

Ofrece una guía para los usuarios finales del sistema. Describe cómo utilizar las diferentes funcionalidades del sistema, con ejemplos prácticos y capturas de pantalla que facilitan la comprensión.

Capítulo 9: Conclusiones

Resumen de los logros del proyecto, conclusiones principales y recomendaciones para futuras mejoras y expansiones del sistema. Se reflexiona sobre el cumplimiento de los objetivos planteados y el impacto del proyecto.

Capítulo 2: Planificación

Este capítulo detalla la planificación completa del proyecto, proporcionando una guía estructurada para su ejecución. En él, se presentan las metodologías de trabajo, la organización del equipo, la asignación de roles y responsabilidades, así como una estimación detallada de los costes involucrados. Además, se identifican los posibles riesgos y se proponen estrategias de mitigación para asegurar el éxito del proyecto.

2.1. Metodología de desarrollo

Para asegurar un desarrollo eficiente y adaptativo del proyecto, se ha optado por utilizar una metodología basada en ciclos iterativos. Este enfoque permite incorporar feedback continuo, adaptarse a cambios y mejorar el producto de manera incremental.

La metodología de desarrollo basada en ciclos iterativos, también conocida como metodología incremental, se centra en dividir el proyecto en pequeñas partes manejables, denominadas iteraciones. Cada iteración implica un ciclo completo de desarrollo, desde el análisis y diseño hasta la implementación y pruebas, culminando con la entrega de una versión funcional del producto. Estas iteraciones comprenden cuatro etapas claves:

- **Análisis:** En esta fase se identifican y evalúan los requisitos específicos a abordar en el ciclo actual. Se priorizan las funcionalidades y se detallan las necesidades del sistema.
- **Diseño:** Basándose en los requisitos analizados, se elabora la arquitectura del sistema y los esquemas de datos necesarios. Se crean diagramas y modelos que guían la implementación, asegurando una estructura coherente y escalable.
- **Implementación:** Durante la implementación, se desarrollan los componentes definidos en la fase de diseño utilizando las herramientas y lenguajes de programación adecuados. Esta etapa incluye la codificación, integración y configuración de los módulos del sistema.
- **Pruebas:** Una vez completada la implementación, se realizan pruebas exhaustivas, tanto funcionales como no funcionales, para verificar que los componentes funcionan correctamente.
- **Refinamiento:** Recopilación de feedback a través de las pruebas para adaptar las posibles mejoras.

Optar por este enfoque iterativo e incremental permite una mayor adaptabilidad, ya que facilita el manejo de tareas complejas de manera más controlada y eficiente.

2.2. Planificación del proyecto

Para el desarrollo de la plataforma, se han planificado varias iteraciones, cada una enfocada en diferentes componentes y funcionalidades del sistema:

Iteración 0: Configuración del entorno de desarrollo y aprendizaje de las tecnologías Spring y Vue.js

En esta primera fase, se adquirió formación en tecnologías que nunca se habían usado como son Spring Boot y Vue.js. Además, se configuró el entorno de desarrollo con las dependencias correspondientes para el correcto funcionamiento

Iteración 1: Sistema de autenticación básico (registro e inicio de sesión)

Se diseñó y desarrolló todo lo relativo al sistema de autenticación y registro de usuarios, comprendiendo éste la seguridad y validaciones de los mismos.

Iteración 2: Gestión de usuario

Implementación de la funcionalidad de gestión de perfil de usuario, permitiendo la visualización y edición de sus datos.

Iteración 3: Gestión de contabilidad personal

Desarrollo de funcionalidades para la gestión de finanzas personales, incluyendo las diferentes vistas, agregar operaciones y visualización de estadísticas.

Iteración 4: Gestión de contabilidad compartida

Ampliación del sistema para incluir la gestión de finanzas compartidas, con sus respectivas estadísticas y funcionalidades.

Iteración 5: Gestión de operaciones de usuario

Implementación de herramienta para consultar transacciones, modificarlas y eliminarlas.

Documentación

Creación de la documentación completa del sistema, incluyendo manuales de usuario, guías de instalación y configuración, y documentación técnica detallada.

2.2.1. Diagram Gantt

Para la correcta planificación del proyecto se ha usado una herramienta llamada Diagrama Gantt [\[5\]](#). Permite organizar las tareas a través de un flujo de tiempo y visualizarlas de forma muy sencilla.



Figura 2.1: Diagrama Gantt

2.3. Organización

En esta sección se detalla la estructura organizativa del proyecto, describiendo cómo se distribuyen las responsabilidades y tareas a lo largo del ciclo de vida del desarrollo. Esta sección cubre la asignación de roles, así como las herramientas y recursos utilizados para la realización del proyecto.

2.3.1. Roles y responsabilidades

Para este proyecto se han definido dos roles principales, cada uno con responsabilidades claras y específicas para asegurar una gestión y ejecución efectiva del trabajo:

- **Director del proyecto:** Es el encargado de realizar una supervisión del proyecto y dar feedback tanto en la parte técnica como en la documentación del sistema.
- **Estudiante y desarrollador del proyecto:** Se encarga de implementar las funcionalidades del sistema. Participa en todas las partes del ciclo iterativo, asegurando un correcto desarrollo del proyecto de forma ordenada.

2.3.2. Recursos

A continuación, se especifican los diferentes recursos usados para el desarrollo del proyecto:

- **Recursos software:**

- **IntelliJ IDEA [6]:** Es el entorno de desarrollo integrado (IDE) utilizado para el desarrollo del sistema.
- **Postman [7]:** Herramienta utilizada para comprobar el correcto funcionamiento de los distintos endpoints que componen la API.
- **Github [8]:** Sistema de control de versiones usado para almacenar el código del software. Permite la colaboración y el seguimiento de cambios, facilitando la gestión del código fuente a lo largo del ciclo de vida del proyecto.
- **Google Drive [9]:** Sistema de alojamiento de archivos en la nube al que se puede acceder desde cualquier dispositivo.
- **Google Docs:** Utilizado para la creación y edición de documentos de manera colaborativa.
- **Draw.io [10]:** Herramienta compatible con Google Drive que se usa para construir diagramas de todo tipo. Es esencial para crear diagramas de flujo, diagramas de arquitectura y otros gráficos necesarios para el diseño y documentación del sistema.
- **Google Meet [11]:** Plataforma utilizada para realizar reuniones virtuales. Se ha utilizado para las reuniones de seguimiento entre director y desarrollador.
- **Maven [12]:** Herramienta de gestión y construcción de proyectos para Java. Se utiliza para manejar dependencias y construir el proyecto.
- **Npm [13]:** Sistema de gestión de paquetes para JavaScript. Utilizado principalmente para gestionar las dependencias del proyecto frontend.
- **Onlinegantt [14]:** Herramienta para la gestión de proyectos que permite crear y seguir diagramas de Gantt.

- **Recursos hardware:**

- **Ordenador portátil:** MSI GF62-8RD
 - Procesador: Intel Core i7-8750H
 - Memoria RAM: 16GB
 - Tarjeta Gráfica: NVIDIA GeForce GTX 1050Ti
 - Sistema operativo: Ubuntu 22.04.4 LTS

- **Otros recursos:** Para el aprendizaje de las nuevas tecnologías se han usado principalmente vídeos didácticos.

2.4. Costes

En esta sección se presenta un análisis detallado de los costes asociados con el desarrollo del proyecto. Estos costes incluyen tanto el coste de recursos y materiales como el coste humano asociado al desarrollo del sistema.

Respecto a los costes humanos, se ha tenido en cuenta el sueldo promedio de un desarrollador Junior en la ciudad de Granada [24], que está valorado en unos 15 euros/hora brutos. Además, se han tomado las siguientes variables para el cálculo del coste:

Tiempo de desarrollo	Fines de semana sin tarea	Promedio de horas por día	Horas totales
26/02/2024 - 30/05/2024 (94 días)	3 (6 días)	5 horas	440 horas

Tabla 2.1: Coste humano del proyecto

Teniendo en cuenta los datos anteriores nos encontramos con que el coste total humano es de 6.600 euros.

En cuanto a costes materiales solo se incluye el coste del equipo con el que se ha realizado el proyecto que es de 1000 euros.

Por lo que en total, el coste del proyecto asciende a **7600.00 euros**.

2.5. Riesgos

En este apartado se detallan los riesgos asociados al proyecto, el impacto que estos mismos podrían conllevar en el desarrollo del proyecto y la probabilidad de ocurrencia.

Riesgo	Impacto	Probabilidad de ocurrencia	Solución
Dificultad en el aprendizaje y dominio de nuevas tecnologías	Retraso en el inicio del desarrollo hasta alcanzar el nivel necesario de competencia.	Alta	Reajustar la fecha de inicio del proyecto para permitir un tiempo adecuado de aprendizaje
Desafíos inesperados en el desarrollo de ciertas funcionalidades	Retraso en el progreso de las etapas y el inicio de las siguientes	Media	Ajustar el cronograma
Caída de servicios externos utilizados en el sistema	Retrasos hasta que los servicios vuelvan a estar operativos	Baja	Ajustar el cronograma para compensar los tiempos de inactividad
Ausencia del desarrollador debido a enfermedad o accidente	Aplazamiento del desarrollo, con la duración del retraso dependiendo de la gravedad.	Baja	Replanificar el cronograma para acomodar tiempos de recuperación
Fallo o avería en el equipo utilizado para el desarrollo	Retrasos durante el tiempo que tome reparar o reemplazar el equipo	Baja	Ajustar el cronograma para incluir tiempo adicional en caso de averías

Tabla 2.2: Tabla de riesgos asociados al proyecto

Capítulo 3: Análisis del sistema

En este capítulo, se presenta una visión general del contenido de los requisitos, que incluye la especificación del modelo funcional, los casos de uso y la descripción del modelo conceptual.

3.1. Catálogo de requisitos

El Catálogo de Requisitos es un documento esencial que describe las necesidades y funcionalidades clave de un sistema. Este catálogo es fundamental para guiar el diseño, desarrollo e implementación del sistema, asegurando que se capturen de manera clara y completa todos los requisitos necesarios para su éxito.

3.1.1. Requisitos funcionales

RF-01	Registro de un usuario
Descripción	Permitir a los nuevos usuarios crear una cuenta usando un correo electrónico y una contraseña. El sistema debe validar la unicidad del correo electrónico y los criterios de seguridad de la contraseña.
Prioridad	Alta.
Observaciones	El usuario deberá llenar un formulario con los datos pertinentes para su registro.

Tabla 3.1: RF-01 (Registro de un usuario)

RF-02	Inicio de sesión de un usuario
Descripción	El sistema debe permitir a los usuarios iniciar sesión utilizando su correo electrónico y contraseña registrados. Si las credenciales son correctas, el usuario accede a su cuenta.
Prioridad	Alta.
Observaciones	Se necesita que el usuario ya haya sido registrado en la base de datos.

Tabla 3.2: RF-02 (Inicio de sesión de un usuario)

RF-03	Cerrar sesión
Descripción	El sistema debe proporcionar una opción claramente visible para que los usuarios cierren sesión de su cuenta. Al seleccionar esta opción, la sesión actual del usuario debe finalizar.
Prioridad	Alta.
Observaciones	El usuario debe haber accedido a una sesión para poder cerrarla.

Tabla 3.3: RF-03 (Cerrar sesión)

RF-04	Visualizar perfil de usuario
Descripción	El sistema debe permitir a los usuarios visualizar su perfil, mostrando la información relevante asociada a su cuenta, como el correo electrónico, nombre de usuario, y cualquier otra información personal que se haya proporcionado durante el registro o actualizado posteriormente.
Prioridad	Media.
Observaciones	El usuario debe haber accedido a una sesión para poder ver sus datos personales.

Tabla 3.4: RF-04 (Visualizar perfil de usuario)

RF-05	Modificar datos de usuario
Descripción	El sistema debe ofrecer una opción para que los usuarios editen la información de su perfil, incluyendo correo electrónico, nombre, contraseña y cualquier otra información personal permitida por la plataforma.
Prioridad	Media.
Observaciones	El usuario debe haber accedido a una sesión para poder modificar sus datos personales. Los cambios deben ser guardados y actualizados inmediatamente en la base de datos tras la confirmación del usuario.

Tabla 3.5: RF-05 (Modificar datos de usuario)

RF-06	Agregar una contabilidad
Descripción	El sistema debe permitir a los usuarios crear nuevas contabilidades, especificando detalles como el nombre de la contabilidad, una breve descripción (opcional), y configurar si será una contabilidad personal o compartida.
Prioridad	Alta.
Observaciones	El usuario debe estar registrado y haber iniciado sesión. Los usuarios deben poder crear múltiples contabilidades bajo su cuenta.

Tabla 3.6: RF-06 (Agregar una contabilidad)

RF-07	Modificar una contabilidad
Descripción	El sistema debe permitir a los usuarios modificar los detalles de una contabilidad existente que hayan creado, incluyendo el nombre de la contabilidad, la descripción y la configuración de privacidad (personal o compartida).
Prioridad	Media.
Observaciones	El usuario debe ser el propietario de la contabilidad para modificarla.

Tabla 3.7: RF-07 (Modificar una contabilidad)

RF-08	Visualizar una contabilidad
Descripción	El sistema debe permitir a los usuarios visualizar los detalles de cualquier contabilidad a la que tengan acceso, incluyendo el nombre, descripción, total de gastos e ingresos registrados, y la lista de usuarios con acceso a la contabilidad.
Prioridad	Alta.
Observaciones	Todo usuario que sea propietario de una contabilidad o que tenga rol visualizador en una contabilidad compartida podrá visualizar los detalles.

Tabla 3.8: RF-08 (Visualizar una contabilidad)

RF-09	Eliminar una contabilidad
Descripción	El sistema debe permitir a los usuarios eliminar una contabilidad que hayan creado. Esta acción debe requerir confirmación debido a su irreversibilidad.
Prioridad	Media.
Observaciones	Solo el propietario de la contabilidad debe tener la capacidad de eliminarla. Debe proporcionarse una clara advertencia sobre las consecuencias de eliminar una contabilidad, incluyendo la pérdida permanente de todos los gastos e ingresos asociados.

Tabla 3.9: RF-09 (Eliminar una contabilidad)

RF-10	Agregar un gasto/ingreso
Descripción	El sistema debe permitir a los usuarios añadir nuevos gastos/ingresos a una contabilidad seleccionada, especificando la cantidad, fecha, categoría (ej., alimentos, vivienda), y una descripción opcional.
Prioridad	Alta.
Observaciones	El usuario debe tener como mínimo una contabilidad creada o ser editor en una contabilidad compartida.

Tabla 3.10: RF-10 (Agregar un gasto/ingreso)

RF-11	Visualizar un gasto/ingreso
Descripción	El sistema debe permitir a los usuarios visualizar una lista de todos los gastos/ingresos registrados en una contabilidad seleccionada, con opciones para filtrar y ordenar por fecha, categoría, y cantidad.
Prioridad	Alta.
Observaciones	El usuario debe tener como mínimo un gasto/ingreso creado o ser editor o visualizador en una contabilidad compartida.

Tabla 3.11: RF-11 (Visualizar un gasto/ingreso)

RF-12	Editar un gasto/ingreso
Descripción	Los usuarios deben poder editar los detalles de cualquier gasto/ingreso registrado, incluyendo cantidad, fecha, categoría, y descripción.
Prioridad	Media.
Observaciones	El usuario debe tener como mínimo un gasto/ingreso creado o ser editor en una contabilidad compartida.

Tabla 3.12: RF-12 (Editar un gasto/ingreso)

RF-13	Eliminar un gasto/ingreso
Descripción	El sistema debe permitir a los usuarios eliminar un gasto/ingreso registrado de una contabilidad seleccionada.
Prioridad	Media.
Observaciones	El usuario debe tener como mínimo un gasto/ingreso creado o ser editor en una contabilidad compartida.

Tabla 3.13: RF-13 (Eliminar un gasto/ingreso)

RF-14	Agregar un editor
Descripción	El sistema debe permitir al propietario de una contabilidad agregar a otros usuarios registrados como editores, otorgándoles permisos para añadir, editar y eliminar gastos e ingresos dentro de la contabilidad compartida.
Prioridad	Alta.
Observaciones	El propietario debe poder invitar a usuarios mediante su nombre de usuario.

Tabla 3.14: RF-14 (Agregar un editor)

RF-15	Dar de baja un editor
Descripción	El sistema debe permitir al propietario de una contabilidad retirar los permisos de editor a cualquier usuario que previamente haya sido agregado como tal, revocando su capacidad para añadir, editar o eliminar gastos e ingresos..
Prioridad	Media.
Observaciones	El usuario que se da de baja debe estar registrado como editor.

Tabla 3.15: RF-15 (Dar de baja un editor)

RF-16	Agregar un visualizador
Descripción	El sistema debe permitir al propietario de una contabilidad agregar a otros usuarios registrados como visualizadores, otorgándoles permisos solo para ver los gastos e ingresos dentro de la contabilidad compartida sin la capacidad de modificarlos.
Prioridad	Alta.
Observaciones	El propietario debe poder invitar a usuarios mediante su nombre de usuario.

Tabla 3.16: RF-16 (Agregar un visualizador)

RF-17	Dar de baja un visualizador
Descripción	El sistema debe permitir al propietario de una contabilidad retirar los permisos de visualizador a cualquier usuario que previamente haya sido agregado como tal, revocando su capacidad para ver los gastos e ingresos.
Prioridad	Media.
Observaciones	El usuario que se da de baja debe estar registrado como visualizador.

Tabla 3.17: RF-17 (Dar de baja un visualizador)

RF-18	Visualizar estadísticas
Descripción	El sistema debe ofrecer una funcionalidad para visualizar estadísticas y gráficos de los gastos e ingresos, permitiendo a los usuarios ver resúmenes diarios, semanales, mensuales, y anuales.
Prioridad	Alta.
Observaciones	Las estadísticas deben ser fácilmente comprensibles

Tabla 3.18: RF-18 (Visualizar estadísticas)

3.1.2. Requisitos no funcionales

RNF-01	Seguridad
Descripción	<p>Para garantizar una seguridad total del usuario, así como de sus datos. El sistema deberá proporcionar:</p> <ul style="list-style-type: none"> • Autenticación: La aplicación soportará autenticación segura, permitiendo solo a usuarios registrados y autenticados el acceso a sus datos. • Autorización: Debe implementar controles de acceso que restrinjan el acceso a la información y funcionalidades según el rol del usuario dentro de una contabilidad compartida. • Confidencialidad: Debe garantizar la confidencialidad de los datos del usuario mediante el cifrado de datos sensibles como las contraseñas.

Tabla 3.19: RNF-01 (Seguridad)

RNF-02	Usabilidad
Descripción	<p>Para garantizar una usabilidad acorde a cualquier usuario. El sistema deberá proporcionar:</p> <ul style="list-style-type: none"> • Interfaz amigable: se debe ofrecer una interfaz de usuario clara e intuitiva que sea fácil de navegar para usuarios de todos los niveles técnicos. La interacción con la aplicación debe ser lógica y fluida, con elementos de interfaz consistentes y predictibles que faciliten una curva de aprendizaje rápida. • Accesibilidad: La aplicación debe ser diseñada pensando en la inclusión, permitiendo que usuarios con diversas capacidades puedan utilizarla eficazmente.

Tabla 3.20: RNF-02 (Usabilidad)

RNF-03	Rendimiento
Descripción	<p>El sistema debe responder de manera rápida a cualquier petición que puedan realizar los distintos usuarios. Para ello se hará énfasis en:</p> <ul style="list-style-type: none"> • Tiempo de respuesta: El software debe siempre responder en un tiempo. • Capacidad de carga: Debe manejar gran cantidad de usuarios y operaciones y que ello no afecte al rendimiento.

Tabla 3.21: RNF-03 (Rendimiento)

RNF-04	Mantenibilidad
Descripción	<p>Para garantizar la mantenibilidad del sistema, es esencial asegurar:</p> <ul style="list-style-type: none"> • Modularidad: El código debe estar bien organizado en módulos lógicos que permitan mantenimientos y actualizaciones eficientes. • Documentación: El código fuente debe estar adecuadamente documentado para facilitar la comprensión y futuras modificaciones por parte de nuevos desarrolladores.

Tabla 3.22: RNF-04 (Mantenibilidad)

RNF-05	Portabilidad
Descripción	Debe garantizarse una compatibilidad total del sistema entre los diferentes sistemas operativos y navegadores del mercado.

Tabla 3.23: RNF-06 (Portabilidad)

3.1.3. Requisitos de información

RINF-01	Identificación del usuario
Descripción	Almacenar datos por usuario al registrarse o cuando se modifica cualquiera de ellos.
Datos específicos	<ul style="list-style-type: none"> • Nombre de usuario • Nombre • Apellidos • Correo electrónico • Contraseña
Prioridad	Alta
Observaciones	<ul style="list-style-type: none"> • El correo electrónico así como el nombre de usuario deben ser únicos en el sistema. El correo además deberá tener un "@" para concatenar el dominio.

Tabla 3.24: RINF-01 (Identificación del usuario)

RINF-02	Datos de sesión
Descripción	Almacenar datos por usuario asociados a la sesión actual.
Datos específicos	<ul style="list-style-type: none"> Token de usuario
Prioridad	Alta
Observaciones	<ul style="list-style-type: none"> El token tendrá expiración de una hora y se usará siempre para realizar peticiones.

Tabla 3.25: RINF-02 (Datos de sesión)

RINF-03	Detalles de contabilidad
Descripción	Almacenar y gestionar información detallada de cada contabilidad
Datos específicos	<ul style="list-style-type: none"> Nombre de la contabilidad Descripción Nombre del usuario creador Tipo
Prioridad	Alta
Observaciones	<ul style="list-style-type: none"> La descripción puede ser opcional El tipo de contabilidad es personal o compartida

Tabla 3.26: RINF-03 (Detalles de contabilidad)

RINF-03	Detalles de operaciones
Descripción	Almacenar y gestionar información detallada de las operaciones de una contabilidad
Datos específicos	<ul style="list-style-type: none"> Tipo de operación Categoría Descripción Fecha de la transacción Cantidad monetaria
Prioridad	Alta
Observaciones	<ul style="list-style-type: none"> La descripción puede ser opcional El tipo de operación es gasto o ingreso

Tabla 3.27: RINF-04 (Detalles de operaciones)

3.1.4. Reglas de negocio

RRN-01	Autenticación de usuarios
Descripción	Todos los usuarios deben registrarse e iniciar sesión para acceder a sus contabilidades
Prioridad	Alta

Tabla 3.28: RRN-01 (Autenticación de usuarios)

RRN-02	Unicidad de correo electrónico y nombre de usuario
Descripción	El correo electrónico y el nombre de usuario utilizado para el registro de usuario debe ser único en el sistema para evitar duplicidades
Prioridad	Alta

Tabla 3.29: RRN-02 (Unicidad de correo electrónico y nombre de usuario)

RRN-03	Acceso y permisos
Descripción	<ul style="list-style-type: none"> • Solo los usuarios con el rol adecuado (propietario, editor, visualizador) pueden realizar acciones específicas dentro de una contabilidad compartida • Los propietarios pueden agregar o eliminar editores y visualizadores, así como eliminar la contabilidad. Todos los propietarios son editores. • Los editores pueden añadir, editar o eliminar gastos e ingresos, pero no pueden eliminar la contabilidad • Los visualizadores solo pueden ver la información de gastos e ingresos y no pueden realizar cambios
Prioridad	Alta

Tabla 3.30: RRN-03 (Acceso y permisos)

RRN-04	Validación de datos y operaciones
Descripción	<ul style="list-style-type: none"> • La cantidad de las operaciones debe ser un número positivo
Prioridad	Alta

Tabla 3.31: RRN-04 (Validación de datos y operaciones)

RRN-05	Integridad de la contabilidad
Descripción	<ul style="list-style-type: none"> • No se permite eliminar una contabilidad que tenga operaciones registradas sin una confirmación explícita del usuario para prevenir pérdidas accidentales de datos • Al eliminar una contabilidad, todas las operaciones asociadas deben ser también eliminadas
Prioridad	Alta

Tabla 3.32: RRN-05 (Integridad de la contabilidad)

RRN-06	Seguridad y privacidad
Descripción	<ul style="list-style-type: none"> • Todas las sesiones de usuario deben expirar automáticamente después de un período de inactividad para proteger la cuenta. Este periodo será de una hora. • Los datos sensibles como contraseñas y detalles financieros deben ser encriptados en la base de datos
Prioridad	Alta

Tabla 3.33: RRN-06 (Seguridad y privacidad)

RRN-07	Reportes y estadísticas
Descripción	<ul style="list-style-type: none"> Los reportes generados deben reflejar el estado actual de las contabilidades hasta la última operación confirmada Los usuarios pueden generar reportes para períodos específicos seleccionados por ellos
Prioridad	Alta

Tabla 3.34: RRN-07 (Reportes y estadísticas)

3.2. Casos de uso

En esta parte se detallan todos los casos de uso del sistema, asignando cada uno de ellos a los distintos actores que forman parte del mismo.

3.2.1. Actores

En el sistema podemos encontrar 4 actores: usuario, visualizador, editor y creador de la contabilidad.

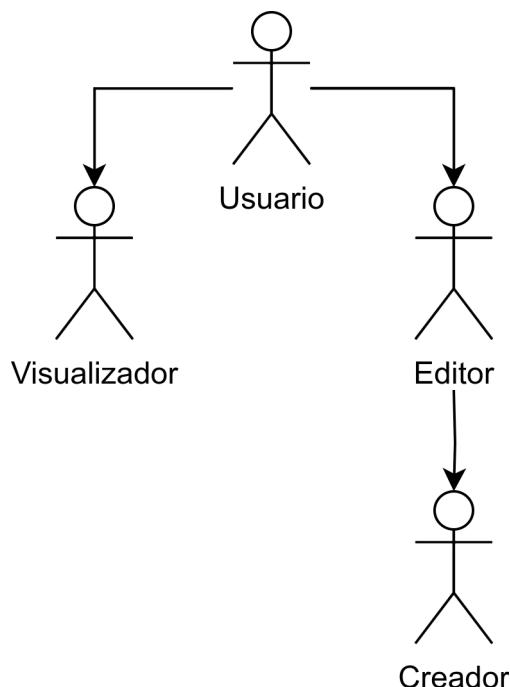


Figura 3.1: Actores del sistema

- **Usuario:** se trata del actor principal y del que derivan el resto. El usuario es aquel que se registra en el sistema y tiene acceso a las funciones básicas del mismo. Desde este rol fundamental, se especializan los otros roles, proporcionando una estructura clara y controlada para la gestión de permisos y acceso a la funcionalidad de la aplicación.
- **Visualizador:** Este actor tiene permisos para acceder y revisar los datos de las contabilidades a las cuales tiene permiso sin realizar cambios. Su función principal es observar y analizar la información presentada.
- **Editor:** Ampliando las capacidades del usuario, el editor puede modificar los datos existentes, añadir nuevos gastos o ingresos y ajustar detalles dentro de las contabilidades a las que tiene acceso. Esta capacidad de edición es vital para la actualización y el mantenimiento de registros precisos.
- **Creador:** Como el rol más avanzado, el creador puede configurar nuevas contabilidades, asignar roles a otros usuarios y tiene plenos derechos para modificar o eliminar la contabilidad. Es responsable de la administración y el buen funcionamiento de las partes de la aplicación que están bajo su control. Además, hereda todas las características del editor.

3.2.2. Diagramas de los casos de uso

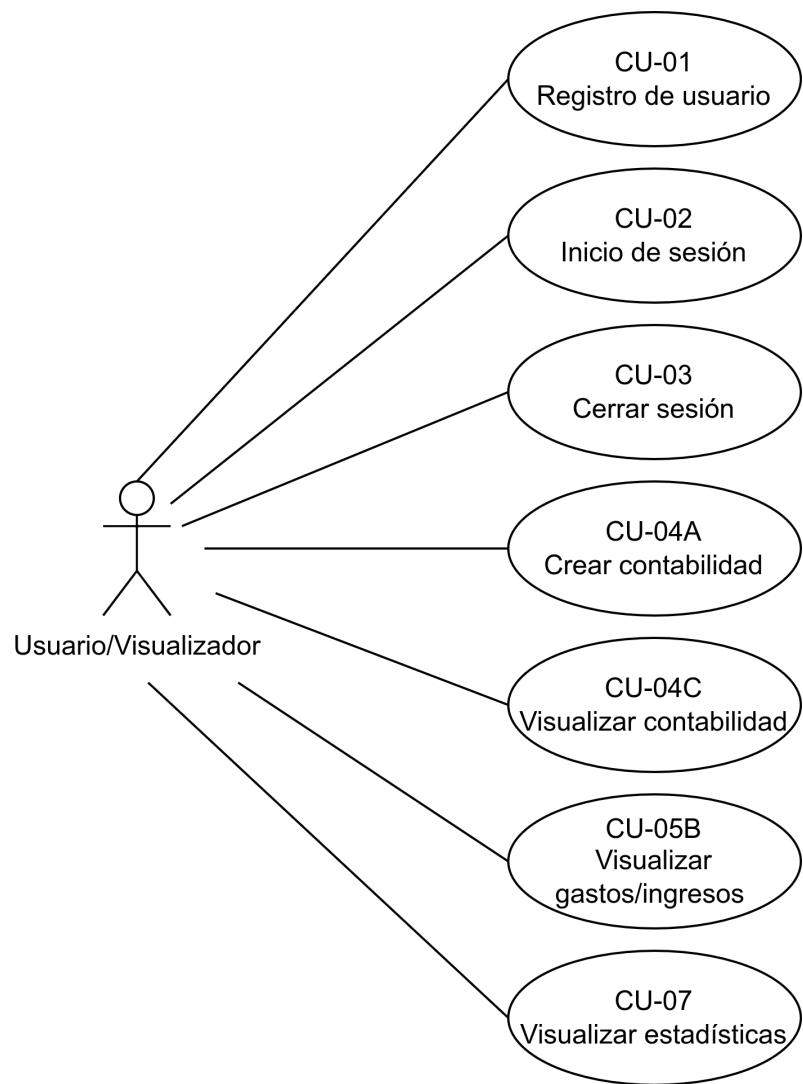


Figura 3.2: Casos de uso usuario/visualizador

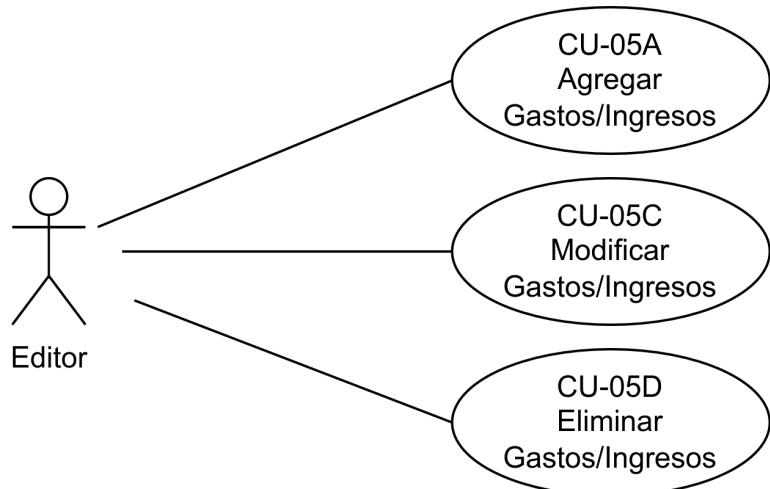


Figura 3.3: Casos de uso editor

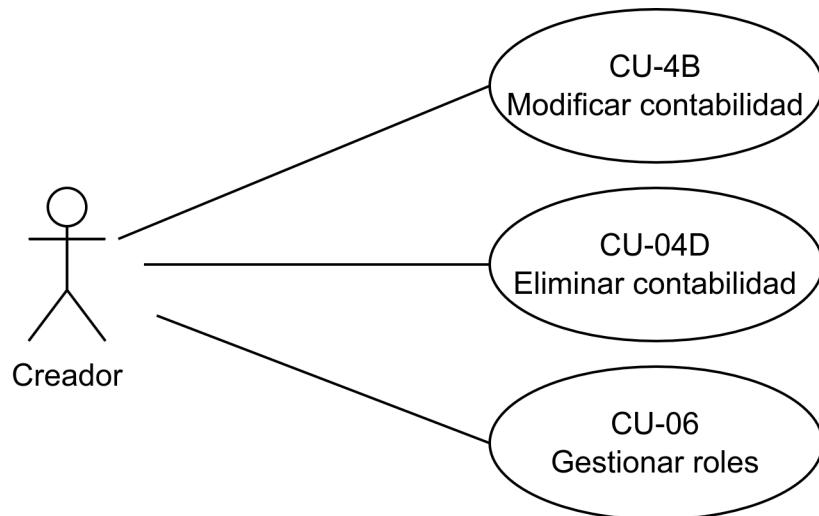


Figura 3.3: Casos de uso creador

3.2.3. Descripción de los casos de uso

CU-01	Registro de usuario
Requisito	RF-01
Precondiciones	
Postcondiciones	El usuario es registrado en el sistema.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona "Regístrate ahora" 2. El sistema presenta el formulario de registro 3. El usuario completa el formulario con su correo electrónico, contraseña y otros datos necesarios. 4. El usuario pulsa "Registrar" para proceder. 5. El sistema verifica la información, valida la unicidad del correo y los criterios de seguridad de la contraseña, y registra al usuario.
Escenarios alternativos	<p>5.a. Si el correo o el nombre de usuario ya existe o los datos no cumplen los criterios, el sistema muestra un mensaje de error y solicita la corrección</p> <p>5.b. Si el usuario decide no continuar con el registro y selecciona "Volver a atrás" el sistema lo redirecciará a la página principal</p>

Tabla 3.35: CU-01 (Registro de usuario)

CU-02	Inicio de sesión
Requisito	RF-02
Precondiciones	El usuario debe estar registrado.
Postcondiciones	El usuario accede a su cuenta.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Iniciar sesión". 2. El sistema presenta el formulario para iniciar sesión. 3. El usuario introduce su correo electrónico y contraseña registrados. 4. El usuario pulsa "Iniciar sesión" para acceder. 5. El sistema valida las credenciales y, si son correctas, autoriza el acceso y redirige al usuario a la página principal de su área personal.
Escenarios alternativos	5.a. Si las credenciales no son correctas (correo o contraseña erróneos), el sistema muestra un mensaje de error y retorna al paso 3.

Tabla 3.36: CU-02 (Inicio de sesión)

CU-03	Cerrar sesión
Requisito	RF-03
Precondiciones	El usuario debe estar logueado.
Postcondiciones	El usuario finaliza su sesión en la aplicación.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la pestaña de gestión de usuario. 2. El usuario selecciona "Cerrar sesión". 3. El sistema cierra la sesión y redirige al inicio de sesión.
Escenarios alternativos	

Tabla 3.37: CU-03 (Cerrar sesión)

CU-04A	Crear Contabilidad
Requisito	RF-06
Precondiciones	El usuario debe estar autenticado.
Postcondiciones	Contabilidad creada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de agregar una contabilidad. 2. El sistema presenta el formulario para agregar una contabilidad. 3. El usuario rellena el formulario y selecciona "Crear". 4. El sistema valida la información y crea la contabilidad.
Escenarios alternativos	4.a. Si los datos son inválidos, el sistema muestra un mensaje de error y solicita correcciones.

Tabla 3.38: CU-04A (Crear contabilidad)

CU-04B	Modificar Contabilidad
Requisito	RF-07
Precondiciones	El usuario debe estar autenticado; debe ser el propietario de la contabilidad. A su vez, la contabilidad a modificar debe existir.
Postcondiciones	Contabilidad modificada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de modificar una contabilidad. 2. El sistema presenta el formulario para modificar la contabilidad seleccionada. 3. El usuario realiza los cambios deseados y selecciona "Guardar cambios". 4. El sistema valida los cambios y actualiza la contabilidad.
Escenarios alternativos	4.a. Si los datos son inválidos, el sistema muestra un mensaje de error y solicita correcciones.

Tabla 3.39: CU-05B (Modificar contabilidad)

CU-04C	Visualizar Contabilidad
Requisito	RF-08
Precondiciones	El usuario debe estar autenticado. La contabilidad debe existir y además el usuario debe estar asignado a ésta.
Postcondiciones	Contabilidad visualizada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de visualizar una contabilidad. 2. El sistema muestra los detalles de la contabilidad seleccionada.
Escenarios alternativos	

Tabla 3.40: CU-04C (Visualizar contabilidad)

CU-04D	Eliminar Contabilidad
Requisito	RF-09
Precondiciones	El usuario debe ser el propietario de la contabilidad y estar autenticado. La contabilidad debe existir.
Postcondiciones	Contabilidad eliminada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de eliminar una contabilidad. 2. El sistema solicita confirmación para eliminar. 3. El usuario confirma la eliminación. 4. El sistema elimina la contabilidad y actualiza la base de datos.
Escenarios alternativos	3.a. Si el usuario cancela la acción, el sistema no realiza cambios.

Tabla 3.41: CU-04D (Eliminar contabilidad)

CU-05A	Agregar Gastos/Ingresos
Requisito	RF-10
Precondiciones	El usuario debe estar autenticado y tener permisos de editor.
Postcondiciones	Gasto/ingreso añadido.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de añadir un gasto/ingreso. 2. El sistema presenta el formulario correspondiente. 3. El usuario completa el formulario y selecciona "Añadir". 4. El sistema valida la información y añade el gasto/ingreso.
Escenarios alternativos	4.a. Datos incorrectos: El sistema informa y no procede con la acción.

Tabla 3.42: CU-05A (Agregar Gastos/Ingresos)

CU-05B	Visualizar Gastos/Ingresos
Requisito	RF-11
Precondiciones	El usuario debe estar autenticado y tener permisos de visualizador o superior.
Postcondiciones	Gasto/ingreso visualizado..
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de visualizar gastos/ingresos. 2. El sistema muestra una lista de gastos/ingresos con opciones para filtrar y ordenar.
Escenarios alternativos	

Tabla 3.43: CU-05B (Visualizar Gastos/Ingresos)

CU-05C	Modificar Gastos/Ingresos
Requisito	RF-12
Precondiciones	El usuario debe estar autenticado y tener permisos de editor.
Postcondiciones	Gasto/ingreso modificado.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de modificar un gasto/ingreso. 2. El sistema presenta el formulario con los datos actuales del gasto/ingreso. 3. El usuario modifica los datos necesarios y selecciona "Guardar cambios". 4. El sistema verifica los permisos, valida los datos y actualiza la información en la base de datos.
Escenarios alternativos	4.a. Datos incorrectos: El sistema informa y no procede con la acción.

Tabla 3.44: CU-05C (Modificar Gasto/Ingreso)

CU-05D	Eliminar Gastos/Ingresos
Requisito	RF-13
Precondiciones	El usuario debe estar autenticado y tener permisos de editor.
Postcondiciones	Gasto/ingreso eliminado.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de eliminar un gasto/ingreso. 2. El sistema solicita confirmación para eliminar. 3. El usuario confirma la eliminación. 4. El sistema elimina el gasto/ingreso y actualiza la base de datos.
Escenarios alternativos	3.a. Si el usuario cancela la acción, el sistema no realiza cambios.

Tabla 3.45: CU-05D (Eliminar Gastos/Ingresos)

CU-06	Gestionar Roles
Requisito	RF-14, RF-15, RF-16, RF-17
Precondiciones	El usuario debe ser el creador de la contabilidad y estar autenticado.
Postcondiciones	Roles de editor y visualizador asignados o revocados.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario accede a la gestión de roles dentro de una contabilidad. 2. El sistema muestra la lista de usuarios y sus roles actuales. 3. El usuario añade o elimina roles de editor o visualizador. 4. El sistema valida la acción y actualiza la configuración de la contabilidad.
Escenarios alternativos	

Tabla 3.46: CU-06 (Gestionar Roles)

CU-07	Visualizar estadísticas
Requisito	RF-18
Precondiciones	El usuario debe tener permisos de visualizador o superior y tener ingresos o gastos registrados.
Postcondiciones	Estadísticas y gráficos generados y visualizados.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona las diferentes pestañas que muestran estadísticas. 2. El sistema recopila los datos de gastos e ingresos. 3. El sistema presenta gráficos y resúmenes estadísticos.
Escenarios alternativos	<ol style="list-style-type: none"> 3.a. El sistema notifica que no hay gastos o ingresos disponibles para mostrar.

Tabla 3.47: CU-07 (Visualizar estadísticas)

3.3. Modelo conceptual de datos

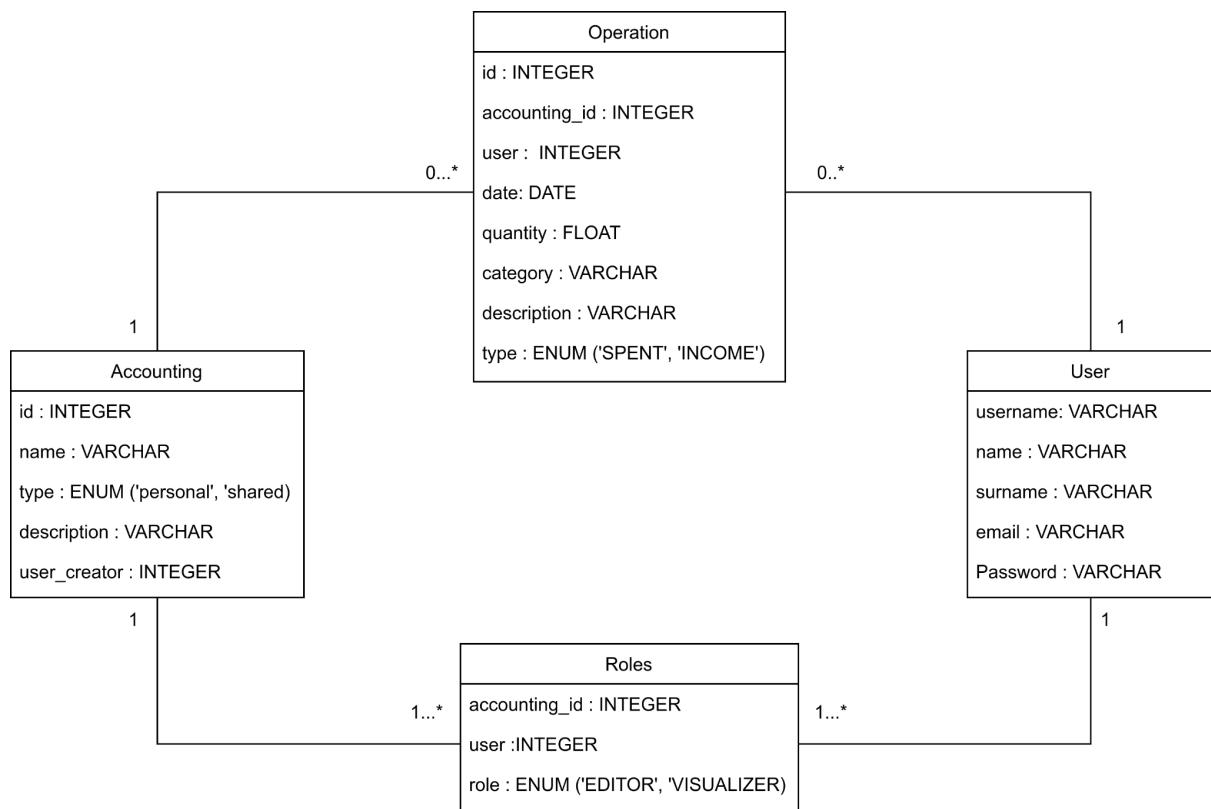


Figura 3.4: Modelo conceptual de datos

3.3.1. Entidades del modelo conceptual de datos

Las entidades del modelo, por tanto son las siguientes:

User (Usuario): Representa a las personas que interactúan con la aplicación. Un usuario puede crear y poseer múltiples contabilidades de tipo personal o compartido, lo cual es gestionado por la entidad Accounting. Además, los usuarios tienen roles específicos (visualizador o editor) dentro de las contabilidades compartidas, como se define en la entidad Roles. Los usuarios también pueden realizar operaciones (gastos o ingresos) asociadas a las contabilidades a las que tienen acceso, lo cual es manejado por la entidad Operations.

Accounting (Contabilidad): Esta entidad representa las contabilidades dentro del sistema. Cada contabilidad puede ser de tipo personal, usada exclusivamente por el creador, o compartida, donde múltiples usuarios pueden interactuar según los roles definidos en la entidad Roles. Las contabilidades registran diversas operaciones que indican los movimientos de dinero (ingresos y gastos), categorizados y datados en la entidad Operations.

Roles (Roles): Esta entidad define los roles de los usuarios dentro de las contabilidades compartidas. Un usuario puede tener el rol de visualizador, con permiso solo para ver la información y gastos, o editor, con capacidad para modificar los datos de la contabilidad. Esta relación garantiza que cada usuario tenga el nivel de acceso adecuado a las contabilidades a las que está vinculado.

Operations (Operaciones): Representa los movimientos financieros dentro de una contabilidad específica. Cada operación, ya sea un gasto o un ingreso, está vinculada a una contabilidad y a un usuario, indicando quién realizó la operación y a qué contabilidad pertenece. Las operaciones están clasificadas por categoría, cantidad, y llevan una descripción y una fecha, proporcionando un registro detallado de la actividad financiera en la contabilidad.

Capítulo 4: Diseño del sistema

En este capítulo se explica todo lo relacionado al diseño del sistema a implementar. Se abordan conceptos como las diferentes arquitecturas tanto lógica como física, además de todo el resto de diseño de los componentes que conforman la aplicación.

4.1. Arquitectura del sistema

En este apartado, se explicará la estructura integral del sistema, contemplando tanto su organización lógica como física. Se analizarán los componentes clave y su interconexión, así como los recursos tecnológicos necesarios para garantizar su funcionamiento eficiente y escalable.

4.1.1. Arquitectura física

En la siguiente sección, se detalla la infraestructura física del sistema, abordando la interacción entre los componentes hardware y software. Los usuarios de la plataforma podrán acceder a través de navegadores compatibles con JavaScript, ya sea desde dispositivos móviles u ordenadores. Para garantizar el funcionamiento óptimo, se establecerán requisitos específicos para los equipos que conforman la arquitectura física:

- Lado del servidor: Se requerirá un equipo con conexión a Internet y sistema operativo Windows, Linux o MacOS. Además, debe contar con Java 11 y JDK 19 o superior. El servidor web debe estar instalado, así como el servidor de la base de datos.
- Lado del cliente: Se necesitará un dispositivo con acceso a Internet y un navegador compatible con JavaScript.

4.1.2. Arquitectura lógica

El proyecto sigue una arquitectura con una clara separación entre el frontend y el backend. En este contexto, el backend se implementa como dos microservicios que exponen una API RESTful y el frontend se conecta a esta API para realizar sus operaciones.

El siguiente diagrama muestra la arquitectura lógica del sistema, incluyendo la comunicación entre el frontend, el backend y la base de datos:

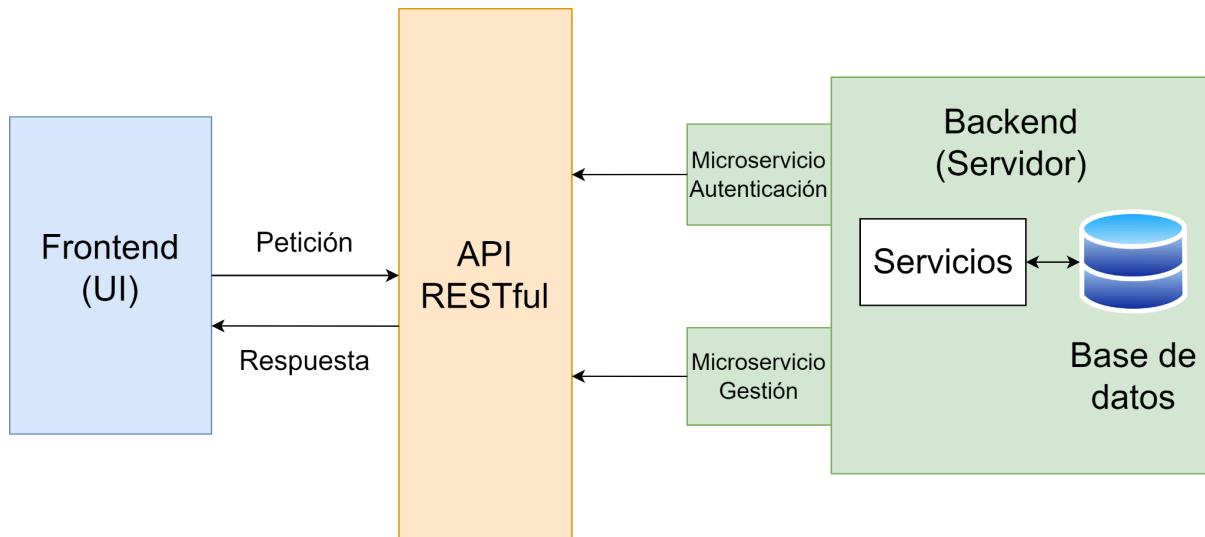


Figura 4.1: Arquitectura lógica

Frontend

La capa de interfaz de usuario está desarrollada en Vue.js y se conecta a la API RESTful expuesta por el backend. Esta capa es responsable de la presentación visual y la interacción del usuario. La aplicación Vue.js consume los servicios del backend para mostrar datos dinámicos y permitir que los usuarios realicen acciones como la gestión de finanzas personales y compartidas.

Backend

El backend está desarrollado utilizando Spring Boot y expone una API RESTful. El backend está compuesto de dos microservicios que se encargan de procesar las solicitudes, aplicar la lógica de negocio y acceder a la base de datos. A continuación se describen ambos microservicios y su funcionalidad:

- **Microservicio de Autenticación:** Responsable de la autenticación y autorización de usuarios. Implementa la lógica de inicio de sesión, verificación de credenciales y generación de tokens de seguridad.
- **Microservicio de Gestión:** Incluye varias funcionalidades, tales como:
 - Gestión de Registro de Usuarios: Maneja la creación de nuevos usuarios en el sistema, incluyendo la validación y almacenamiento de la información del usuario en la base de datos.
 - Gestión de Usuarios: Administra la información de los usuarios registrados, permitiendo actualizaciones, eliminaciones y consultas de los datos del usuario.

- Gestión de Contabilidades: Se encarga de todas las operaciones relacionadas con la gestión de las contabilidades personales y compartidas, como la creación, actualización y eliminación de contabilidades.
- Gestión de Operaciones: Maneja el registro y seguimiento de operaciones financieras (ingresos y gastos) dentro de las contabilidades. Incluye funciones para agregar, actualizar y eliminar operaciones.
- Gestión de Roles: Administra los roles de usuario dentro del sistema, asignando permisos específicos y gestionando la relación entre usuarios y roles.
- Gestión de DTOs (Data Transfer Objects): Facilita la transferencia de datos entre el microservicio y la interfaz de usuario, ayudando a reducir el número de llamadas y simplificar los objetos enviados o recibidos por la red.

La comunicación entre los componentes dentro del mismo microservicio se realiza mediante la inyección de dependencias, creando beans u objetos de servicio según sea necesario. Esto permite que los servicios llamen a métodos de otros servicios directamente, facilitando una integración más sencilla y rápida.

Base de Datos

El backend interactúa con una base de datos que maneja el almacenamiento, recuperación y manipulación de los datos. La base de datos asegura que cada funcionalidad del microservicio tenga acceso a la información necesaria para su funcionamiento.

4.2. Diseño físico de datos

La comunicación entre capas es posible gracias a la inyección de dependencias y además permite que las capas estén desacopladas, fomentando sencillez y claridad en las mismas. La detección de errores también se maneja y propaga a través de estas capas, dotando de robustez y mantenibilidad a la aplicación.

Para el diseño de la base de datos se ha realizado un diagrama entidad-relación (E-R) que se presenta a continuación. Este diagrama ilustra las principales entidades, relaciones y cardinalidades en el sistema, y sirve como base para la implementación del modelo de datos en el gestor de base de datos. En este caso, las principales entidades son User y Accounting, con sus respectivas relaciones que son dos: Operations y Roles.

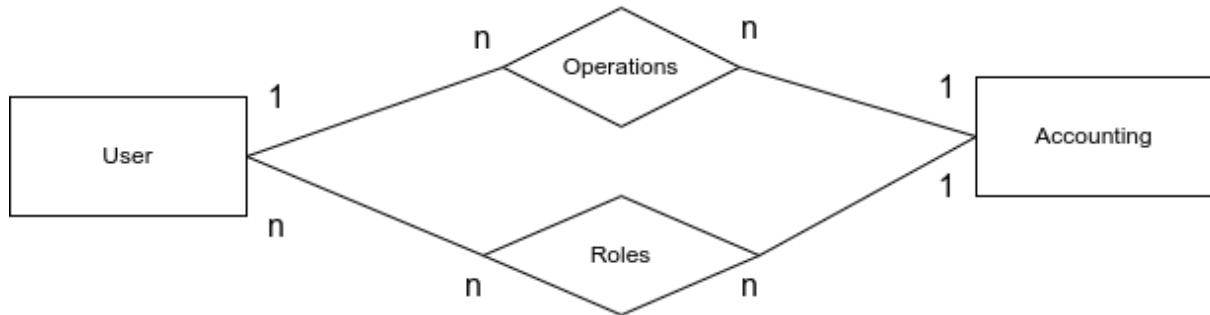


Figura 4.2: Diagrama entidad-relación

En cuanto a cardinalidades:

- **Operations:** un User puede realizar muchas Operations en una Accounting, y una Accounting puede tener muchas Operations realizadas por diferentes Users.
- **Roles:** un User puede tener múltiples roles en diferentes contabilidades, y una Accounting puede tener múltiples usuarios con distintos roles.

El paso a tablas del modelo entidad-relación, se traduce en las siguientes tablas:

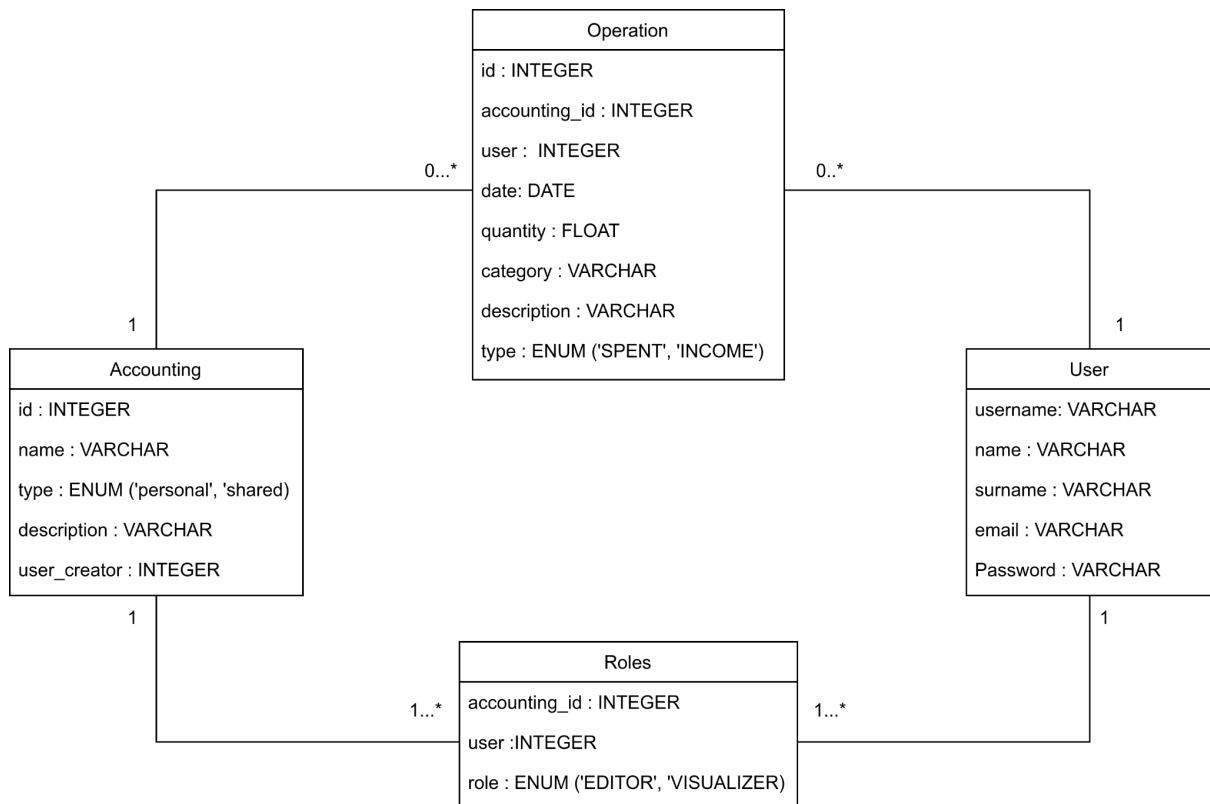


Figura 4.3: Paso a tablas

Tablas principales:

- User: Esta tabla contiene información sobre los usuarios del sistema y su clave primaria es “id”.
- Accounting: Esta tabla contiene información sobre las contabilidades gestionadas en el sistema y su clave primaria es “id”.

Tablas relacionales:

- Operations: Esta tabla registra las operaciones (gastos e ingresos) realizadas en las contabilidades. Su clave primaria es id y sus claves externas son user (referencia a User) y accounting_id (referencia a Accounting).
- Roles: Esta tabla asigna roles a los usuarios dentro de las contabilidades. Su clave primaria está compuesta por las dos claves externas que son user (referencia a User) y accounting_id (referencia a Accounting).

4.3. Patrones de diseño

Spring boot da muchas facilidades a la hora de implementar patrones de diseño. En concreto, para el proyecto “econoMe” se han usado los siguientes:

- **Microservicios [15]**: La arquitectura de microservicios se ha adoptado para dividir el sistema en servicios independientes que se comunican entre sí a través de interfaces bien definidas. Cada microservicio es responsable de una funcionalidad específica dentro del sistema, lo que permite una mayor flexibilidad, escalabilidad y facilidad de mantenimiento. Esta arquitectura mejora la capacidad de despliegue y la gestión de los servicios, permitiendo que cada uno pueda ser desarrollado, desplegado y escalado de manera independiente.
- **Repositorio [16]**: Encapsula la lógica para acceder a los datos almacenados en la base de datos. Los repositorios en Spring son interfaces que Spring automáticamente implementa en tiempo de ejecución.
- **Inyección de Dependencias (DI) [17]**: Spring gestiona las dependencias de las clases mediante la inyección de dependencias, utilizando anotaciones como @Autowired para injectar objetos automáticamente.
- **Singleton [18]**: Los beans de Spring Boot, como los servicios y controladores, generalmente se manejan como singletons, lo que significa que una única instancia de cada bean es creada y gestionada por el contenedor de Spring.
- **DTO (Data Transfer Object) [19]**: Utilizados para encapsular datos y enviarlos de un sistema a otro, ayudando a reducir el número de llamadas y simplificar los objetos enviados o recibidos por la red.

4.4. Diseño de los servicios

Es importante visualizar cómo se estructuran y relacionan las clases en el sistema a nivel de implementación. A continuación, se presenta el diagrama de clases, que detalla la arquitectura del sistema dividida en tres capas principales: Presentación, Lógica de Negocio y Acceso a Datos.

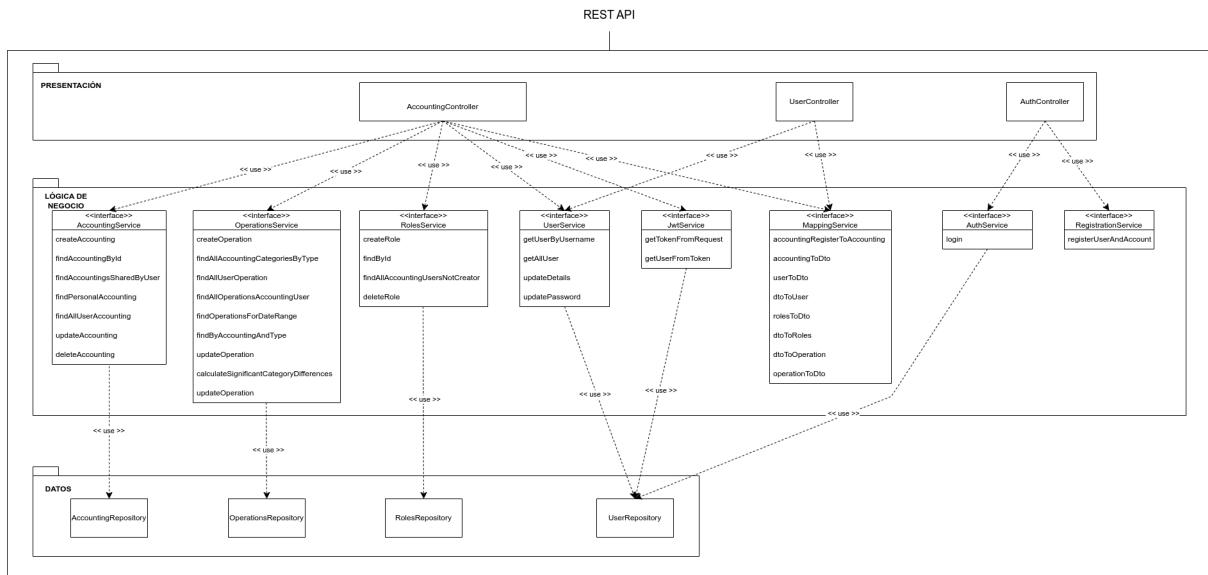


Figura 4.4: Diagrama de clases

En cuanto a la comunicación entre capas:

- Los controladores en la capa de presentación utilizan los servicios de la capa de lógica de negocio para procesar las solicitudes del cliente.
- Los servicios en la capa de lógica de negocio utilizan los repositorios de la capa de acceso a datos para realizar operaciones en la base de datos.
- Las interfaces (<<interface>>) en los servicios indican los métodos disponibles para la interacción entre las capas.

4.5. Diseño API Rest

Todos los controladores son autorizados por un JSON Web Token [\[20\]](#) para la identificación del usuario que es devuelto al realizar el login. Esto permite dotar de una mayor seguridad a la API y también de poder identificar al usuario en las peticiones sin pasar parámetros. La API se compone de tres controladores con sus respectivos endpoints que se detallan en las siguientes tablas:

- Accounting (contabilidad):

ID	Método	URI	Argumentos	Descripción	Salida
1	POST	/api/accounting/register	En formato JSON: Nombre, descripción, nombre de usuario creador y tipo de contabilidad (compartida o personal).	Crea una nueva contabilidad.	En formato JSON: los atributos pasados como argumentos y el id de la contabilidad.
2	PUT	/api/accounting/{id}	Id de contabilidad y en formato JSON: Nombre, descripción, nombre de usuario creador y tipo de contabilidad (compartida o personal).	Actualiza los datos de una contabilidad	En formato JSON: argumentos pasados y tipo de contabilidad.
3	GET	/api/accountingUserShared	Ninguno.	Devuelve los datos de las contabilidades del usuario que sean compartidas.	En formato JSON: id contabilidad, nombre, descripción, nombre de usuario del creador y tipo.
4	GET	/api/accounting/accountingPersonal	Ninguno.	Devuelve los datos de la contabilidad personal del usuario.	En formato JSON: id contabilidad, nombre, descripción, nombre de usuario del creador y tipo.

5	GET	/api/accounting/accountingUser	Ninguno.	Devuelve los datos de todas las contabilidades donde el usuario posea un rol.	En formato JSON: id contabilidad, nombre, descripción, nombre de usuario del creador y tipo.
6	GET	/api/accounting/{id}/rol	Id de contabilidad.	Devuelve el rol del usuario en la contabilidad con {id}.	En formato JSON: nombre de usuario, id de contabilidad y rol.
7	GET	/api/accounting/{id}/userCreator	Id de contabilidad.	Devuelve el creador de la contabilidad con {id}.	En formato JSON: nombre de usuario del creador.
8	GET	/api/accounting/{id}/categoriesSpent	Id de contabilidad.	Devuelve las categorías de gasto de la contabilidad {id}.	En formato JSON: lista de categorías de gasto.
9	GET	/api/accounting/{id}/categoriesIncome	Id de contabilidad.	Devuelve las categorías de ingreso de la contabilidad {id}.	En formato JSON: lista de categorías de ingreso.
10	POST	/api/accounting/operation/register	En formato JSON: atributos de operación (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario).	Registra una operación.	En formato JSON: mismos atributos que los recibidos.
11	PUT	/api/accounting/operation	En formato JSON: atributos de operación (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario).	Actualiza los detalles de una operación.	En formato JSON: mismos atributos que los recibidos.
12	GET	/api/accounting/{id}/operation/all	Id de	Devuelve todas	En formato

			contabilidad, tipo de filtro (7 días, 30 días o personalizado). En caso de personalizado fecha de inicio y fin.	las operaciones de la contabilidad {id}.	JSON: lista de operaciones (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
13	GET	/api/accounting/operation/all	Ninguno.	Devuelve todas las operaciones de todas las contabilidades creadas por el usuario.	En formato JSON: lista de operaciones (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
14	GET	/api/accounting/operationAccoun tings/all	Ninguno.	Devuelve todas las operaciones de todas las contabilidades del usuario.	En formato JSON: lista de operaciones (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
15	GET	/api/accounting/{id}/operation/sp ent	Id de contabilidad.	Devuelve los gastos de la contabilidad {id}.	En formato JSON: lista de gastos (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
16	GET	/api/accounting/{id}/operation/sp entFiltered	Id de contabilidad, tipo de filtro (7 días, 30 días o personalizado). En caso de personalizado fecha de inicio y fin.	Devuelve los gastos filtrados por intervalo de tiempo de la contabilidad {id}.	En formato JSON: lista de gastos (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)

17	GET	/api/accounting/{id}/operation/income	Id de contabilidad.	Devuelve los ingresos de la contabilidad {id}.	En formato JSON: lista de ingresos (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
18	GET	/api/accounting/{id}/operation/incomeFiltered	Id de contabilidad, tipo de filtro (7 días, 30 días o personalizado). En caso de personalizado fecha de inicio y fin.	Devuelve los ingresos filtrados por intervalo de tiempo de la contabilidad {id}.	En formato JSON: lista de gastos (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
19	GET	/api/accounting/operation/filterOperation	En formato JSON: id de contabilidad, tipo de operación, categoría, cantidad mínima y máxima y fecha de inicio y fin.	Devuelve las operaciones filtradas respecto a una serie de parámetros.	En formato JSON: lista de operaciones (tipo de operación, categoría, fecha, cantidad, descripción, id de contabilidad, nombre de usuario)
20	GET	/api/accounting/{id}/categoryDifferences	Id de contabilidad, tipo de filtro (7 días, 30 días o personalizado). En caso de personalizado fecha de inicio y fin.	Devuelve las diferencias más significativas entre categorías de ingresos y gastos de la contabilidad {id}.	En formato JSON: diferencias de ingresos y de gastos, tanto positivas como negativas.
21	POST	/api/accounting/{id}/addUser	Id de contabilidad, y en formato JSON: nombre de usuario, id de contabilidad y rol.	Añade usuario a la contabilidad {id}.	En formato JSON: nombre de usuario, id de contabilidad y rol.
22	GET	/api/accounting/{id}/users	Id de contabilidad.	Devuelve la lista de usuarios de la contabilidad	En formato JSON: lista de roles (nombre de

				{id}.	usuario, id de contabilidad y rol).
23	DELETE	/api/accounting/operation/{id}	Id de operación.	Elimina la operación {id}.	OK si se completa.
24	DELETE	/api/accounting/{id}/{username}	Id de contabilidad, nombre de usuario.	Elimina la contabilidad {id} del usuario {username} que es administrador.	OK si se completa.
25	DELETE	/api/accounting/{id}/deleteUser	Id de contabilidad, lista de usuarios en formato JSON (nombre de usuario).	Elimina un usuario de la contabilidad {id}.	OK si se completa.

Tabla 4.1: Endpoints de AccountingController

- Users (usuarios)

ID	Método	URI	Argumentos	Descripción	Salida
1	GET	/api/users/all	Ninguno.	Devuelve todos los usuarios del sistema.	Lista de usuarios en formato JSON (id de usuario, nombre de usuario, nombre, apellidos y correo)
2	GET	/api/users/details	Ninguno.	Devuelve los detalles del usuario asociado al token de la petición.	En formato JSON: nombre de usuario, nombre, apellidos y correo.
3	PUT	/api/users/modif yDetails	En formato JSON: nombre de usuario, nombre, apellidos y correo.	Modifica los detalles del usuario asociado al token de la petición.	En formato JSON: id de usuario, nombre de usuario, nombre, apellidos y correo.
4	PUT	/api/users/modif yPassword	En formato JSON: contraseña antigua y nueva contraseña.	Modifica la contraseña asociado al token de la petición.	OK si se actualiza correctamente.

5	DELETE	/api/users/delete	Ninguno.	Elimina el usuario asociado al token de la petición.	OK si se elimina.
---	--------	-------------------	----------	--	-------------------

Tabla 4.2: Endpoints de UsersController

- Auth (Autorización):

ID	Método	URI	Argumentos	Descripción	Salida
1	POST	/api/auth/register	En formato JSON: atributos de usuario (nombre de usuario, nombre.. etc) y atributos de contabilidad personal (nombre, descripción y tipo).	Registra un nuevo usuario.	Los mismos atributos que los argumentos recibidos en formato JSON.
2	POST	/api/auth/login	En formato JSON: nombre de usuario y contraseña.	Inicia sesión de un usuario proporcionando un token.	En formato JSON: token y nombre de usuario.

Tabla 4.3: Endpoints de AuthController

4.6. Diseño de la interfaz de usuario

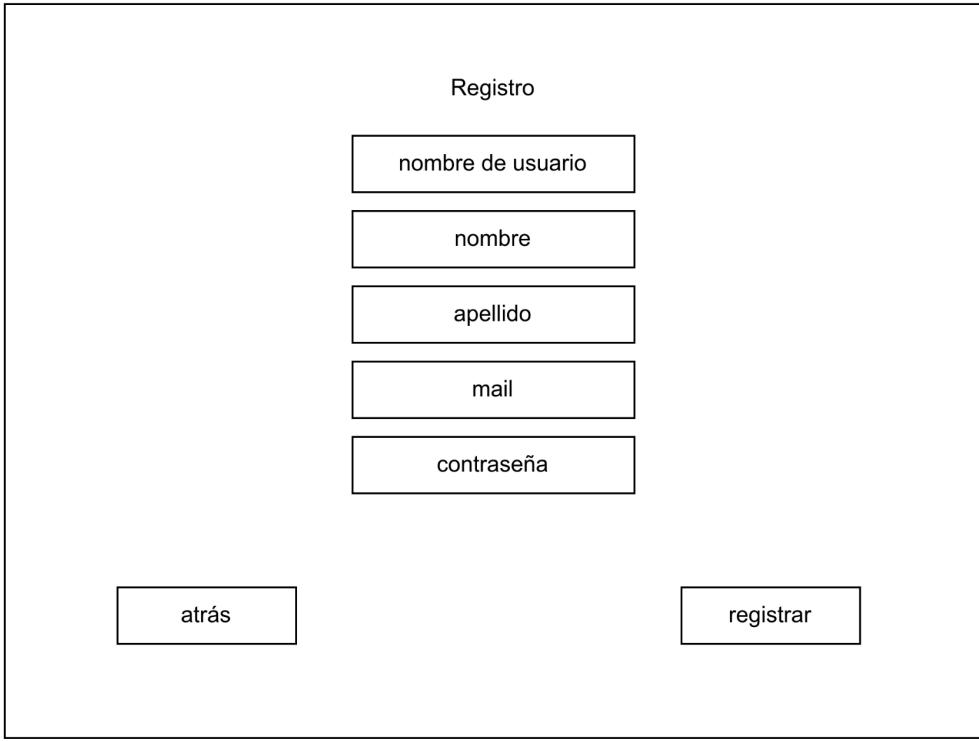
En esta sección se muestran los diseños iniciales de las diferentes vistas que componen la aplicación. Sobre cada una de ellas, se realizará una breve explicación.

4.6.1. Pantalla inicial



Figura 4.5: Diseño de pantalla inicial econoMe

Esta es la pantalla inicial con la que se encuentra el usuario al inicio de la aplicación. Aparece un formulario de inicio de sesión para los usuarios que posean cuenta y un enlace a un formulario de registro de usuario para aquellos que no dispongan de una.



Este diagrama ilustra el diseño de un formulario para el registro de usuarios. El formulario se encierra en un cuadro grande. Dentro de este cuadro, en la parte superior central, se encuentra el título "Registro". A continuación, se presentan cinco cuadros rectangulares uno sobre otro, cada uno destinado a un campo de texto: "nombre de usuario", "nombre", "apellido", "mail" y "contraseña". En la parte inferior izquierda del cuadro principal se sitúa un botón "atrás" y en la parte inferior derecha se sitúa un botón "registrar".

Figura 4.6: Diseño de formulario registro de usuario

Este es el formulario que se muestra para registrar un usuario. Los botones sirven para volver a la vista anterior y para mandar los datos y registrar el usuario si éstos son correctos.

4.6.2. Pantalla principal

Estos son los elementos comunes a gran parte de las vistas más usadas por el usuario como son: home, gastos, ingresos, evolución, operaciones y contabilidad compartida.

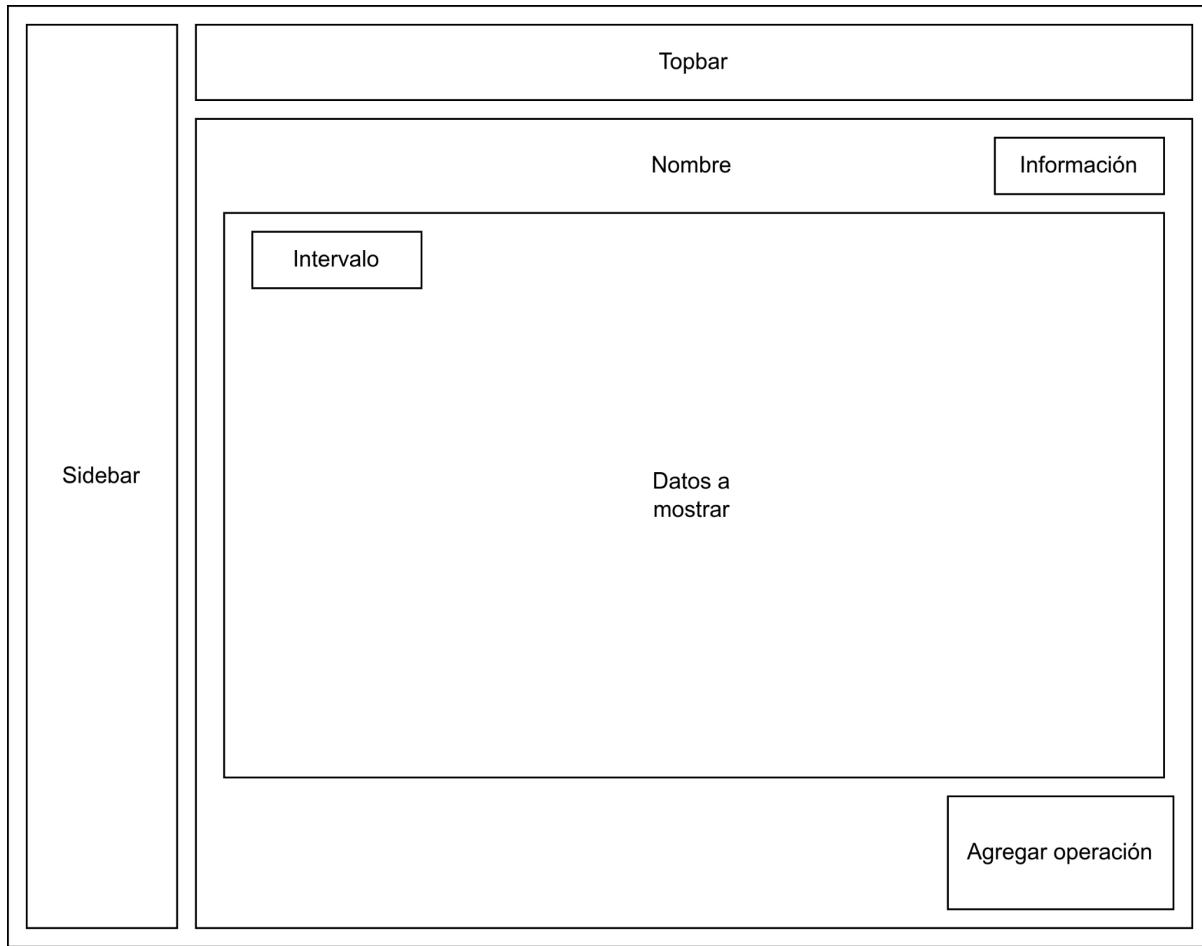


Figura 4.7: Diseño de pantalla principal

La pantalla principal está compuesta por los siguientes elementos:

- Sidebar: se navega a través de las diferentes vistas que he mostrado anteriormente.
- Topbar: se mostrará el nombre de la contabilidad actual así como un menú para poder acceder a la modificación de datos de usuario, contraseña, y cerrar sesión.
- Intervalo: es un componente interactivo para mostrar los datos en función de un intervalo de tiempo.
- Información: se mostrarán los detalles de la contabilidad.
- Agregar operación: se mostrará el formulario para poder agregar la operación. En el caso de contabilidades compartidas donde el usuario tenga rol de visualizador no aparecerá.
- Datos a mostrar: aquí se renderizan las vistas que se han mencionado anteriormente y que posteriormente se explicarán.

4.6.3. Vistas

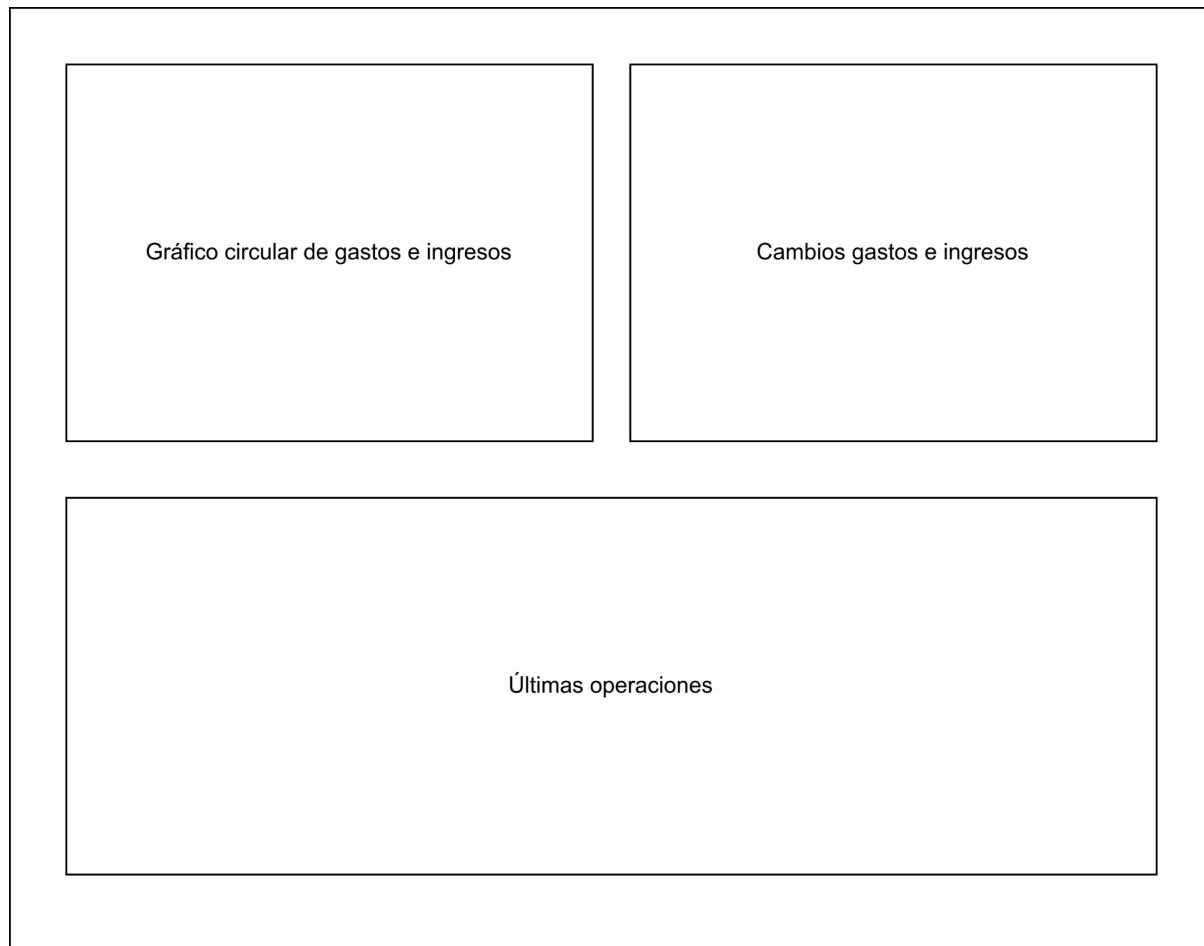


Figura 4.8: Diseño de home

Esta vista será la vista por defecto al iniciar sesión, donde se muestra un resumen de los datos de la contabilidad personal.

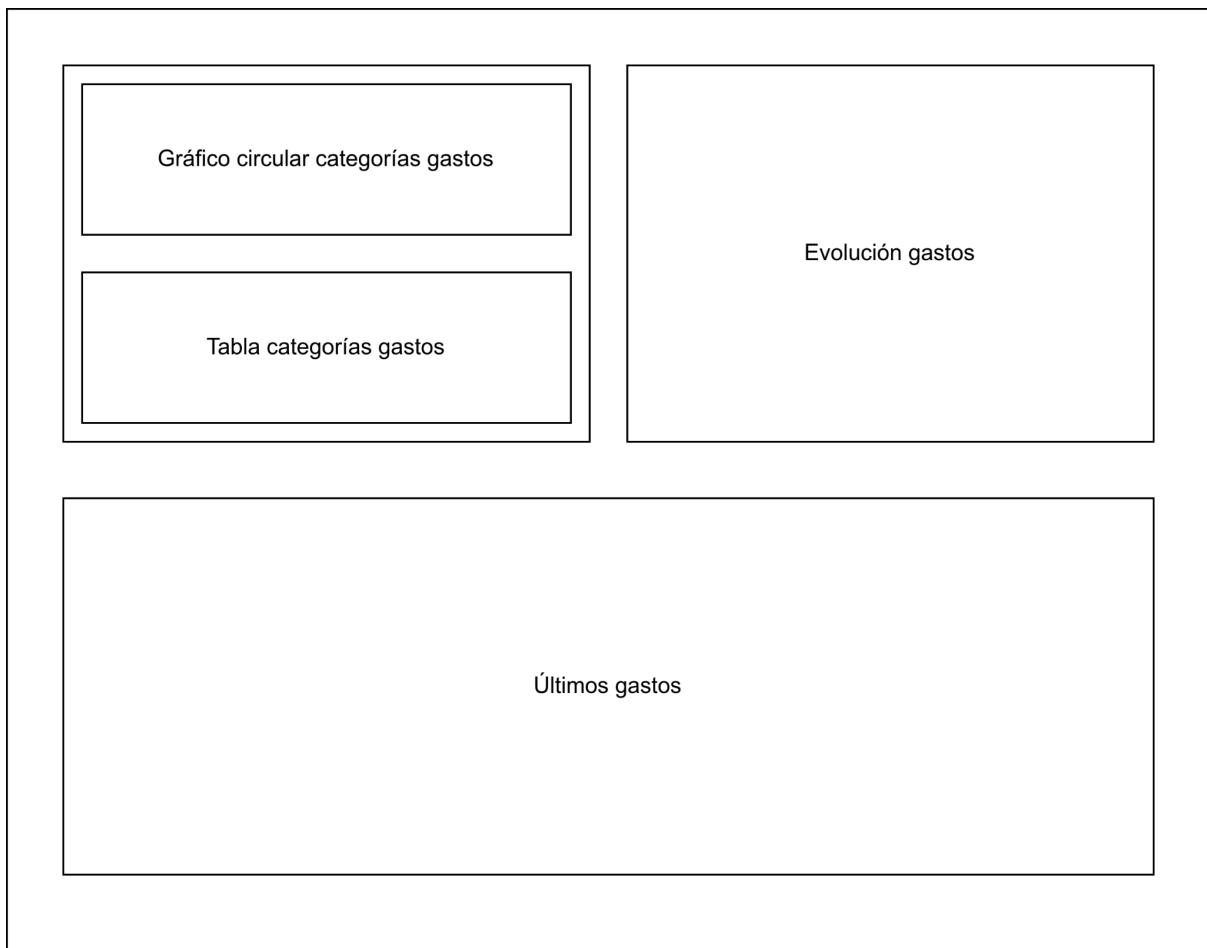


Figura 4.9: Diseño de gastos

Esta vista será para mostrar los gastos de la contabilidad personal.

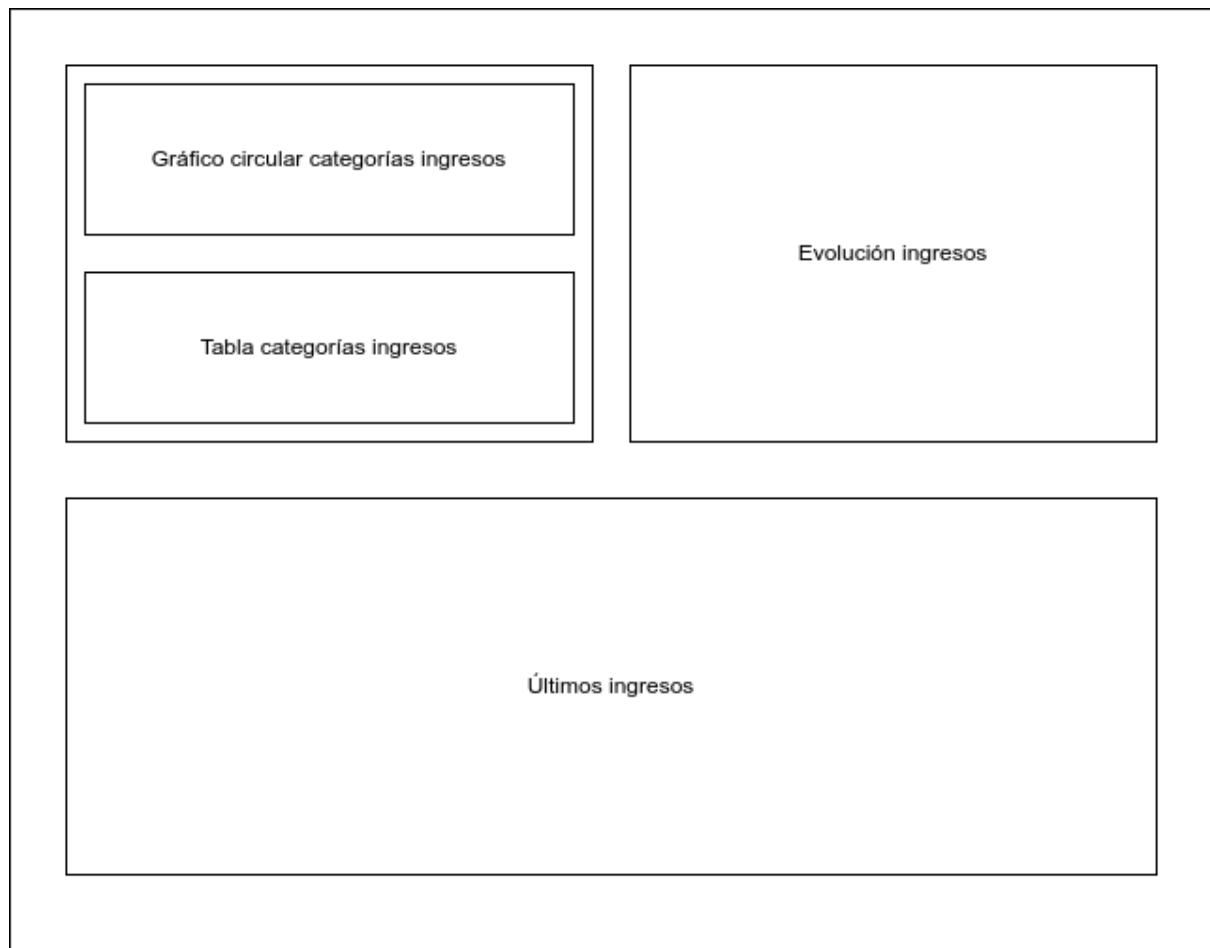


Figura 4.10: Diseño de ingresos

Esta vista será para mostrar los ingresos de la contabilidad personal.

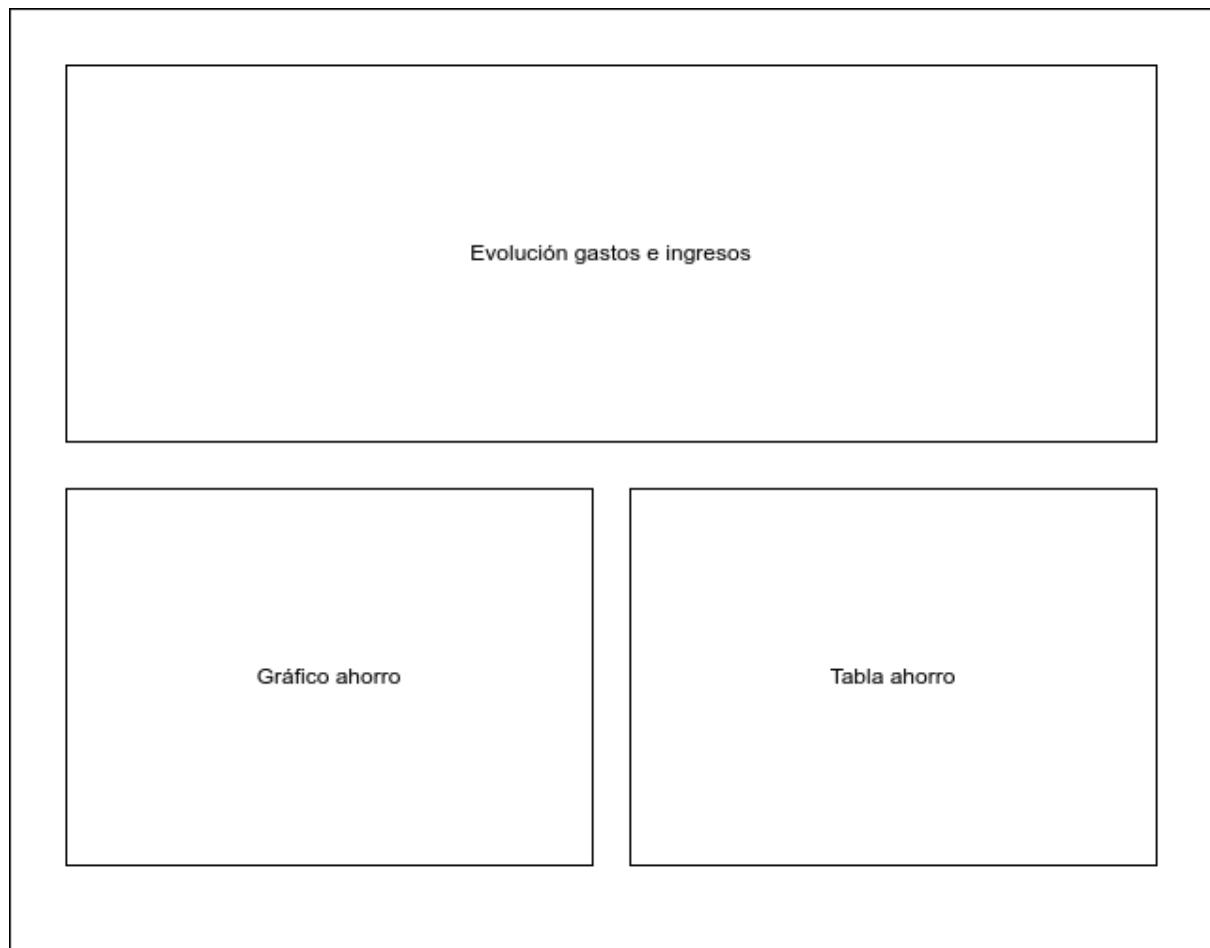


Figura 4.11: Diseño de evolución

Esta vista será para mostrar la evolución de los gastos e ingresos de la contabilidad personal, así como el ahorro.



Figura 4.12: Diseño de operaciones

Esta vista será para mostrar los gastos de las contabilidades a las que pertenece el usuario. Si posee un rol de editor se mostrarán los botones de editar y eliminar. En caso de ser visualizador, solo se mostrará el botón ver.



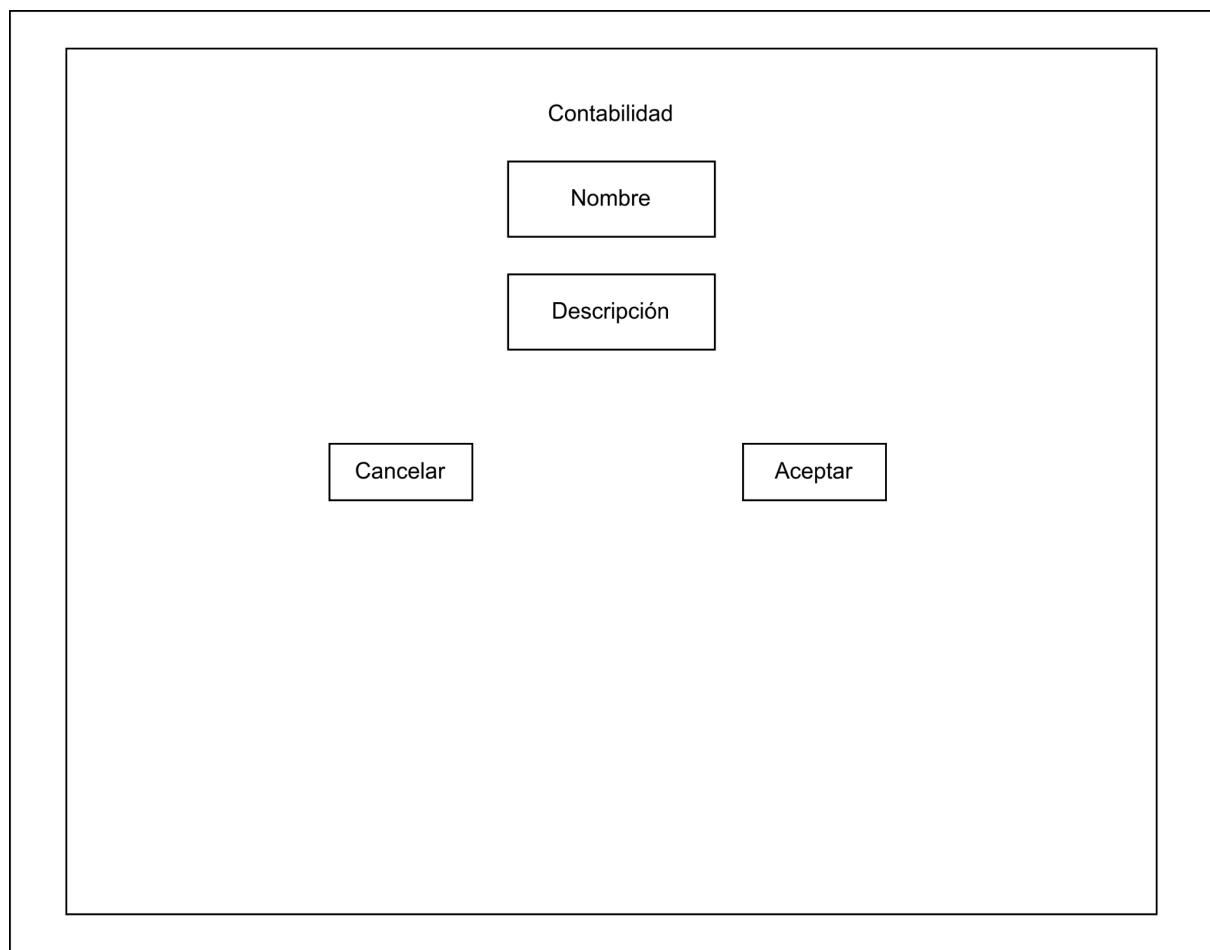
Figura 4.13: Diseño de contabilidades compartidas

Esta vista será para mostrar todos los datos de la contabilidad compartida seleccionada. En caso de ser administrador, se mostrarán los botones de gestionar usuarios y eliminar contabilidad.

The diagram illustrates the layout of a form titled 'Operación'. At the top center is the title 'Operación'. Below it are five input fields arranged vertically: 'Tipo', 'Categoría', 'Descripción', 'Fecha', and 'Cantidad'. At the bottom left is a button labeled 'Cancelar' and at the bottom right is a button labeled 'Aceptar'.

Figura 4.14: Diseño de formulario operación

Este formulario será el mostrado para agregar una operación a la contabilidad y además tendrá la misma estructura para modificar una operación pero con los datos ya cargados.



Contabilidad

Nombre

Descripción

Cancelar

Aceptar

The diagram illustrates a user interface for managing accounting entries. It features a large rectangular container labeled "Contabilidad". Inside, there are two input fields: one for "Nombre" (Name) and another for "Descripción" (Description). At the bottom left is a button labeled "Cancelar" (Cancel), and at the bottom right is a button labeled "Aceptar" (Accept).

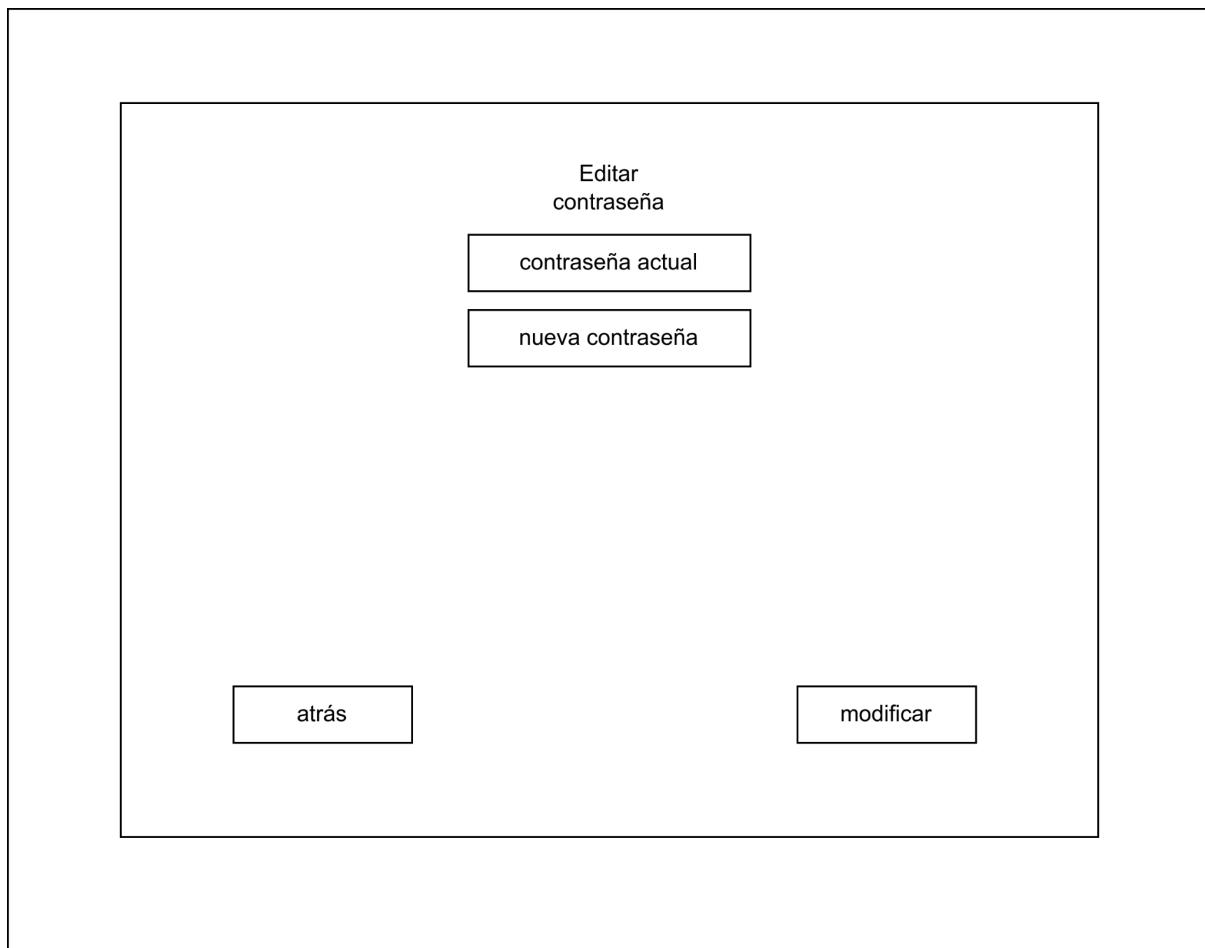
Figura 4.15: Diseño de formulario contabilidad

Este formulario será el mostrado para agregar una contabilidad y además tendrá la misma estructura para modificar una contabilidad pero con los datos ya cargados.

El diseño de formulario se encierra en un cuadro grande. Dentro de este cuadro, en la parte superior central, se encuentra el título "Editar usuario". A continuación, se presentan cuatro cuadros rectangulares uno sobre otro, cada uno con un campo de texto. Los campos están etiquetados como "nombre de usuario", "nombre", "apellido" y "mail". En la parte inferior izquierda del cuadro grande hay un botón "atrás" y en la parte inferior derecha hay un botón "modificar".

Figura 4.16: Diseño de formulario modificar detalles de usuario

Este formulario será el mostrado para modificar los detalles del usuario.



Este diagrama es un diseño de formulario para la modificación de contraseña. El formulario se encierra en un cuadro grande. Dentro de este cuadro, en la parte superior central, se encuentra el texto "Editar contraseña". Directamente debajo de él, hay dos cuadros uno sobre el otro, cada uno con un botón que dice "contraseña actual" y "nueva contraseña" respectivamente. En la parte inferior izquierda del cuadro grande hay un cuadro con el botón "atrás". En la parte inferior derecha hay otro cuadro con el botón "modificar".

Figura 4.17: Diseño de formulario modificar contraseña

Este formulario será el mostrado para modificar la contraseña del usuario.

Capítulo 5: Construcción del sistema

En este apartado se describirán y explicarán brevemente las herramientas usadas para la construcción del sistema.

5.1. Implementación

5.1.1. Java y Spring Boot

El backend de este sistema está desarrollado utilizando Java y Spring Boot [21], una combinación poderosa para la creación de aplicaciones robustas y escalables. Java, creado por Sun Microsystems en 1995, es uno de los lenguajes de programación más populares y ampliamente utilizados gracias a sus características clave:

- Independencia de Plataforma: Permite ejecutar el código en cualquier sistema operativo con JVM.
- Orientado a Objetos: Facilita la reutilización y modularidad del código.
- Seguridad: Proporciona un entorno seguro con gestión automática de memoria y manejo de excepciones.
- Multithreading: Soporta la ejecución de múltiples tareas simultáneamente.

Para optimizar el desarrollo en Java, se utiliza Spring Boot, un marco que simplifica la creación de aplicaciones de producción. Sus principales ventajas incluyen:

- Configuración Automática: Reduce significativamente el tiempo de configuración inicial.
- Microservicios: Facilita la creación y gestión de arquitecturas de microservicios.
- Seguridad Integrada: Proporciona herramientas y configuraciones para implementar medidas de seguridad fácilmente.
- Ecosistema Extenso: Ofrece un amplio conjunto de bibliotecas y herramientas adicionales.

La combinación de Java y Spring Boot ofrece un entorno de desarrollo eficiente y flexible, ideal para construir aplicaciones backend robustas y escalables.

5.1.2. Vue

Vue.js [22] es un framework progresivo de JavaScript creado por Evan You en 2014. Ideal para construir interfaces de usuario y aplicaciones de una sola página (SPA), destaca por su capacidad de actualizar automáticamente la interfaz cuando cambian los datos gracias a su sistema de reactividad.

Su arquitectura basada en componentes permite crear aplicaciones modulares y reutilizables, mientras que su simplicidad y documentación clara facilitan el aprendizaje y

uso. Vue.js también maneja la vinculación del DOM con los datos del modelo de manera eficiente y clara mediante directivas y plantillas declarativas.

El ecosistema de Vue.js incluye herramientas como Vue CLI, Vue Router y Vuex, que facilitan la configuración, el manejo de rutas y la gestión del estado. Además, Vue.js está optimizado para ser ligero y rápido, ofreciendo un rendimiento superior en navegadores modernos. Todo esto lo convierte en una excelente opción para desarrollar aplicaciones web modernas de manera eficiente y sencilla.

5.1.3. MySQL

MySQL [\[23\]](#) es un sistema de gestión de bases de datos relacional de código abierto, lanzado en 1995. SQL es utilizado para gestionar y manipular datos, destacándose por su rapidez, fiabilidad y facilidad de uso.

Ideal para aplicaciones web, MySQL es utilizado por plataformas como WordPress y Facebook. Su arquitectura cliente-servidor permite acceso simultáneo por múltiples usuarios, asegurando la integridad de los datos mediante transacciones ACID y soporte de claves foráneas.

MySQL tiene una fuerte comunidad de usuarios y desarrolladores, con amplia documentación y soporte. Su integración con tecnologías como Java lo hace una opción robusta y flexible para la gestión de bases de datos.

5.2. Entorno de desarrollo

5.2.1. IntelliJ IDEA

IntelliJ IDEA [\[6\]](#), desarrollado por JetBrains, es un entorno de desarrollo integrado (IDE) ideal para el desarrollo en Java y otros lenguajes. Destaca por su asistencia de codificación inteligente, ofreciendo autocompletado avanzado, refactorización y detección de errores en tiempo real.

Cuenta con herramientas integradas como control de versiones (Git, SVN), construcción de proyectos (Maven, Gradle) y pruebas (JUnit), todo en una interfaz intuitiva y personalizable. Sus potentes capacidades de depuración y análisis facilitan la identificación y resolución de problemas.

IntelliJ IDEA es una opción robusta y eficiente para manejar proyectos complejos, mejorando la productividad y haciendo el desarrollo más fluido.

5.3. Herramientas adicionales

5.3.1. Google Drive

Google Drive [\[9\]](#) es un servicio de almacenamiento en la nube que permite guardar, sincronizar y compartir archivos desde cualquier dispositivo con internet. Es ideal para gestionar documentos y asegurar acceso a la versión más reciente en todo momento. Además tiene integrado Google Docs que es una herramienta de procesamiento de texto. Permite crear, editar y colaborar en documentos en línea. Los usuarios pueden trabajar simultáneamente en el mismo documento, ver cambios en tiempo real y utilizar comentarios y sugerencias para comunicarse. Los documentos se guardan automáticamente y son accesibles desde cualquier dispositivo.

Ambas herramientas juntas proporcionan un entorno eficiente y flexible para la redacción y almacenamiento de documentos, siendo esenciales para la creación y gestión de la memoria del proyecto.

5.3.2. Draw.io

Draw.io [\[10\]](#) es una herramienta en línea para crear y editar diagramas de manera sencilla y eficiente. Ideal para diagramas de flujo, organigramas y más, ofrece una interfaz intuitiva de arrastrar y soltar, permitiendo a los usuarios crear diagramas rápidamente.

Facilita la colaboración en tiempo real, integrándose con servicios como Google Drive, Dropbox y OneDrive, para guardar y acceder a tus diagramas desde cualquier lugar. Con una amplia gama de plantillas y formas prediseñadas, draw.io se adapta a diversas necesidades de diagramación.

Los diagramas pueden exportarse en varios formatos, como PNG, JPEG y PDF, facilitando su uso en documentos y presentaciones. En resumen, draw.io es una herramienta práctica y colaborativa para visualizar y comunicar ideas efectivamente.

5.3.3. Github

GitHub [\[8\]](#) es una plataforma de desarrollo colaborativo que utiliza Git para el control de versiones. Permite a los desarrolladores almacenar y gestionar código, colaborando en proyectos de manera eficiente. Facilita el trabajo en equipo mediante pull requests y issues, y soporta tanto repositorios públicos como privados.

Integrado con herramientas de desarrollo y CI/CD, GitHub mejora el flujo de trabajo. Además, ofrece funcionalidades para documentar proyectos y mantener wikis, haciendo que la gestión de proyectos de software sea más organizada y colaborativa.

5.3.4. Postman

Postman [\[7\]](#) es una herramienta utilizada para desarrollar, probar y documentar APIs. Facilita la creación y ejecución de solicitudes HTTP, permitiendo a los desarrolladores verificar y depurar las respuestas de sus APIs.

Postman ofrece una interfaz intuitiva para construir y organizar solicitudes, además de funcionalidades avanzadas como pruebas automatizadas, documentación de APIs y colaboración en equipo. Es esencial para asegurar el correcto funcionamiento y la calidad de las APIs en el desarrollo de software.

Capítulo 6: Pruebas del sistema

En este apartado se describirán brevemente los utensilios y metodologías empleadas para realizar las pruebas del sistema. Se detallarán los tipos de pruebas y las herramientas utilizadas para garantizar la calidad y el correcto funcionamiento del sistema.

6.1. Estrategia

La estrategia escogida para probar el sistema se ha basado en la combinación de pruebas automáticas y pruebas manuales.

6.2. Entorno de pruebas

Para la realización de las pruebas se ha empleado un equipo con las siguientes características:

- Modelo: MSI GF62-8RD
 - Procesador: Intel Core i7-8750H
 - Memoria RAM: 16GB
 - Tarjeta Gráfica: NVIDIA GeForce GTX 1050Ti
 - Sistema operativo: Ubuntu 22.04.4 LTS

6.3. Pruebas del sistema

En esta sección se muestran las pruebas realizadas para comprobar los requisitos del sistema descritos en el capítulo 3.

6.3.1. Pruebas funcionales

Las pruebas funcionales como su propio nombre indica evaluarán todos los requisitos funcionales de la aplicación.

PF-01	Prueba registro de usuario
Descripción	Se comprobará que el usuario sea añadido al sistema y quede registrado correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Nombre de usuario: jesus • Nombre: Jesús • Apellidos: Leyva Fernández • Correo electrónico: jesusleyva@gmail.com • Contraseña: 123456
Salida esperada	El usuario se registra correctamente y queda guardado en la base de datos.

Tabla 6.1: PF-01 (Prueba registro de usuario)

PF-02	Prueba inicio de sesión
Descripción	Se comprobará que el usuario pueda iniciar sesión en el sistema correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Nombre de usuario: jesus • Contraseña: 123456
Salida esperada	El usuario inicia sesión, obteniendo un token con validez de una hora.

Tabla 6.2: PF-02 (Prueba inicio de sesión)

PF-03	Prueba cierre de sesión
Descripción	Se comprobará que cierre sesión en el sistema correctamente.
Datos de entrada	No aplica
Salida esperada	El usuario cierra sesión y es redirigido a la página principal.

Tabla 6.3: PF-03 (Prueba cierre de sesión)

PF-04	Prueba consultar datos de usuario
Descripción	Se comprobará que los datos del usuario se muestran correctamente.
Datos de entrada	No aplica
Salida esperada	Los datos del usuario se muestran y coinciden con los registrados en la base de datos.

Tabla 6.4: PF-04 (Prueba consultar datos de usuario)

PF-05	Prueba modificar datos de usuario
Descripción	Se comprobará que los datos del usuario se modifican correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Nombre de usuario: jesusNuevo • Nombre: JesúsNuevo • Apellidos: Leyva FernándezNuevo • Correo electrónico: jesusleyvaNuevo@gmail.com
Salida esperada	Los datos del usuario se modifican y se reflejan en la base de datos.

Tabla 6.5: PF-05 (Prueba modificar datos de usuario)

PF-06	Prueba modificar contraseña de usuario
Descripción	Se comprobará que la contraseña del usuario se modifica correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Contraseña: 654321
Salida esperada	La contraseña del usuario se modifica y se refleja en la base de datos.

Tabla 6.6: PF-06 (Prueba modificar contraseña de usuario)

PF-07	Prueba agregar contabilidad
Descripción	Se comprobará que la contabilidad quede registrada en el sistema.
Datos de entrada	<ul style="list-style-type: none"> • Nombre de la contabilidad: contabilidad • Descripción: descripción
Salida esperada	Los datos de la contabilidad se agregan y se reflejan en la base de datos.

Tabla 6.7: PF-07 (Prueba consultar datos de contabilidad)

PF-08	Prueba modificar datos de contabilidad
Descripción	Se comprobará que los datos de la contabilidad se modifican correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Nombre de la contabilidad: nueva • Descripción: nueva descripción
Salida esperada	Los datos de la contabilidad se modifican y se reflejan en la base de datos.

Tabla 6.8: PF-08 (Prueba modificar datos de contabilidad)

PF-09	Prueba visualizar una contabilidad
Descripción	Se comprobará que los datos de la contabilidad se muestran correctamente.
Datos de entrada	No aplica
Salida esperada	Los datos de la contabilidad se muestran y coinciden con los registrados en la base de datos.

Tabla 6.9: PF-09 (Prueba visualizar una contabilidad)

PF-10	Eliminar contabilidad compartida
Descripción	Se comprobará que la contabilidad compartida se elimina correctamente.
Datos de entrada	No aplica
Salida esperada	La contabilidad compartida se elimina y se refleja en la base de datos.

Tabla 6.10: PF-10 (Eliminar contabilidad compartida)

PF-11	Prueba agregar gasto/ingreso a una contabilidad
Descripción	Se comprobará que la operación se registra en el sistema.
Datos de entrada	<ul style="list-style-type: none"> • Tipo: Ingreso • Categoría: Salario • Descripción: - • Fecha: 23/05/2024 • Cantidad: 2000,80
Salida esperada	Los datos de la operación se registran y se asocian a la contabilidad.

Tabla 6.11: PF-11 (Prueba agregar gasto/ingreso a una contabilidad)

PF-12	Prueba visualizar gasto/ingreso de una contabilidad
Descripción	Se comprobará que los datos de la operación se muestran correctamente.
Datos de entrada	No aplica
Salida esperada	Los datos de la operación se muestran y coinciden con los registrados en la base de datos.

Tabla 6.12: PF-12 (Prueba visualizar gasto/ingreso a una contabilidad)

PF-13	Prueba editar gasto/ingreso de una contabilidad
Descripción	Se comprobará que los datos de la operación cambian correctamente.
Datos de entrada	<ul style="list-style-type: none"> • Tipo: Ingreso • Categoría: Salario • Descripción: descripción agregada • Fecha: 23/05/2024 • Cantidad: 1000,80
Salida esperada	Los datos de la operación cambian y se reflejan en la propia aplicación.

Tabla 6.13: PF-13 (Prueba editar gasto/ingreso de una contabilidad)

PF-14	Prueba eliminar gasto/ingreso de una contabilidad
Descripción	Se comprobará que la operación se elimina correctamente.
Datos de entrada	No aplica
Salida esperada	Los datos de la operación se eliminan y desaparecen del sistema.

Tabla 6.14: PF-14 (Prueba eliminar gasto/ingreso de una contabilidad)

PF-15	Prueba filtrar operaciones
Descripción	Se comprobará que las operaciones mostradas se ajusten a los filtros usados.
Datos de entrada	<ul style="list-style-type: none"> • Contabilidad: Contabilidad Personal • Operación: Gasto • Categoría: Transporte • Cantidad mínima: 0 • Cantidad máxima: 1000 • Fecha inicio: 01/04/2024 • Fecha fin: 31/05/2024
Salida esperada	Las operaciones se muestran y coinciden con los filtros introducidos.

Tabla 6.15: PF-15 (Prueba filtrar operaciones)

PF-16	Agregar usuario editor a una contabilidad compartida
Descripción	Se comprobará que el usuario quede registrado en la contabilidad compartida con permisos de editor.
Datos de entrada	<ul style="list-style-type: none"> • Usuario: editor • Rol: Editor
Salida esperada	El usuario editor queda añadido a la contabilidad compartida con permisos de editor y se refleja en la base de datos.

Tabla 6.16: PF-16 (Agregar usuario editor a una contabilidad compartida)

PF-17	Eliminar usuario editor de una contabilidad compartida
Descripción	Se comprobará que el usuario editor quede eliminado de la contabilidad compartida.
Datos de entrada	<ul style="list-style-type: none"> • Usuario: editor
Salida esperada	El usuario editor queda eliminado de la contabilidad compartida y se refleja en la base de datos.

Tabla 6.17: PF-17 (Eliminar usuario editor de una contabilidad compartida)

PF-18	Agregar usuario visualizador a una contabilidad compartida
Descripción	Se comprobará que el usuario visualizador quede registrado en la contabilidad compartida.
Datos de entrada	<ul style="list-style-type: none"> • Usuario: visualizador • Rol: Visualizador
Salida esperada	El usuario visualizador queda añadido a la contabilidad compartida con permisos de visualizador y se refleja en la base de datos.

Tabla 6.18: PF-18 (Agregar usuario visualizador a una contabilidad compartida)

PF-19	Eliminar usuario visualizador de una contabilidad compartida
Descripción	Se comprobará que el usuario visualizador quede eliminado de la contabilidad compartida.
Datos de entrada	<ul style="list-style-type: none"> • Usuario: visualizador
Salida esperada	El usuario visualizador queda eliminado de la contabilidad compartida y se refleja en la base de datos.

Tabla 6.19: PF-19 (Eliminar usuario visualizador de una contabilidad compartida)

PF-20	Visualizar estadísticas
Descripción	Se comprobará que las estadísticas mostradas se ajusten a los filtros usados y sean consecuentes con los datos introducidos por el usuario.
Datos de entrada	No aplica
Salida esperada	Las estadísticas se muestran y coinciden con los datos que existen en la base de datos.

Tabla 6.20: PF-20 (Visualizar estadísticas)

6.3.2. Pruebas no funcionales

Los requisitos no funcionales han sido probados a través de numerosas pruebas manuales y objetivas para garantizar el cumplimiento de los mismos. A continuación se describen las pruebas usadas para cada uno:

Seguridad

- Prueba de autenticación segura:
Para asegurar que solo los usuarios registrados puedan acceder al sistema, se llevaron a cabo pruebas de inicio de sesión. Se utilizaron credenciales válidas e inválidas para verificar que el acceso se otorgará únicamente a los usuarios autenticados correctamente. Los resultados confirmaron que solo las credenciales correctas permitían el acceso, mientras que los intentos con credenciales incorrectas fueron rechazados con mensajes de error claros y apropiados.
- Prueba de controles de acceso por rol:
Se evaluaron los controles de acceso basados en los roles de usuario dentro de las contabilidades compartidas. Se asignaron roles específicos como administrador, editor y visualizador a diferentes usuarios y se probó el acceso a diversas funcionalidades. Los resultados mostraron que los permisos se aplicaban

correctamente: los administradores podían gestionar todas las funciones, los editores podían añadir y modificar datos, y los visualizadores tenían acceso solo a la visualización de información.

- **Prueba de cifrado de datos sensibles:**

Para garantizar la confidencialidad de los datos de los usuarios, se revisó el almacenamiento de contraseñas en la base de datos. Se comprobó que las contraseñas estaban cifradas utilizando algoritmos seguros. Esta inspección confirmó que las contraseñas no se almacenaban en texto claro, asegurando así la protección de la información sensible de los usuarios.

Usabilidad

Para verificar la claridad e intuitividad de la interfaz, se realizaron sesiones de prueba con usuarios de diferentes niveles técnicos. Estos usuarios interactuaron con diversas funciones de la aplicación, desde el registro hasta la gestión de contabilidades. Se recopilaron observaciones y se realizaron encuestas posteriores para evaluar la experiencia del usuario. Los resultados indicaron que la mayoría de los usuarios encontraron la interfaz fácil de usar y navegar, con una curva de aprendizaje rápida debido a la disposición lógica y consistente de los elementos de interfaz.

Rendimiento

Para garantizar que el sistema respondiera rápidamente a las solicitudes de los usuarios, se realizaron pruebas de tiempo de respuesta en diversos escenarios. Se simularon acciones comunes como inicio de sesión, consulta de datos, y operaciones de ingreso y gasto, midiendo el tiempo que el sistema tardaba en responder. Además se realizaron varias consultas simultáneamente y en un breve lapso de tiempo para medir cómo respondía el sistema ante una considerable carga de trabajo.

Las pruebas de rendimiento confirmaron que el sistema cumple con los requisitos establecidos. El tiempo de respuesta del sistema es rápido y consistente, proporcionando una experiencia de usuario ágil. Además, el sistema demuestra una capacidad robusta para manejar grandes volúmenes de usuarios y operaciones sin comprometer el rendimiento, asegurando así su fiabilidad y eficiencia en escenarios de alta carga.

Mantenibilidad

Para verificar la mantenibilidad del sistema, se realizó una revisión exhaustiva de la estructura del código y la documentación. En cuanto a la modularidad, se evaluó la organización del código en módulos lógicos, comprobando que cada módulo tiene una responsabilidad clara y que las dependencias entre ellos son mínimas y bien gestionadas. Esto permitió confirmar que las actualizaciones y mantenimientos pueden realizarse de manera eficiente sin afectar a otros módulos.

En relación a la documentación, se revisó el código fuente para asegurar que estuviera bien documentado, incluyendo comentarios explicativos y documentación de funciones y métodos.

Portabilidad

Para asegurar la portabilidad del sistema, se llevaron a cabo pruebas exhaustivas en diversas plataformas y navegadores. Se probó la aplicación en los principales sistemas operativos, incluyendo Windows y Linux. Además, se verificó su funcionamiento en los navegadores más utilizados, como Google Chrome, Mozilla Firefox, Safari, Microsoft Edge y Opera.

Durante las pruebas, se evaluó el comportamiento de todas las funcionalidades principales de la aplicación, asegurando que la interfaz de usuario se renderizara correctamente y que las interacciones fueran consistentes en todas las plataformas.

Capítulo 7: Manual de instalación e implementación

En este capítulo se detallan todos los requisitos necesarios para la instalación, así como los pasos a seguir por el usuario.

7.1. Requisito previos

Para el adecuado funcionamiento y despliegue del sistema es necesario que se cumplan una serie de requisitos previos. A continuación se detallan y enumeran los mismos:

Requisitos hardware:

- Un ordenador con acceso a internet.
- Un mínimo de 4GB de memoria RAM.

Requisitos de software:

- Java Development Kit (JDK): Versión 17 o superior (Necesario para compilar y ejecutar la aplicación de Spring Boot).
- Maven: Versión 3.6.0 o superior (Utilizado para gestionar dependencias y construir el proyecto del backend).
- Node.js: Versión 19 o superior (Necesario para construir la aplicación Vue).
- MySQL: Versión 8 o superior (Servidor de base de datos para almacenar los datos de la aplicación).
- Sistema operativo: Windows, Linux o MacOS.
- Servidor web: Nginx o Apache (Utilizado para servir los archivos estáticos del frontend).

7.2. Inventario de componentes

El sistema consta elementalmente de 2 elementos:

- La aplicación servidor de econoMe.
- La aplicación cliente de econoMe.

7.3. Procedimiento de instalación

Paso a paso detallado para la instalación y configuración del sistema. Este proceso presupone la instalación de los elementos detallados en la sección de requisitos previos:

1. **Clonación del repositorio del proyecto:** Clonar el repositorio del backend y del frontend desde el sistema de control de versiones. El enlace del mismo es:
<https://github.com/leyva7/econoMe>

2. **Configuración del Backend:** Navegar al directorio del backend y ejecutar:

- mvn clean install
- java -jar target/gestor-gastos-back-0.0.1-SNAPSHOT.jar

El backend se ejecutará en el puerto 8081.

3. **Configuración de la base de datos:**

- Crear la base de datos y los usuarios necesarios. Puede hacerlo utilizando los siguientes comandos en MySQL:

```
CREATE DATABASE econome;
CREATE USER 'econome_user'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON econome.* TO 'econome_user'@'localhost';
FLUSH PRIVILEGES;
```

En caso de usar un usuario o contraseña diferente debe actualizar la información en application.properties.

4. **Configuración del Frontend:** Navegar al directorio del frontend y ejecutar:

- npm install
- npm run build

Subir la carpeta dist generada al servidor web. La aplicación frontend funcionará en el puerto 8080.

Notas adicionales:

- Asegurar que MySQL está instalado y configurado según los requisitos previos.
- Verificar que los puertos 8080 y 8081 estén disponibles y no bloqueados por el firewall o en uso por otros servicios.
- Consultar la documentación específica del servidor web (Nginx, Apache, etc.) para configurar la carpeta dist correctamente.

Capítulo 8: Manual de usuario

Este capítulo proporciona una guía completa para el usuario final del sistema, detallando las características principales, los requisitos previos y las instrucciones para el uso del sistema. Este manual está diseñado para ayudar a los usuarios a entender y utilizar el gestor de gastos e ingresos personales y grupales de manera efectiva.

8.1. Características principales

El sistema de gestor de gastos e ingresos personales y grupales ofrece las siguientes características principales:

- Creación y edición de una cuenta de usuario en la aplicación, así como la contabilidad personal.
- Creación, edición y eliminación de contabilidades compartidas.
- Agregar y eliminar usuarios de contabilidades compartidas si se es el creador de la contabilidad.
- Agregar, editar y eliminar un gasto o ingreso en una contabilidad del usuario, siempre que se tenga un rol de editor en caso de ser compartida.
- Visualización de estadísticas por intervalos de tiempo de los gastos e ingresos de una contabilidad concreta.

8.2. Requisitos previos

Los requisitos previos para poder usar la aplicación son:

- Un ordenador con acceso a un navegador web y conexión a internet.
- Haber completado la instalación detallada en el capítulo anterior.

8.3. Uso del sistema

En este apartado se explican una a una las funcionalidades del sistema. Además se muestran imágenes para ilustrar estas funcionalidades.

8.3.1. Registro e inicio de sesión en la aplicación

Al lanzar la aplicación, se muestra la pantalla principal, donde los usuarios pueden registrarse o iniciar sesión.

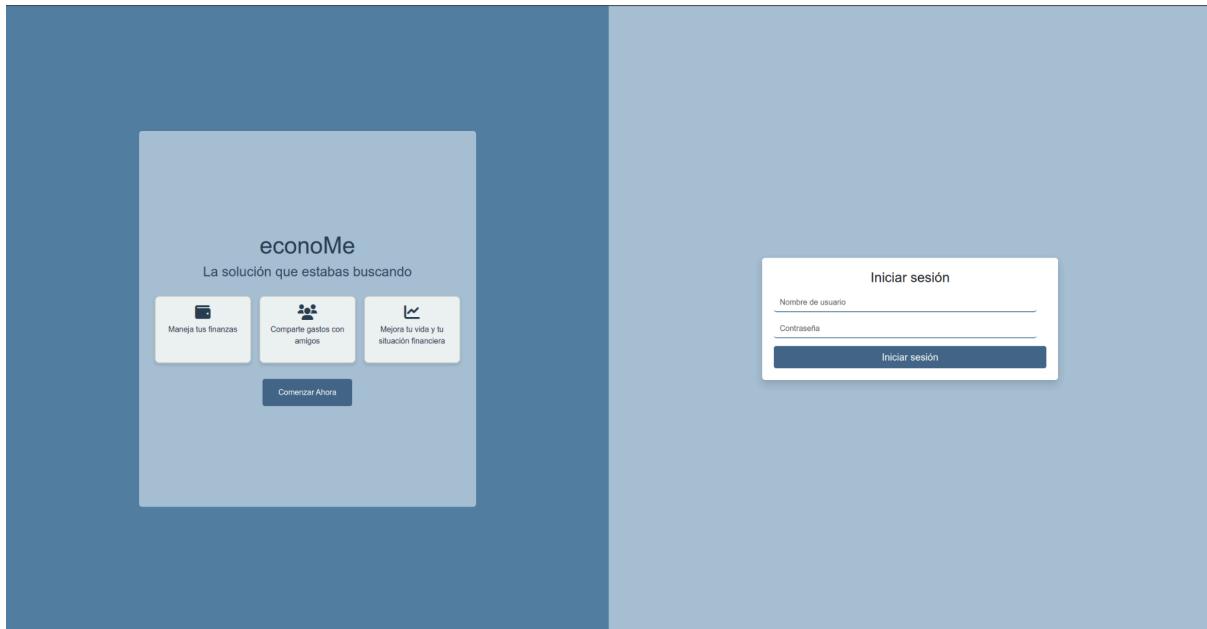


Figura 8.1: Pantalla principal

Para registrarse, haga clic en "Comenzar ahora". Esto lo redirigirá a una nueva vista para registrar sus datos. Complete todos los campos del formulario y haga clic en "Enviar". Una vez registrado, se creará su contabilidad personal y será redirigido a la página principal para iniciar sesión.

Para iniciar sesión, complete el formulario de la derecha en la página principal con sus credenciales y haga clic en "Iniciar Sesión". Esto lo llevará a la sección "Home", donde se muestra un resumen de su contabilidad personal y las últimas operaciones.

The screenshot shows a user registration form titled "Registro de usuario" (User Registration) within a dark-themed application. The form fields are as follows:

- Nombre de usuario: jesus_leyva
- Nombre: Jesús
- Apellidos: Leyva Fernández
- Correo Electrónico: jesusleyvafer@correo.ugr.es
- Contraseña: (represented by a series of dots)

At the bottom right of the form are two buttons: "Volver atrás" (Back) and "Registrar" (Register).

Figura 8.2: Registro de usuario

Para iniciar sesión se deben llenar con las credenciales el formulario que aparece a la derecha en la página principal.

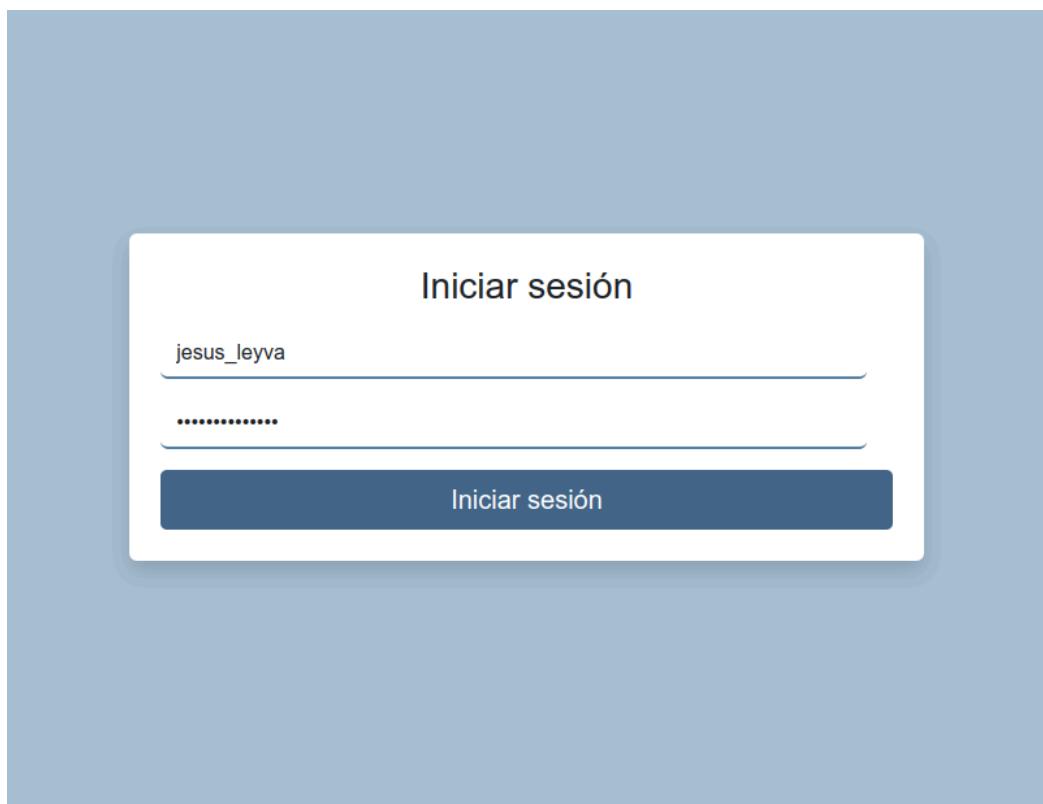


Figura 8.3: Inicio de sesión

Una vez que iniciamos sesión en la aplicación se nos redirigirá a la página por defecto que es la sección “Home” donde se muestra un pequeño resumen con algunas métricas de la contabilidad personal, así como las últimas operaciones de esta contabilidad.

A screenshot of the econoMe application's home screen. The top navigation bar is dark blue with the text "Contabilidad personal" and a user icon labeled "jesus_leyva". On the left is a sidebar with the "econoMe" logo and menu items: "Home", "Gastos", "Ingresos", "Evolución", and "Operaciones". Below this is a section for "Contabilidades Compartidas" with a "Añadir Contabilidad" button. The main content area has a light gray background. It features a "Resumen" section with a dropdown menu set to "Últimos 7 días". Inside are three boxes: "Gastos VS Ingresos" (No hay datos disponibles), "Gastos Totales: No hay datos disponibles", and "Ingresos Totales: No hay datos disponibles". Below this is a section for "Últimas operaciones" with the message "No hay operaciones recientes disponibles." A small "+" button is located in the bottom right corner of the screen.

Figura 8.4: Página home por defecto

8.3.2. Área de usuario

En la parte superior derecha de la pantalla, se muestra su nombre de usuario con una pequeña flecha. Al hacer clic en ella, se desplegará un menú con las siguientes opciones:

- Datos de usuario: Redirige a un formulario similar al de registro, pero sin el campo de contraseña. Los datos se muestran de forma predeterminada sin posibilidad de edición. Haga clic en "Editar" para desbloquear los campos y permitir la escritura.
- Modificar contraseña: Redirige a un formulario donde se pide la contraseña actual y la nueva.
- Cerrar sesión: Cierra la sesión del usuario y lo redirige a la página principal.



Figura 8.5: Área de usuario

The screenshot shows a dark-themed user interface for 'econoMe'. At the top, the brand name 'econoMe' is displayed in white. Below it, a modal window titled 'Datos usuario' (User Data) is open. Inside the modal, there are four input fields with placeholder text: 'Nombre de usuario' (username1), 'Nombre' (username1), 'Apellidos' (surname1), and 'Correo Electrónico' (username1@username.com). At the bottom right of the modal are two buttons: 'Editar' (Edit) and 'Cerrar' (Close).

Figura 8.6: Datos usuario

The screenshot shows a dark-themed user interface for 'econoMe'. At the top, the brand name 'econoMe' is displayed in white. Below it, a modal window titled 'Cambiar Contraseña' (Change Password) is open. Inside the modal, there are two input fields: 'Contraseña actual' (Current Password) and 'Contraseña nueva' (New Password). At the bottom right of the modal are two buttons: 'Cambiar contraseña' (Change Password) and 'Volver atrás' (Go Back).

Figura 8.7: Cambiar contraseña

8.3.3. Barra lateral de navegación

A la izquierda de la vista principal, siempre aparece la barra lateral de navegación. Esta barra permite desplazarse por las diferentes secciones de la aplicación:

- Home: Muestra un resumen de la contabilidad personal.
- Gastos: Muestra los gastos de la contabilidad personal del usuario.
- Ingresos: Muestra los ingresos de la contabilidad personal del usuario.
- Evolución: Muestra la evolución de los gastos e ingresos de la contabilidad personal del usuario.
- Operaciones: Permite localizar, visualizar, editar y eliminar operaciones tanto personales como compartidas.
- Contabilidades compartidas: Permite gestionar las contabilidades compartidas y navegar por ellas.

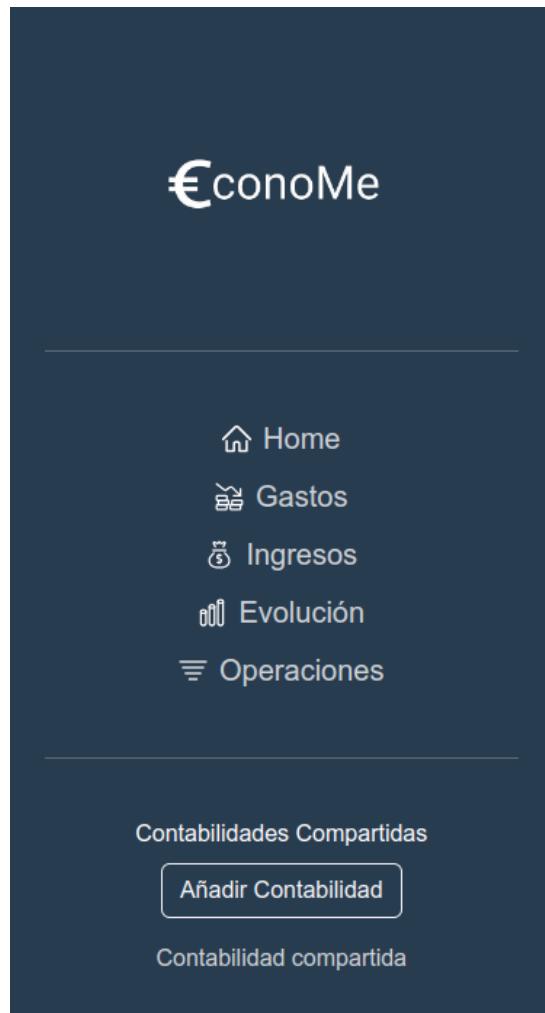


Figura 8.8: Barra lateral de navegación

8.3.4. Vistas

En cada vista, hay un botón de información en la esquina superior derecha que muestra los detalles de la contabilidad en la sección actual. Al hacer clic en él, se abre un modal que permite visualizar y modificar estos detalles.

En la esquina inferior derecha, hay un botón para añadir operaciones, visible en todas las secciones excepto en "Operaciones". Para las contabilidades compartidas, este botón se muestra dependiendo del rol del usuario (visualizador o editor).

Cada vista también incluye un desplegable para filtrar los datos y estadísticas por intervalos de tiempo.

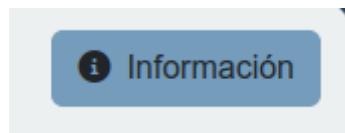


Figura 8.9: Botón de información de contabilidad

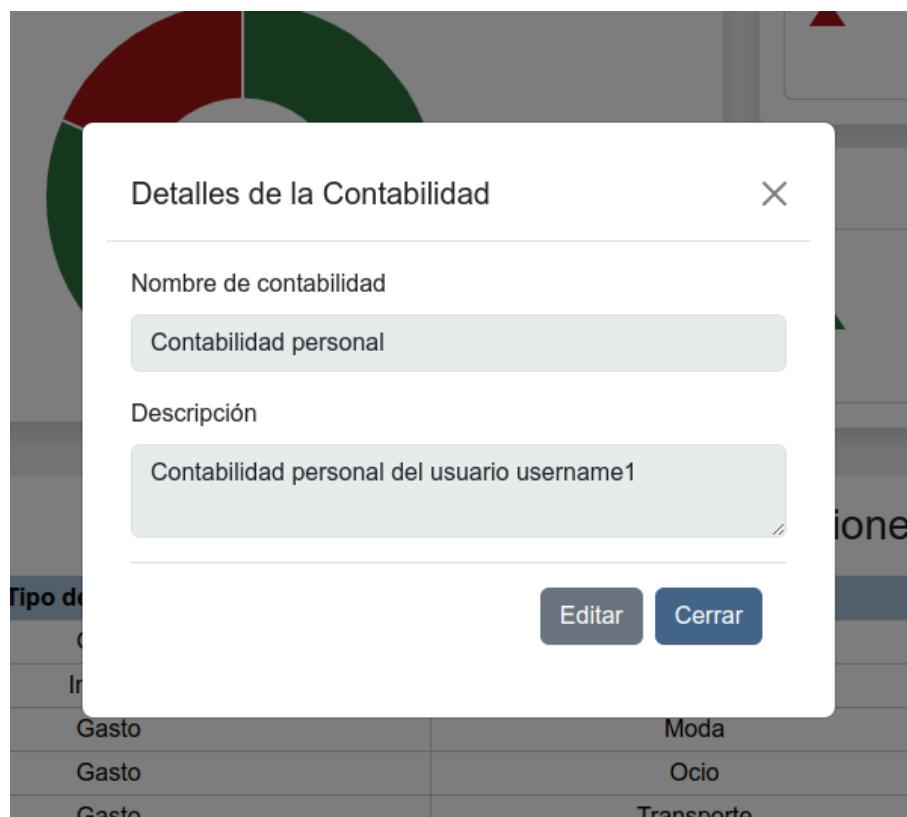


Figura 8.10: Modal de información de contabilidad

Abajo a la derecha de cada vista se muestra un botón para añadir operaciones (esto no es así para la sección operaciones). Para la contabilidad personal siempre aparece, pero para

las contabilidades compartidas este botón se muestra dependiendo de su rol en la misma (visualizador o editor).

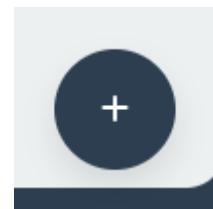


Figura 8.11: Botón para añadir operación

Tras pulsar este botón, de forma similar a los detalles de la contabilidad se muestra un modal para añadir la operación con distintos campos.

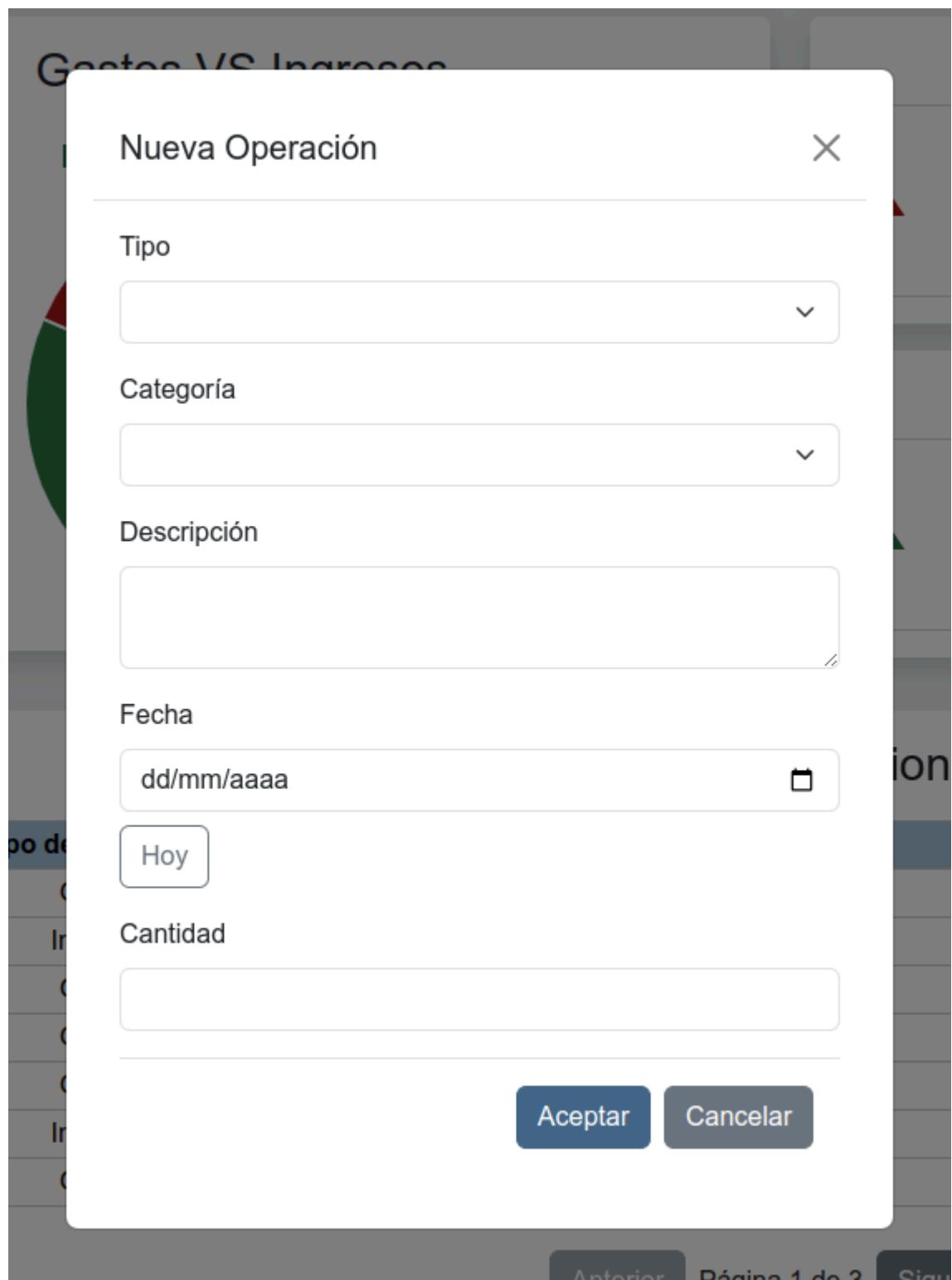


Figura 8.12: Modal para añadir operación

Además para cada una de las vistas (no es así para la sección operaciones) se muestra un desplegable para filtrar los datos y estadísticas por tiempo.

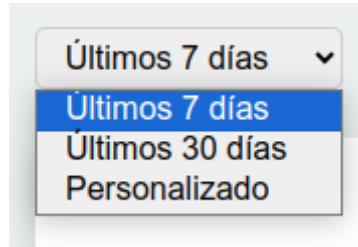


Figura 8.13: Intervalo para filtrar datos y estadísticas

8.3.5. Home

La vista Home muestra un resumen de la contabilidad personal y diferentes estadísticas de la misma. En ella figuran tres secciones que será la tónica habitual para el resto de ellos en términos de estructura. Todas las secciones se muestran en función del intervalo seleccionado:

- Gráfico Gastos VS Ingresos: muestra una comparación de los gastos y los ingresos.
- Cambios de tendencia y gastos e ingresos totales: a la derecha del gráfico se muestran los gastos e ingresos totales. Además, se presentan los cambios de tendencia de mayor magnitud tanto positivos como negativos en gastos e ingresos.
- Tabla de últimas operaciones: debajo de las secciones anteriores una tabla para representar las últimas operaciones.

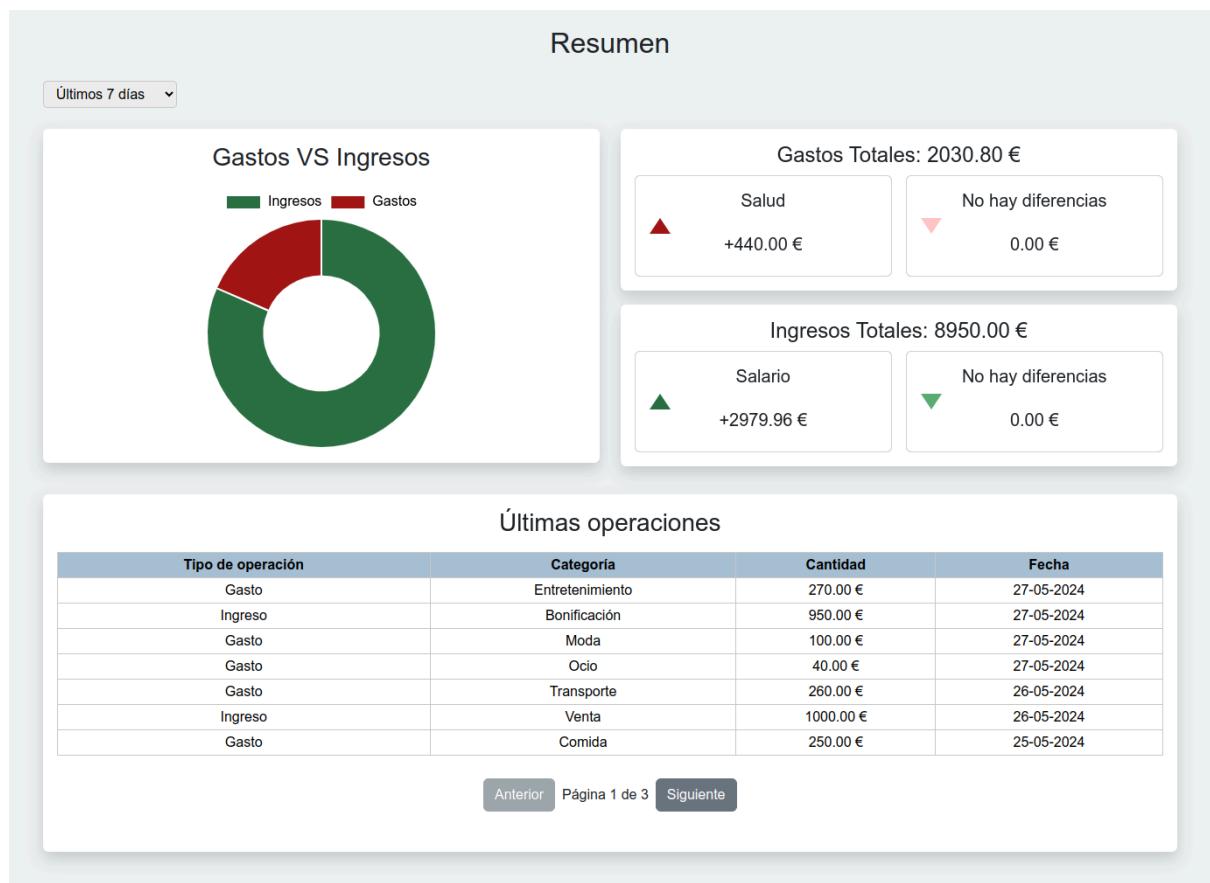


Figura 8.14: Vista Home

8.3.6. Gastos

Esta vista representa los gastos del usuario en la contabilidad personal. Se muestra el gasto total para el intervalo de tiempo seleccionado. Se presentan en la parte superior dos gráficos y en la inferior una tabla:

- En el primer gráfico se representan las categorías más frecuentes, así como el total de las mismas.
- En el segundo gráfico, se dispone la tendencia de gastos en el intervalo de tiempo.
- La tabla posee los últimos de gastos de la misma forma que en la sección anterior.

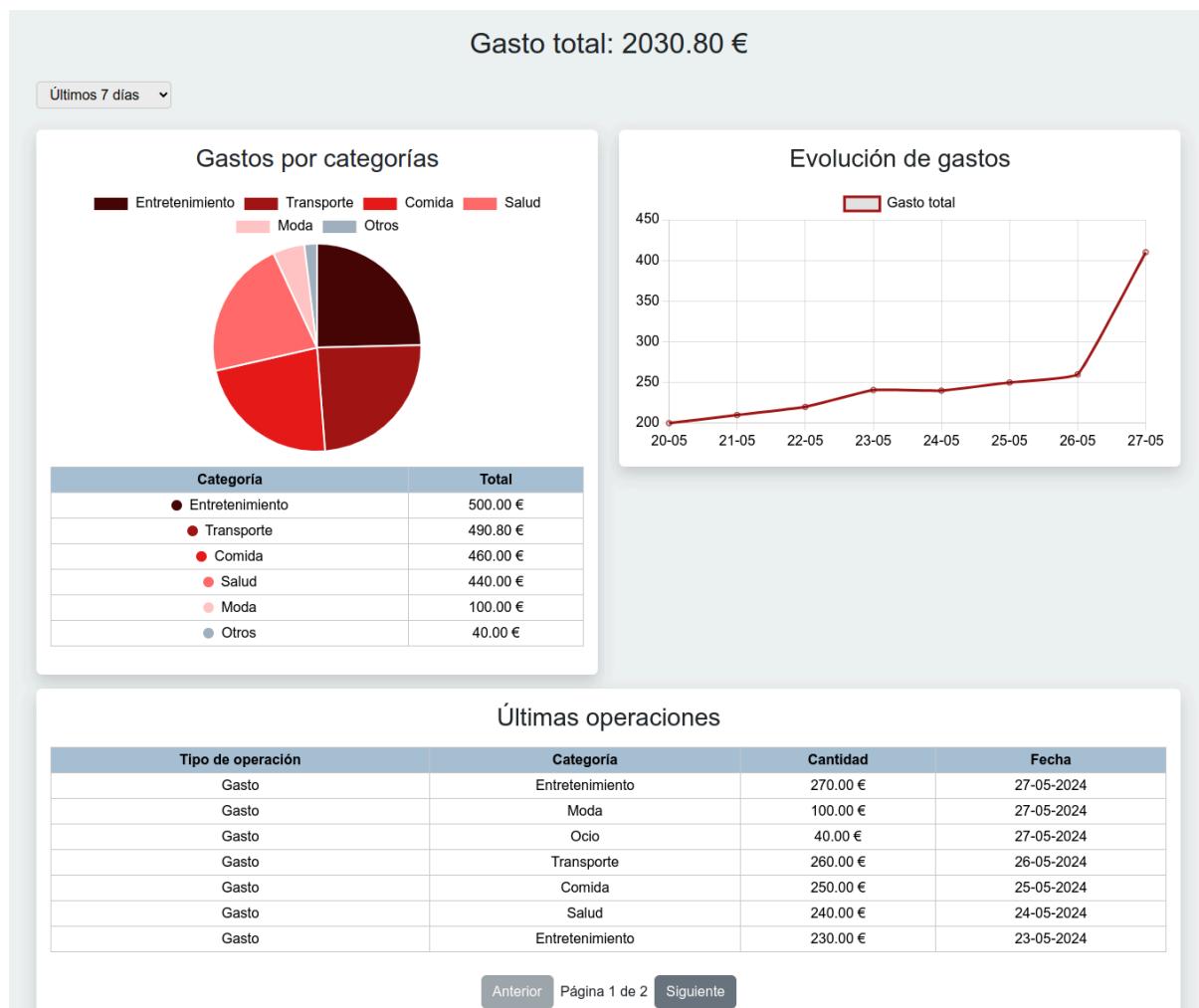


Figura 8.15: Vista Gastos

8.3.7. Ingresos

El comportamiento de la sección Ingresos es idéntico al de la sección de Gastos.

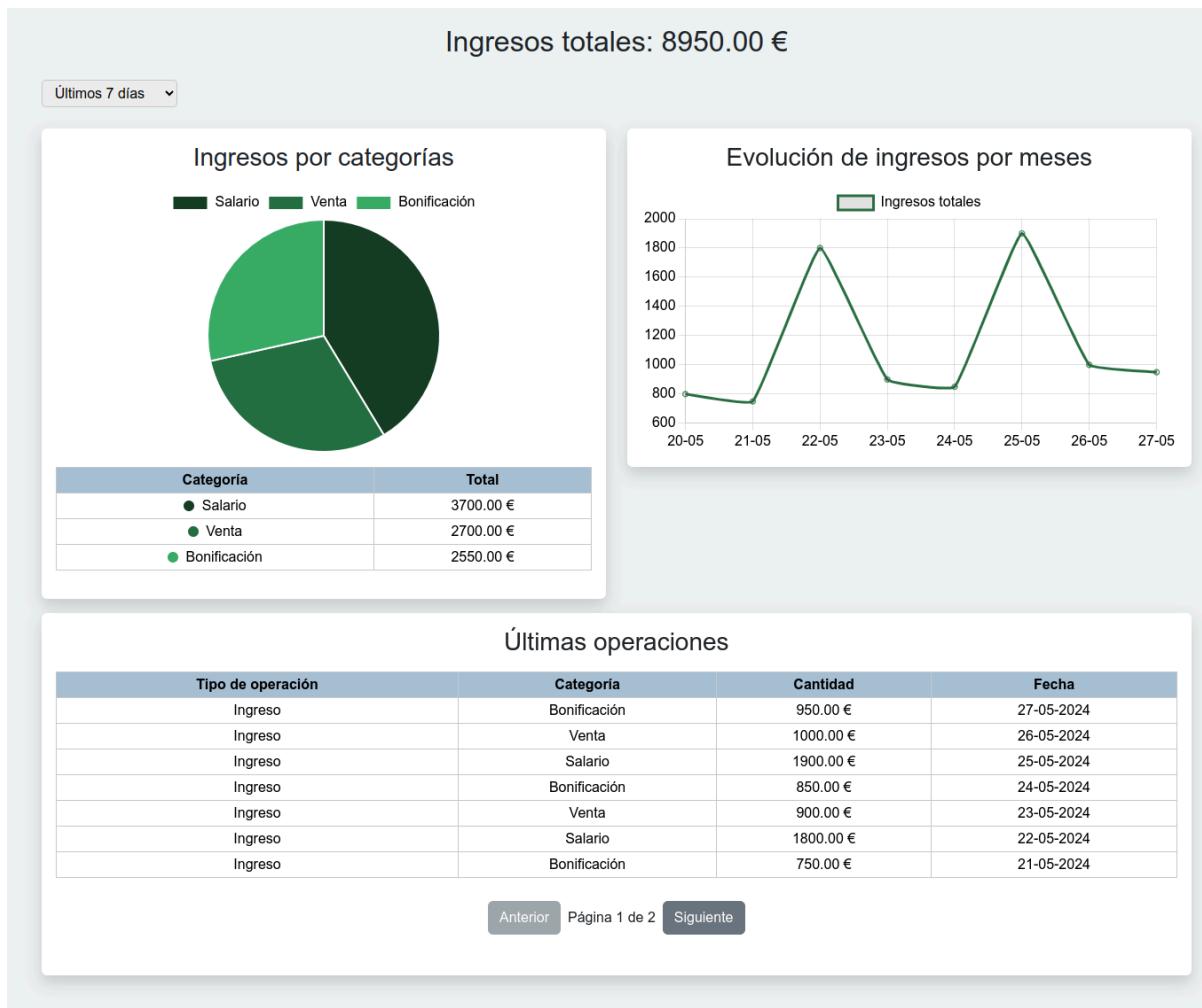


Figura 8.16: Vista Ingresos

8.3.8. Evolución

En este caso disponemos de un gráfico en la parte superior y debajo de él un gráfico a la izquierda y una tabla a la derecha:

- Gráfico de evolución gastos e ingresos: combina en una misma representación la evolución de los gastos e ingresos.
- Gráfico de ahorro: muestra los diferentes ahorros en función del tiempo.
- Tabla de ahorro: complementa el gráfico anterior mostrando de forma directa las cantidades concretas de los distintos ahorros.



Figura 8.17: Vista Evolución

8.3.9. Operaciones

Esta vista es un poco diferente a las anteriores y permite al usuario visualizar, filtrar, editar y eliminar sus operaciones. Estos filtros se hacen a través de una serie de parámetros: nombre de contabilidad, tipo de operación, categoría, cantidad mínima, cantidad máxima y un intervalo de tiempo con fechas de inicio y fin.

Operaciones

Contabilidad:
Operación:
Categoría:
Cantidad mínima:
0
Cantidad máxima:
1000
Fecha inicio:
dd/mm/aaaa
Fecha fin:
dd/mm/aaaa

Usuario	Contabilidad	Operación	Categoría	Cantidad	Fecha	Acciones		
username1	Contabilidad personal	Gasto	Transporte	20.00 €	03-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Entretenimiento	40.00 €	04-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Comida	25.00 €	05-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Transporte	35.00 €	06-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Entretenimiento	45.00 €	07-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Entretenimiento	40.00 €	08-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
username1	Contabilidad personal	Gasto	Comida	30.00 €	09-01-2024	<input type="button" value="Ver"/>	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Anterior Página 1 de 53 Siguiente

Figura 8.18: Vista Operaciones

Una vez introducidos los parámetros el usuario pulsa aplicar filtros y se muestran las operaciones que cumplan los mismos.

The screenshot shows a user interface for managing operations. At the top, there's a title 'Operaciones'. Below it is a search/filter bar with several dropdowns and input fields:

- Contabilidad: Contabilidad personal
- Operación: Ingreso
- Categoría: Salario
- Cantidad mínima: 0
- Cantidad máxima: 1000

Below the filter bar are two date inputs:

- Fecha inicio: 01/05/2024
- Fecha fin: 03/05/2024

At the bottom right of the filter area is a blue button labeled 'Aplicar Filtros' (Apply Filters).

Below the filter bar is a table with the following columns:

Usuario	Contabilidad	Operación	Categoría	Cantidad	Fecha	Acciones
username1	Contabilidad personal	Ingreso	Salario	1000.00 €	01-05-2024	<button>Ver</button> <button>Editar</button> <button>Eliminar</button>

Figura 8.19: Filtro aplicado en Vista Operaciones

8.3.10. Contabilidad compartida

Esta vista corresponde a las contabilidades compartidas y se muestran una serie de gráficos y tablas para mostrar información.

- Gráficos de categorías: existen gráficos por categorías como los vistos anteriormente tanto de ingresos como de gastos.
- Tabla de últimas operaciones: usada para representar las últimas operaciones de cualquier usuario de la contabilidad compartida.
- Detalles de gastos e ingresos: para cada usuario que haya introducido operaciones se suman todas sus operaciones, ofreciendo un total para ingresos y gastos.
- Lista de usuarios: enumera una lista con los usuarios que forman parte de la contabilidad compartida, así como sus roles.

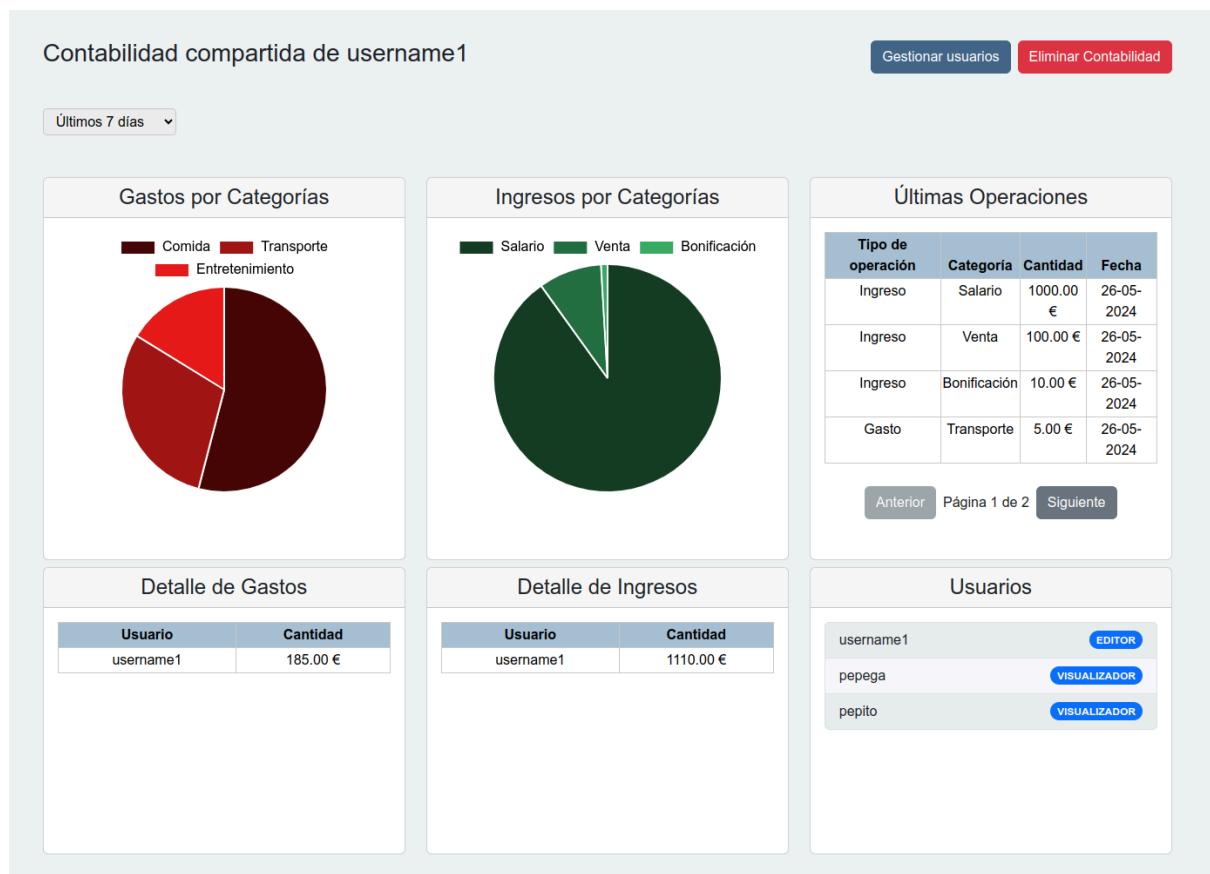


Figura 8.20: Vista Contabilidad Compartida

Arriba a la derecha, si el usuario es el creador de la contabilidad compartida se muestran botones de gestión de la misma.

- Gestionar usuarios: permite tanto añadir usuarios a la contabilidad compartida como eliminarlos.
- Eliminar contabilidad: como su nombre indica elimina la contabilidad compartida.

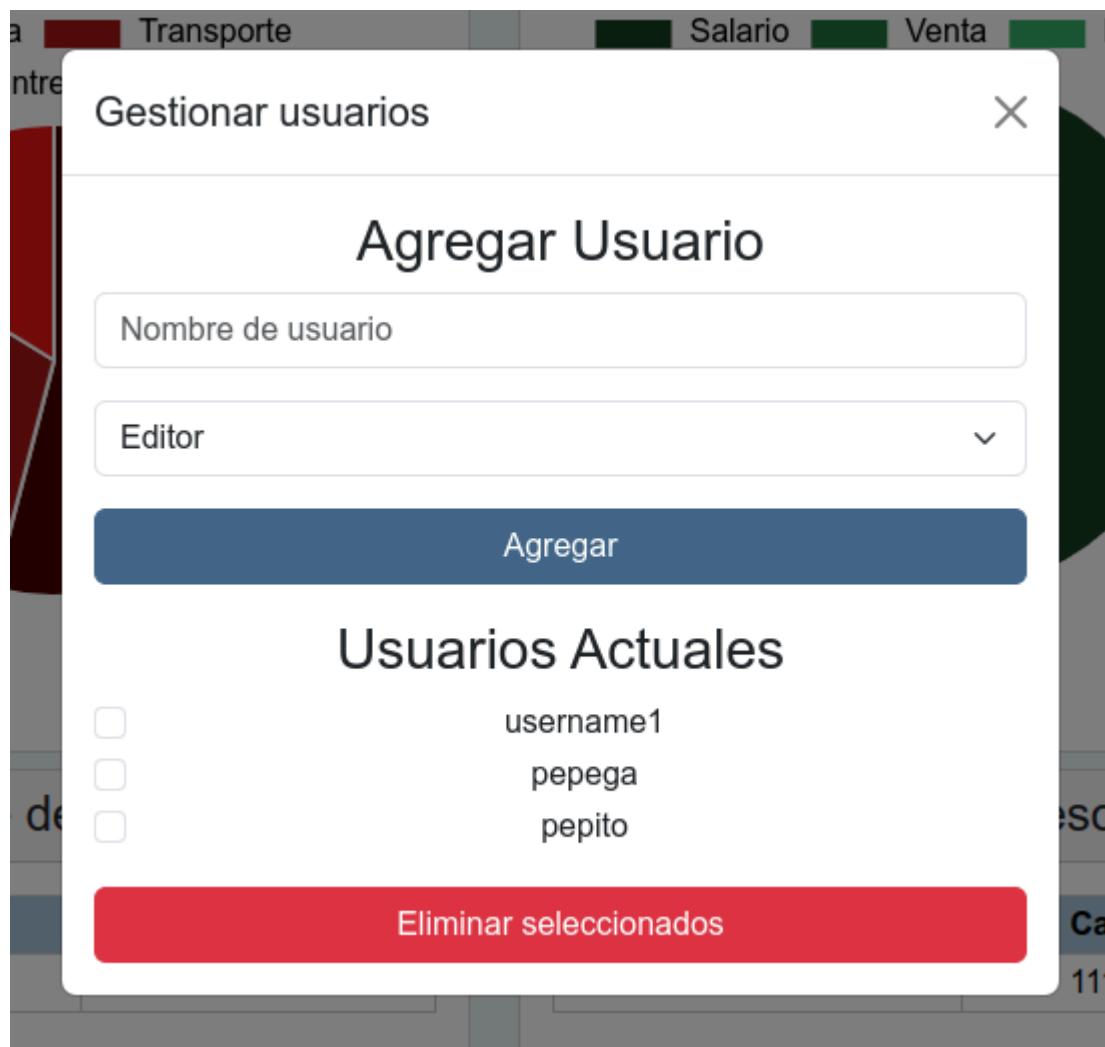


Figura 8.21: Modal de gestión de Contabilidad compartida

Capítulo 9: Conclusiones

Este capítulo concluye la memoria y sirve como reflexión sobre los objetivos conseguidos y los conocimientos aprendidos por el estudiante. Además se nombran algunas posibles vías de desarrollo futuras para el proyecto.

9.1. Análisis de objetivos

Los objetivos de este proyecto se detallaron en el primer capítulo. A continuación, se realizará un análisis de los mismos para determinar si se han completado o no.

Objetivos primarios:

- **Desarrollar una plataforma web para la gestión de finanzas personales y compartidas:** El desarrollo de la plataforma web ha sido exitoso. Se ha implementado una solución completa que permite a los usuarios gestionar tanto sus finanzas personales como compartidas desde cualquier PC con acceso a internet.
- **Mejorar la eficiencia en el registro y seguimiento de gastos:** Gracias a la interfaz y a las intuitivas funcionalidades, los usuarios pueden llevar un registro y seguimiento de sus gastos e ingresos de forma eficiente y sencilla. La plataforma ofrece herramientas de visualización y análisis que facilitan la toma de decisiones financieras.

Objetivos específicos/secundarios:

- **Diseñar un modelo de datos apropiado para la aplicación y sencillo para no dotar de gran complejidad al sistema:** Se ha creado un modelo de datos optimizado que soporta todas las funcionalidades de la aplicación sin añadir complejidad innecesaria. Este modelo ha demostrado ser eficiente y escalable durante las pruebas.
- **Implementar un sistema de autenticación eficaz y sencillo, que permita acceder a usuarios solo registrados y proteger sus datos:** Gracias a Spring y JSON Web Token, se ha logrado desarrollar un sistema de autenticación que dota al sistema de seguridad. Los datos de los usuarios están protegidos mediante encriptación, y el acceso está restringido a usuarios autenticados.
- **Implementar una API Rest que facilite la interacción entre el servidor y la interfaz de usuario:** Se ha implementado una API Rest robusta que permite una comunicación eficiente entre el servidor y la interfaz de usuario. Esta API soporta todas las operaciones necesarias para la gestión de finanzas.

9.2. Lecciones aprendidas

El desarrollo de este proyecto ha proporcionado valiosas lecciones que abarcan aspectos técnicos, de gestión y de colaboración.

Una de las principales lecciones aprendidas es la adopción de nuevas tecnologías. Integrar tecnologías como Spring Boot y Vue.js, que no habían sido utilizadas anteriormente, fue un desafío significativo pero extremadamente educativo. Aunque el tiempo dedicado al aprendizaje inicial ralentizó el progreso, a largo plazo mejoró considerablemente la calidad del trabajo y la eficiencia del desarrollo. Invertir tiempo en entender profundamente las nuevas tecnologías es crucial para el éxito de proyectos futuros.

Otra lección importante es la efectividad de la metodología iterativa. Optar por un enfoque iterativo permitió realizar ajustes y mejoras continuas, asegurando que el proyecto se mantuviera en la dirección correcta.

Además, la importancia de mantener una documentación detallada y actualizada no puede subestimarse. Facilitar tanto el desarrollo como las fases de pruebas y optimización fue esencial para el éxito del proyecto. Una buena documentación es útil no solo para el desarrollador actual, sino también para futuros mantenimientos y expansiones del sistema.

En resumen, este proyecto ha sido una experiencia de aprendizaje integral, reforzando la importancia de la adopción de nuevas tecnologías, la metodología iterativa y la documentación detallada. Estas lecciones no solo contribuyeron al éxito del proyecto actual, sino que también proporcionarán una base sólida para futuros desarrollos.

9.3. Trabajo futuro

A pesar de los logros alcanzados en este proyecto, hay varias áreas en las que se puede expandir y mejorar la plataforma en el futuro. Estas mejoras aumentarían la funcionalidad del sistema y además mejorarían la experiencia de usuario.

- **Integración con servicios bancarios:** La sincronización automática de transacciones bancarias con la plataforma permitiría a los usuarios tener una visión más completa y actualizada de sus finanzas. Esta funcionalidad reduciría la necesidad de entradas manuales, mejorando la precisión y la conveniencia.
- **Incorporación de funciones de red social:** Añadir funciones de red social podría enriquecer la experiencia del usuario y fomentar una mayor interacción dentro de la plataforma. Las funcionalidades podrían incluir listas de amigos, mensajería instantánea por chat o la capacidad de compartir logros financieros y consejos. Esto mejoraría la funcionalidad de las contabilidades compartidas y enriquecería la

aplicación.

- **Automatización de reportes financieros avanzados:** Desarrollar herramientas para la generación automática de reportes financieros avanzados es otra área prometedora. Estos reportes podrían incluir análisis detallados de gastos e ingresos, predicciones financieras basadas en patrones históricos y recomendaciones personalizadas para la optimización de las finanzas personales y compartidas.

Bibliografía

[1] Rise of money management app engagement. (s. f.).

<https://www.kantarmedia.com/news-and-resources/rise-of-money-management-app-engagement#:~:text=The%20impact%20of%20the%20cost.%20they%20can%20budget%20better.>

[2] Should you use a money management app? - Moya Financial Credit Union. (s. f.). Moya Financial Credit Union.

<https://moyafinancial.ca/blog/should-you-use-a-money-management-app/>

[3] Split expenses with friends. (s. f.). Splitwise. <https://www.splitwise.com/>

[4] SPENDEE a.s. www.spendee.com. (s. f.). Money Manager & Budget Planner | Spendee. Copyright 2024 SPENDEE a.s. | All Rights Reserved. <https://www.spendee.com/>

[5] Meardon, D. E. (s. f.). Diagramas de gantt | Atlassian. Atlassian. <https://www.atlassian.com/es/agile/project-management/gantt-chart>

[6] Software development the JetBrains way: dogfooding. (2024, May 27). [Video]. JetBrains. <https://www.jetbrains.com/es-es/>

[7] Postman API Platform | Sign Up for Free. (n.d.). Postman. <https://www.postman.com/>

[8] GitHub: Let's build from here. (2024). GitHub. <https://github.com/>

[9] Plataforma de almacenamiento personal en la nube y uso compartido de archivos - Google. (n.d.). https://www.google.com/intl/es_es/drive/

[10] draw.io - free flowchart maker and diagrams online. (n.d.). diagrams.net. <https://app.diagrams.net/>

[11] Google Meet. (n.d.). Google Meet. <https://meet.google.com/>

[12] Porter, B., Van Zyl, J., & Lamy, O. (n.d.). Maven – Welcome to Apache Maven. <https://maven.apache.org/>

[13] npm | Home. (n.d.). <https://www.npmjs.com/>

[14] Free online gantt chart software. (n.d.). <https://www.onlinegantt.com/#/gantt>

[15] Caules, C. Á. (2023, January 11). ¿Qué es un Microservicio? Arquitectura Java. <https://www.arquitecturajava.com/que-es-un-microservicio/>

[16] Caules, C. Á. (2023, December 29). Spring Boot JPA y su configuración. Arquitectura Java. <https://www.arquitecturajava.com/spring-boot-jpa-y-su-configuracion/>

- [17] Caules, C. Á. (2024, February 5). Inyección de Dependencia y su utilidad. Arquitectura Java. <https://www.arquitecturajava.com/el-patron-de-inyeccion-de-dependencia/>
- [18] Strmecki, D., & Strmecki, D. (2024, January 8). Singleton Design Pattern vs Singleton Beans in Spring Boot | Baeldung. Baeldung. <https://www.baeldung.com/spring-boot-singleton-vs-beans>
- [19] Caules, C. Á. (2023, January 11). Data Transfer Object (DTO) un concepto clave. Arquitectura Java. <https://www.arquitecturajava.com/data-transfer-object-dto-un-concepto-clave/>
- [20] Magaña, L. M. L. (2023, April 14). Qué es Json Web Token y cómo funciona. OpenWebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- [21] Spring boot. (n.d.). Spring Boot. <https://spring.io/projects/spring-boot>
- [22] Vue.js. (n.d.). <https://es.vuejs.org/>
- [23] MySQL. (n.d.). <https://www.mysql.com/>
- [24] Sueldo: Desarrolladores Junior en Granada 2024. (n.d.). Glassdoor. https://www.glassdoor.es/Sueldos/granada-desarrolladores-junior-sueldo-SRCH_IL.0.7_IC26_14045_KO8,30.htm