

MyFinance

Autor: Miguel Reyes Gómez

Tutor: David de la Torre Olegario

25/05/2025

Grado Superior Diseño de Aplicaciones Multiplataforma



salesianos
ALCALÁ DE HENARES

Índice

1. Resumen	6
2. Abstract.....	7
3. Palabras clave	8
4. DEFINICIONES, ABREVIATURAS Y ACRÓNIMOS.....	9
5. Autenticidad de la información	11
5.1 Declaración de Autenticidad del Trabajo	11
5.2 Origen del Contenido.....	11
5.3 Proceso de Desarrollo	11
5.4 Verificación de la Autenticidad	11
5.5 Conclusión	11
6. Motivación.....	12
6.1 Motivación personal	12
6.2 Motivación profesional.....	12
7. Objetivos del proyecto.....	13
7.1 Objetivo general	13
7.2 Objetivos específicos.....	13
8. Estado del arte	14
8.1 Money Manager.....	14
8.2 Wallet.....	14
8.3 Diferenciación de MyFinance	14
9. Propuesta	15
9.1 Contextualización del Dominio y Marco Tecnológico.....	15
9.2 Propuesta de Valor y Contribución al Ecosistema	15
9.3 Innovaciones y Características Diferenciadoras	15
10. Análisis del problema	16
10.1 Especificación de requisitos	16
10.2 Requisitos funcionales.....	16
10.3 Requisitos no funcionales.....	17
10.4 Arquitectura Conceptual del Sistema	17
10.4.1 Diagrama de casos de uso.....	17
10.4.2 Diagrama de flujo	19
10.4.3 Diagramas de secuencia.....	20
11. Análisis de la seguridad.....	25
11.1 Amenazas.....	25
11.2 Controles.....	26
12. Identificación y análisis de soluciones posibles	27

12.1 Alternativas analizadas	27
12.2 Criterios de selección.....	27
13. Solución propuesta	28
13.1 Descripción de la solución.....	28
13.2 Fases del desarrollo	28
13.3 Implementación y validación.....	29
14. Diseño de la solución.....	30
14.1 Arquitectura de capas	30
14.2 Componentes reutilizables.....	30
14.3 Flujo de navegación.....	30
14.4 Gestión de estados y reactividad	30
15. Diseño detallado	31
16. Tecnologías utilizadas	32
16.1 Entorno de desarrollo.....	32
16.2 Lenguaje de desarrollo.....	32
16.3 Sistema operativo	32
16.4 Frameworks y bibliotecas	32
16.5 Control de versiones	32
17. Desarrollo de solución propuesta	33
17.1 Proceso de desarrollo.....	33
17.2 Problemas y dificultades.....	33
17.3 Decisiones tomadas	33
17.4 Particularidades de la solución final.....	33
18. Implantación	34
18.1 Instalación en dispositivos de prueba	34
18.2 Pruebas en entorno real.....	34
18.3 Pantalla inicio	35
18.4 Pantalla Historial	36
18.5 Pantalla Transacciones	37
18.6 Pantalla calculadora.....	38
18.7 Pantalla perfil.....	39
19. Plan de pruebas	40
20. Conclusiones.....	42
21. Relación con los estudios cursados	42
21.1 Base de datos	42
21.2 Programación.....	42
21.3 Desarrollo de interfaces	42

22. Referencias	43
23. Anexos	44
23.1 Estructura del proyecto	44
23.1.1 Módulo raíz.....	44
23.1.2 Módulo componentes	44
23.1.3 Módulo data.....	44
23.1.4 Módulo pantallas	45
23.1.5 Módulo theme.....	45
23.1.6 Módulo viewmodel.....	45
23.2 Manual de usuario	46
23.2.1 ¿Qué es MyFinance?	46
23.2.2 Requisitos mínimos	46
23.2.3 Instalación.....	46
23.2.4 Primero pasos	46
23.2.5 Navegación general	47
23.2.6 Uso de la aplicación.....	47
23.2.7 Seguridad y privacidad	48
23.2.8 Preguntas frecuentes (FAQ)	48
23.2.9 Contacto.....	49

Tabla de ilustraciones

Figura 1 - Diagrama de casos de uso	18
Figura 2 - Diagrama de flujo	19
Figura 3 - Diagrama secuencia – Inicio App	20
Figura 4 - Diagrama secuencia – Añadir transacción	21
Figura 5 - Diagrama secuencia – Historial	22
Figura 6 - Diagrama secuencia – Uso calculadora	23
Figura 7 - Diagrama secuencia – Actualizar perfil	24
Figura 8 - Pantalla Inicio	35
Figura 9 - Pantalla Historial	36
Figura 10 - Pantalla Transacción	37
Figura 11 - Pantalla Calculadora	38
Figura 12 - Pantalla Perfil	39
Figura 13 - Módulo Raíz	44
Figura 14 - Módulo components	44
Figura 15 - Módulo data	44
Figura 16 - Módulo pantallas	45
Figura 17 - Módulo theme	45
Figura 18 - Módulo viewmodel	45

1. Resumen

MyFinance es una aplicación móvil desarrollada para el Trabajo de Fin de Grado, cuyo objetivo principal es facilitar el control financiero personal de forma práctica, accesible e intuitiva.

Esta app está dirigida al público general y permite registrar y categorizar tanto ingresos como gastos, ofreciendo una visión clara del estado económico del usuario mediante estadísticas personalizadas, un historial de movimientos recientes y una hucha virtual que se actualiza automáticamente con cada operación. Además, incluye una calculadora de interés compuesto personalizable, gestión de perfil con nombre de usuario y foto, y la opción de alternar entre modo claro y oscuro para mejorar la experiencia.

La metodología de desarrollo la organicé en tres puntos clave. Análisis de requisitos, diseño de interfaz y codificación y validación mediante pruebas.

Para la implementación se emplearon Kotlin y Jetpack Compose en Android Studio, con SQLite como base de datos local. El control de versiones se gestionó a través de GitHub.

En conclusión, este proyecto forma parte del ciclo formativo de Desarrollo de Aplicaciones Multiplataforma (DAM), donde he podido aplicar lo aprendido sobre programación, bases de datos y diseño de interfaces. Más allá del desarrollo técnico, esta aplicación refleja mi interés por crear herramientas útiles que puedan ayudar a otras personas a gestionar mejor su economía del día a día.

2. Abstract

MyFinance is a mobile application developed for the Final Degree Project, whose main objective is to facilitate personal financial management in a practical, accessible and intuitive way.

This app is aimed at the general public and allows users to record and categorize both income and expenses, offering a clear view of their financial status through personalized statistics, a history of recent transactions, and a "virtual piggy bank" that updates automatically with each transaction. It also includes a customizable compound interest calculator, profile management with a username and photo, and the option to switch between light and dark mode to improve the experience.

I organized the development methodology into three key points: requirements analysis, interface design, and coding and validation through testing.

I organized the work in three steps: first I analyzed the requirements, then I designed the screens, and finally I built and tested the app. I wrote the code in Android Studio using Kotlin and Jetpack Compose, and I used SQLite for the database. Version control was managed through GitHub and all my code is there.

In conclusion, this project is part of the Desarrollo de Aplicaciones Multiplataforma (DAM). I used what I learned about coding, databases, and design. Beyond the technical side, MyFinance shows my wish to create helpful tools that make everyday money management easier.

3. Palabras clave

Finanzas personales

Android offline-first

Kotlin

Header

Jetpack Compose

Room (SQLite)

Interés compuesto

Framework

Estructura MVVM

4. DEFINICIONES, ABREVIATURAS Y ACRÓNIMOS

- **SQLite**
Motor de base de datos relacional embebido que permite almacenamiento local sin necesidad de servidor externo.
- **Room**
Biblioteca de persistencia de datos de Android que actúa como capa de abstracción sobre SQLite, simplificando operaciones CRUD.
- **CRUD**
Create, Read, Update, Delete. Operaciones básicas de persistencia de datos.
- **Jetpack Compose**
Framework de UI moderno de Android que permite construir interfaces de usuario de forma declarativa en Kotlin.
- **DAO – Data Access Object**
Componente de Room que encapsula consultas SQL y devuelve objetos Kotlin.
- **UI**
User Interface (Interfaz de Usuario). Parte visual con la que interactúa el usuario.
- **MVVM – Model-View-ViewModel**
Patrón arquitectónico que separa la lógica de presentación (ViewModel) de la UI (View) y de los datos (Model).
- **Kotlin**
Lenguaje de programación moderno, expresivo y seguro, oficial para el desarrollo de aplicaciones Android.

- **Material Design**

Sistema de diseño creado por Google que proporciona pautas visuales y de interacción para construir interfaces consistentes y accesibles.

- **Graddle**

Sistema de automatización de compilación que gestiona dependencias, tareas de construcción y empaquetado en proyectos Android.

- **IDE – Integrated Development Enviroment**

Entorno de Desarrollo Integrado donde se escribe y depura la aplicación

- **POO – Programación Orientada a Objetos**

Programación Orientada a Objetos. Paradigma de programación basado en la organización del código mediante clases y objetos.

5. Autenticidad de la información

5.1 Declaración de Autenticidad del Trabajo

Yo, Miguel Reyes Gómez [09147403G], autor del Trabajo de Fin de Grado (TFG), declaro que el presente proyecto titulado “MyFinance”, es fruto de mi trabajo personal e inédito. Todos los contenidos, datos y resultados expuestos han sido generados por mí y no provienen de ninguna fuente ajena.

5.2 Origen del Contenido

El material presentado en este TFG ha sido desarrollado íntegramente por el autor, sin recurrir indebidamente a ideas, textos o recursos de terceros. Los algoritmos, diseños y soluciones mostrados se han programado y documentado desde cero.

5.3 Proceso de Desarrollo

El diseño, la implementación y las pruebas de esta aplicación se han llevado a cabo de forma individual. He dedicado tiempo a investigar las mejores prácticas, diseñar la interfaz, codificar cada módulo y ejecutar pruebas unitarias e integradas para validar su correcto funcionamiento.

5.4 Verificación de la Autenticidad

He revisado detalladamente cada sección de este documento y comprobado que no existe plagio ni apropiación indebida de trabajos previos. Asimismo, he sometido el texto a herramientas de detección de similitudes para asegurar su originalidad.

5.5 Conclusión

En virtud de lo expuesto, certifico que este Trabajo de Fin de Grado es fruto de mi esfuerzo personal y original, y asumo plena responsabilidad por su integridad. Me comprometo a responder ante cualquier eventualidad relacionada con el cumplimiento de los principios éticos y académicos establecidos.

6. Motivación

6.1 Motivación personal

Mi motivación para emprender este Trabajo de Fin de Grado (TFG), surge de mi inquietud por profundizar en Kotlin y Jetpack Compose, ambos los descubrí durante el ciclo y me resultaron muy atractivos para el desarrollo móvil.

Al pensar en el tema de mi Trabajo de Fin de Grado, me resultó muy fácil elegirlo, el mundo de las finanzas personales siempre me ha llamado la atención, cómo las personas organizan su presupuesto y qué herramientas utilizan para ello. Quise unir ambas pasiones, la tecnología y las finanzas, en una sola aplicación, y así nació la idea de MyFinance.

Trabajar en un proyecto que combina un lenguaje con el que me siento cómodo y un tema que me apasiona me ha mantenido motivado en cada fase.

En definitiva, MyFinance representa para mí la oportunidad perfecta de consolidar mis habilidades en Kotlin, al tiempo que desarrollo una herramienta útil vinculada a una disciplina que siempre me ha interesado.

6.2 Motivación profesional

Desde un enfoque profesional, este proyecto representa una oportunidad única para afianzar mis competencias en el diseño y construcción de aplicaciones Android. La experiencia práctica que obtenga al desarrollar esta app reforzará mi perfil como desarrollador especializado en el ecosistema Android, un sector con alta demanda en la industria tecnológica.

Además, trabajar con Android Studio y aplicar las mejores prácticas de desarrollo en esta plataforma son aspectos esenciales para mi crecimiento profesional. Gracias a este proyecto, no solo pongo en práctica mis habilidades técnicas, sino que demuestro mi capacidad para superar retos complejos y entregar soluciones innovadoras en el ámbito del desarrollo móvil para Android.

7. Objetivos del proyecto

7.1 Objetivo general

El propósito principal de este Trabajo de Fin de Grado es desarrollar una aplicación móvil que ofrezca una solución integral para la gestión de finanzas personales, combinando una interfaz intuitiva y accesible con funcionalidades avanzadas de análisis y proyección económica. La herramienta permitirá al usuario registrar, categorizar y visualizar sus ingresos y gastos de forma clara y estructurada; planificar ahorros mediante una calculadora de interés compuesto totalmente personalizable; y gestionar su perfil.

7.2 Objetivos específicos

Para cumplir con el objetivo general del proyecto, tuve en cuenta una serie de objetivos específicos. En primer lugar, se contempla la implementación de un sistema de registro y categorización de ingresos y gastos. Este sistema incluirá una pantalla de transacciones para introducir datos como la fecha, el importe y la categoría de cada transacción. La información se guardará localmente gracias a SQLite. Además, la pantalla transacciones validará que los datos sean correctos y gestionará los errores con el fin de garantizar la integridad de los datos.

Otro objetivo es diseñar una interfaz de usuario moderna y visual, para ello aplicaré Material Design y utilizaré Jetpack Compose. La aplicación tendrá cinco pantallas principales (Inicio, Historial, Añadir transacciones, Calculadora y Perfil), todas ellas tendrán el mismo header y una barra de navegación.

Para el último objetivo específico, se desarrollará una sección de perfil donde el usuario podrá editar el nickname y poner una foto de perfil, guardando estos datos localmente con SQLite. También contará con un selector para alternar entre modo oscuro y claro.

8. Estado del arte

La sección de estado del arte revisa las soluciones y conocimientos existentes en el ámbito de las aplicaciones para finanzas personales, sirviendo de base para identificar carencias y oportunidades de mejora.

8.1 Money Manager

Money Manager destaca por sus avanzadas herramientas de análisis de gastos, que incluyen gráficos detallados por categoría y períodos. Ofrece plantillas de presupuesto predefinidas y permite sincronizar cuentas bancarias mediante importación automática de CSV. Aunque tiene buena reputación por su profundidad analítica, puede resultar complejo para usuarios comunes debido a la gran cantidad de opciones y ajustes disponibles.

8.2 Wallet

Wallet se centra en la gestión colaborativa de presupuestos, facilitando compartir cuentas familiares o de grupo. Incluye recordatorios de facturas y plantillas de presupuestos, así como sincronización con bancos para importación de movimientos. Su interfaz es limpia, pero algunas funciones avanzadas requieren suscripción de pago, lo que puede limitar a usuarios con necesidades básicas.

8.3 Diferenciación de MyFinance

A diferencia de Money Manager y Wallet, MyFinance apuesta por un enfoque 100% local, sin depender de la nube ni de suscripciones. Su “hucha virtual” gamificada y la calculadora de interés compuesto están integradas de serie, sin módulos de pago extra. Además, MyFinance ofrece un diseño minimalista basado en Material Design y un esquema de navegación unificado en cinco pantallas, garantizando una curva de aprendizaje suave para cualquier usuario. La alternancia inmediata entre modo claro y oscuro y la gestión de perfil local subrayan su compromiso con la privacidad y la simplicidad.

9. Propuesta

9.1 Contextualización del Dominio y Marco Tecnológico

Este Trabajo de Fin de Grado se posiciona en el contexto del desarrollo de aplicaciones móviles enfocadas a la gestión de finanzas personales, un área con creciente demanda de soluciones accesibles y seguras para el usuario. Aunque existen numerosas apps que ofrecen registro de transacciones y análisis de gastos, muchas dependen de la nube o presentan interfaces complejas que dificultan su adopción por el público general.

9.2 Propuesta de Valor y Contribución al Ecosistema

MyFinance se distingue por ofrecer privacidad total y gran facilidad de uso, cubriendo registro, categorización y visualización de gastos sin depender de suscripciones o servicios en la nube. Al emplear Kotlin y Jetpack Compose, demuestra cómo crear apps Android modulares, accesibles y preparadas para futuras extensiones manteniendo siempre los datos en el dispositivo.

9.3 Innovaciones y Características Diferenciadoras

Esta aplicación propone innovaciones y en este punto voy a hablar de ellas. Una de las principales es una hucha virtual, ésta es un componente que se actualiza automáticamente con cada transacción registrada. Esto ofrece al usuario la facilidad de observar el balance de su cuenta de un simple vistazo.

Otra innovación es la calculadora de interés compuesto, permite a los usuarios simular escenarios financieros ajustando los parámetros como el capital inicial, la aportación mensual, la tasa de interés y finalmente elegir durante cuánto tiempo quieren verlo. Tras rellenar estos datos, aparecerá un gráfico que muestra cuánto dinero generaría la supuesta cuenta de interés en base a los datos que haya introducido el usuario

Por último, una innovación destacable puede ser el funcionamiento de la app tanto en modo claro como oscuro, al alternar de modo, tanto las pantallas como la barra de navegación y el header se adaptan correctamente en función del modo seleccionado.

10. Análisis del problema

El análisis del problema identifica las carencias y dificultades que enfrentan los usuarios al gestionar sus finanzas personales con las soluciones existentes, sirviendo de base para justificar el desarrollo de MyFinance.

10.1 Especificación de requisitos

Se llevará a cabo una definición exhaustiva de los requerimientos del sistema, que servirá de guía para el diseño y la implementación de MyFinance. Estos requisitos se estructurarán en dos bloques claramente diferenciados y documentados.

10.2 Requisitos funcionales

La aplicación debe cumplir unos requisitos funcionales, en este punto hablaremos sobre ellos. En primer lugar, el sistema debe permitir registrar nuevas transacciones, indicando el importe, la descripción, la fecha y la categoría.

El usuario también contará con una pantalla en la que podrá ver el historial completo de todas las transacciones. En esta pantalla el usuario podrá filtrar por el tipo de transacción (ingreso/gasto), también aparecerá un icono al lado de la transacción para indicar el tipo de categoría. Adicionalmente, la aplicación generará gráficos en la pantalla de inicio que mostrarán los ingresos y gastos organizados por categorías.

Uno de los componentes más importantes es la calculadora de interés compuesto, que permitirá al usuario realizar simulaciones financieras. Deberá introducir el capital inicial, la tasa de interés, la duración del periodo y las aportaciones periódicas, al pulsar el botón de calcular, el sistema generará un gráfico que mostrará el crecimiento del ahorro.

Finalmente, se implementará una funcionalidad para gestionar el perfil, desde la cual el usuario podrá editar su nombre de usuario y seleccionar o modificar una foto de perfil. Además, podrá cambiar la apariencia de la aplicación eligiendo entre modo claro u oscuro.

10.3 Requisitos no funcionales

La aplicación debe cumplir con unos requisitos no funcionales que garanticen su privacidad, rendimiento y accesibilidad. Todos los datos del usuario deben mantenerse exclusivamente en el dispositivo, sin servicios en la nube. Además, el sistema ofrecerá un rendimiento rápido y fluidos en todas las operaciones de lectura y escritura. La interfaz debe ser intuitiva y fácil de usar para todos los públicos.

10.4 Arquitectura Conceptual del Sistema

Antes de iniciar con los diagramas, se llevará a cabo un modelado conceptual del sistema empleando UML (Unified Modeling Language). Se utilizarán diagramas de casos de uso, de secuencia y de flujo para representar gráficamente las estructuras y relaciones internas, así como la interacción entre los usuarios y los distintos componentes del sistema.

10.4.1 Diagrama de casos de uso

El diagrama de casos de uso muestra cómo el usuario se relaciona con las funcionalidades clave de MyFinance. El actor principal es el usuario, cada una de estas funcionalidades se representa mediante un caso de uso, que refleja una acción concreta que el usuario puede realizar.

Las flechas punteadas con la etiqueta «include» indican funcionalidades obligatorias que siempre se ejecutan dentro de otro caso de uso, por ejemplo, la acción de registrar una transacción incluye de forma obligatoria la categorización de dicha transacción.

Al visualizar el historial de transacciones, el usuario tiene la posibilidad de filtrar según el tipo de transacción (ingreso/gasto), lo que crea otra relación de inclusión. En la gestión del perfil, el usuario podrá hacer varias acciones; la edición del nombre del usuario, la selección de la foto de perfil, y la alternancia entre el modo claro y oscuro.

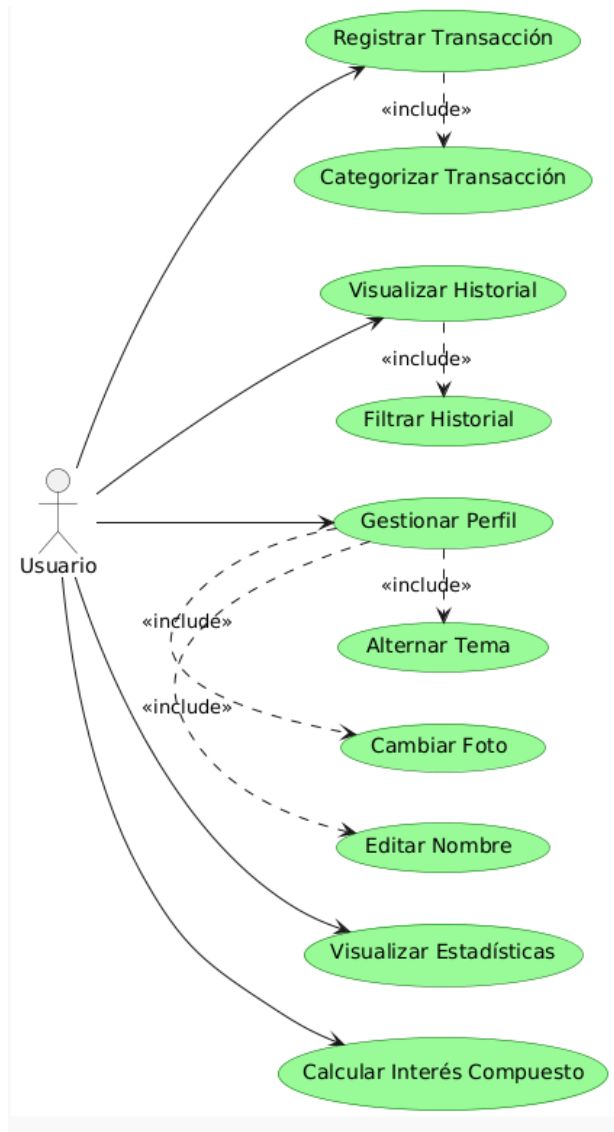


Figura 1 - Diagrama de casos de uso

10.4.2 Diagrama de flujo

El diagrama de flujo muestra que puede hacer el usuario en cada pantalla. La aplicación se inicia en la pantalla de inicio, en esta pantalla el usuario puede ver el balance actual de su cuenta, el total de gastos e ingresos y dos gráficos que muestran los gastos e ingresos organizados por categoría. En la pantalla del historial, el usuario podrá ver todas las transacciones en una lista ordenadas por fecha y también podrá filtrar según el tipo de transacción (ingreso/gasto).

En la pantalla de transacciones, el usuario podrá rellenar un formulario en el que introduciendo; el nombre de la transacción, el tipo, el monto y la categoría, podrá añadir una transacción nueva. Automáticamente la aplicación recogerá la fecha actual y no te permitirá añadir una transacción si no está todos los campos rellenos.

La aplicación también cuenta con la pantalla de la calculadora de interés compuesto. En esta pantalla, el usuario añadiendo el capital inicial, la tasa de interés, la aportación mensual y el plazo, podrá ver en un gráfico una proyección de este interés.

Finalmente, en la pantalla de perfil, el usuario podrá asignar su nombre de usuario y su foto de perfil, también podrá alternar entre modo oscuro y claro.

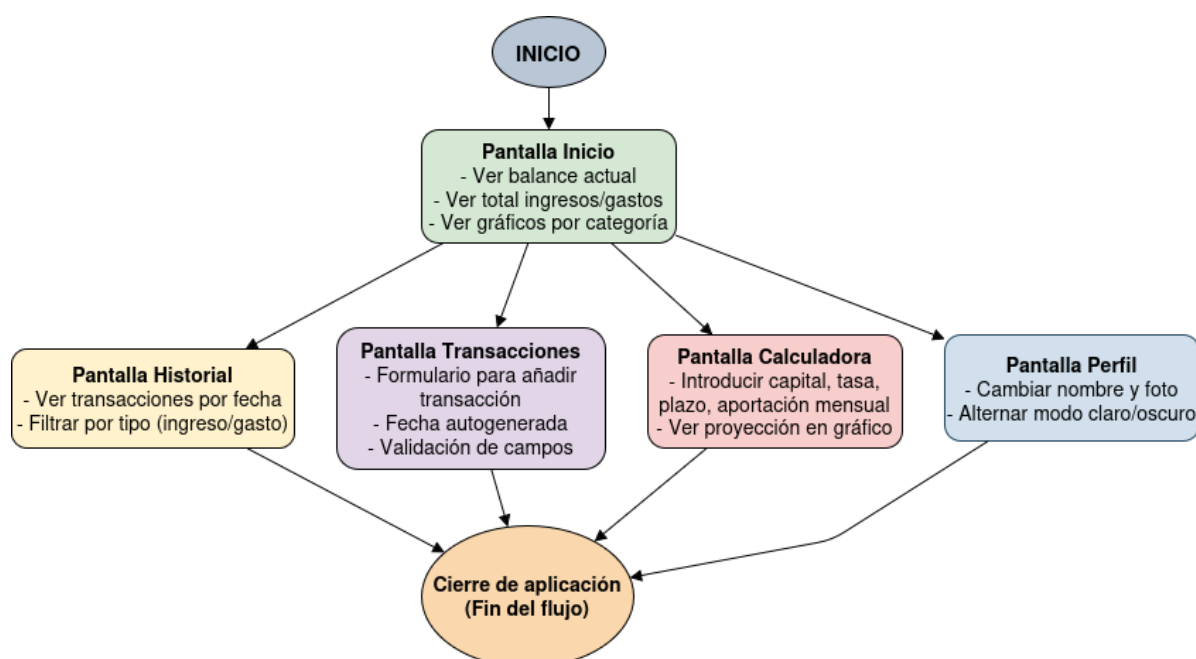


Figura 2 - Diagrama de flujo

10.4.3 Diagramas de secuencia

Cuando el usuario abre la aplicación, el sistema operativo se encarga de inicializarla. A continuación, la actividad principal (MainActivity) se lanza automáticamente y se encarga de renderizar la pantalla de inicio. La pantalla de inicio, muestra balance actual, los ingresos y gastos actuales y dos gráficos estadísticos que muestran los ingresos y gastos ordenados por categoría, desde aquí, se podrá acceder a todas las pantallas de la aplicación. Este flujo de eventos constituye la secuencia básica al arrancar la aplicación.

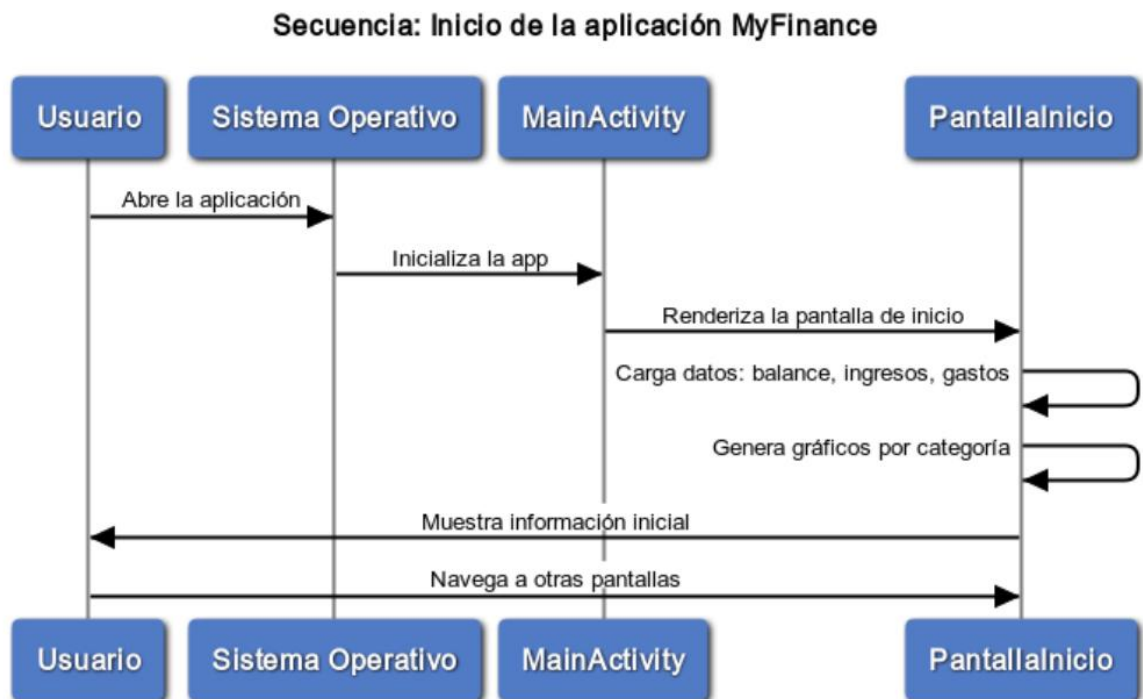


Figura 3 - Diagrama secuencia – Inicio App

Cuando el usuario desea añadir una nueva transacción, debe ir a la pantalla de transacciones. Desde allí, debe rellenar todos los campos y una vez rellenados debe pulsar el botón de guardar. El botón de guardar permanecerá deshabilitado hasta que estén todos los campos rellenos. Una vez pulsado el botón de guardar, la aplicación crea una nueva transacción y la guarda localmente. Automáticamente se actualizan todos los campos tanto en la pantalla del historial como en la pantalla de inicio.

Secuencia: Añadir nueva transacción con validación

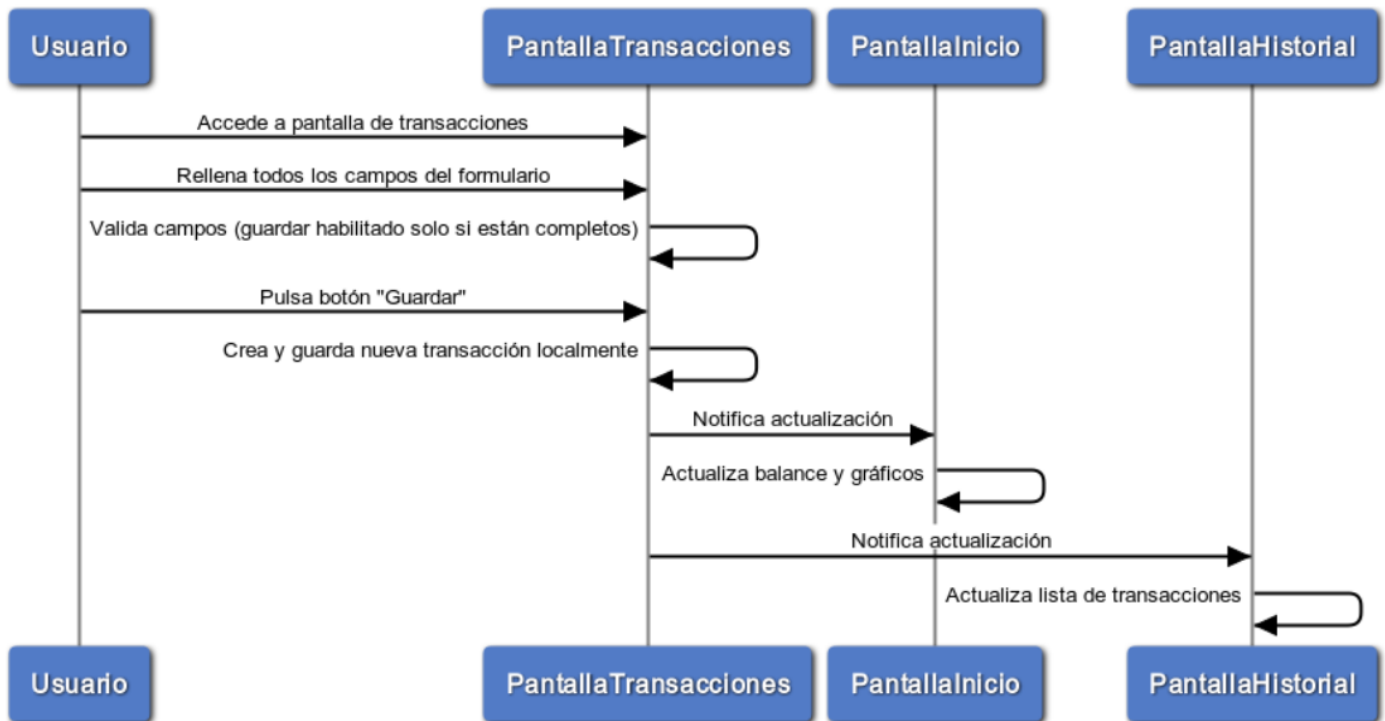


Figura 4 - Diagrama secuencia – Añadir transacción

Cuando el usuario accede a la pantalla del historial, ésta muestra una lista completa y ordenada por fecha reciente de todas las transacciones. Esto permite al usuario visualizar fácilmente sus transacciones, la pantalla tiene tres botones que permiten filtrar la lista según el tipo de transacción; ingresos, gastos o todas. La lista se actualiza automáticamente cada vez que se añade una transacción.

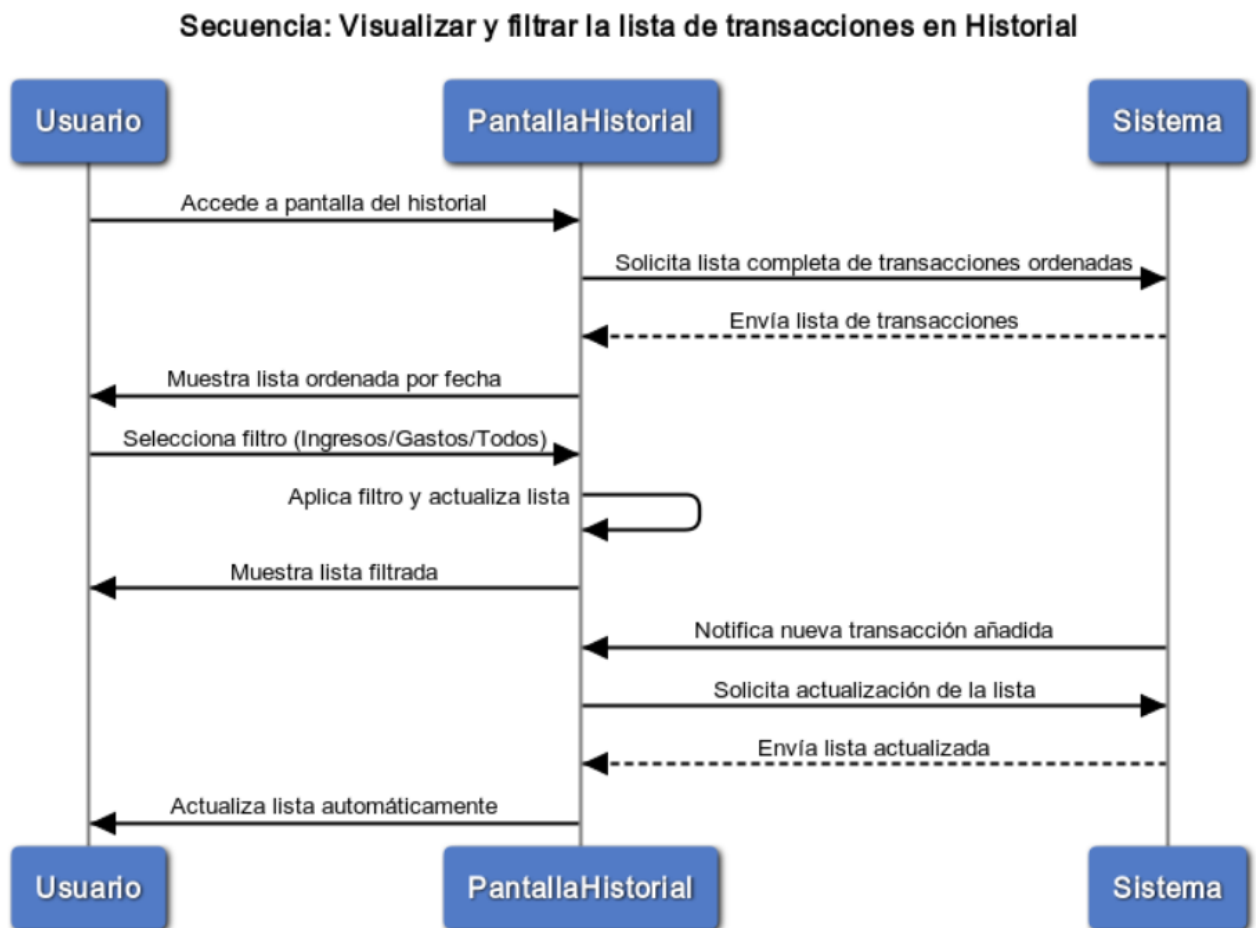


Figura 5 - Diagrama secuencia – Historial

Cuando el usuario accede a la pantalla de la calculadora, rellena los campos necesarios para realizar el cálculo, deben estar todos rellenos, sino la aplicación no te permitirá calcular nada ya que el botón de "Calcular", estará deshabilitado. Una vez todos los campos estén rellenos, podemos pulsar en ese botón y aparecerá un gráfico que permite ver al usuario una proyección del interés compuesto según los datos introducidos.

Secuencia: Uso de la calculadora de interés compuesto

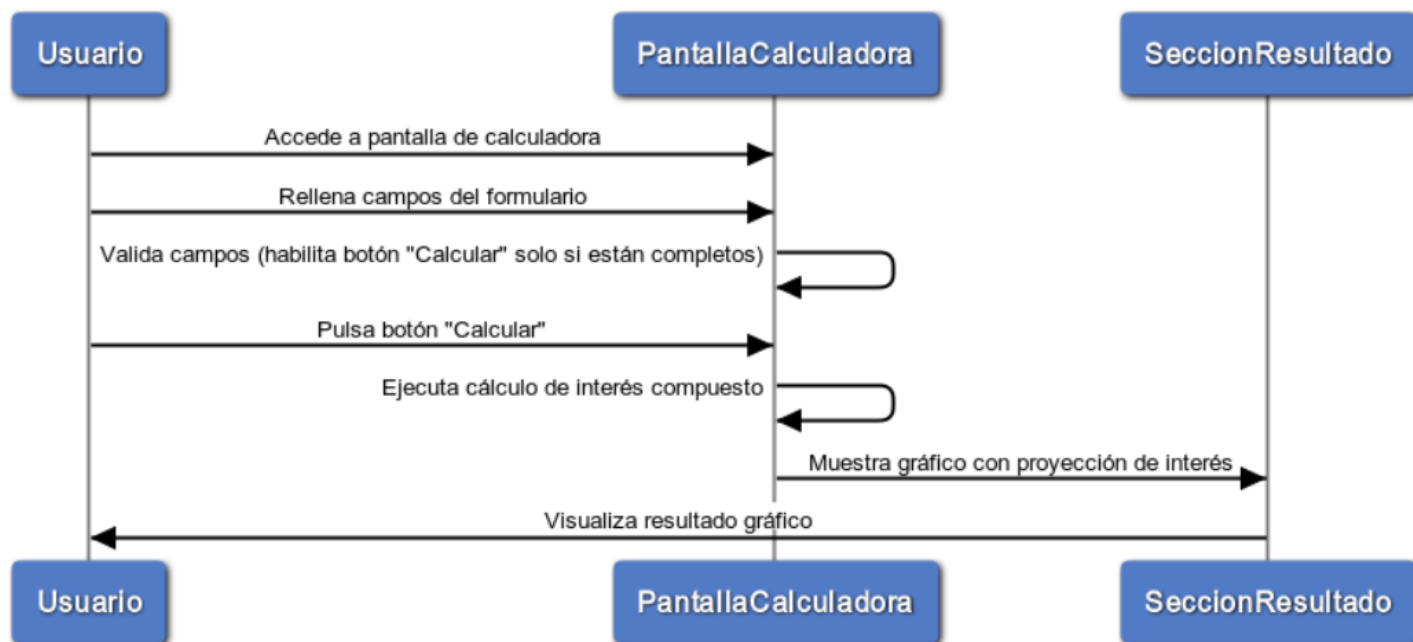


Figura 6 - Diagrama secuencia – Uso calculadora

Cuando el usuario accede a la pantalla del perfil, puede tocar el botón “Editar perfil” y la pantalla cambia a un modo de edición dónde el usuario puede modificar tanto el nombre del usuario como la foto de perfil. Una vez realizados los cambios, el usuario pulsa el botón “Guardar cambios” para confirmar. La aplicación guarda la información actualizada y la pantalla muestra los datos actualizados.

En esta pantalla el usuario también podrá alternar entre modo oscuro y claro, al pulsar en el botón con una luna el tema de la app cambiará a oscuro y al pulsar en el botón con un sol, el tema de la app cambiará a claro.

Secuencia: Actualizar perfil y alternar tema en pantalla de Perfil

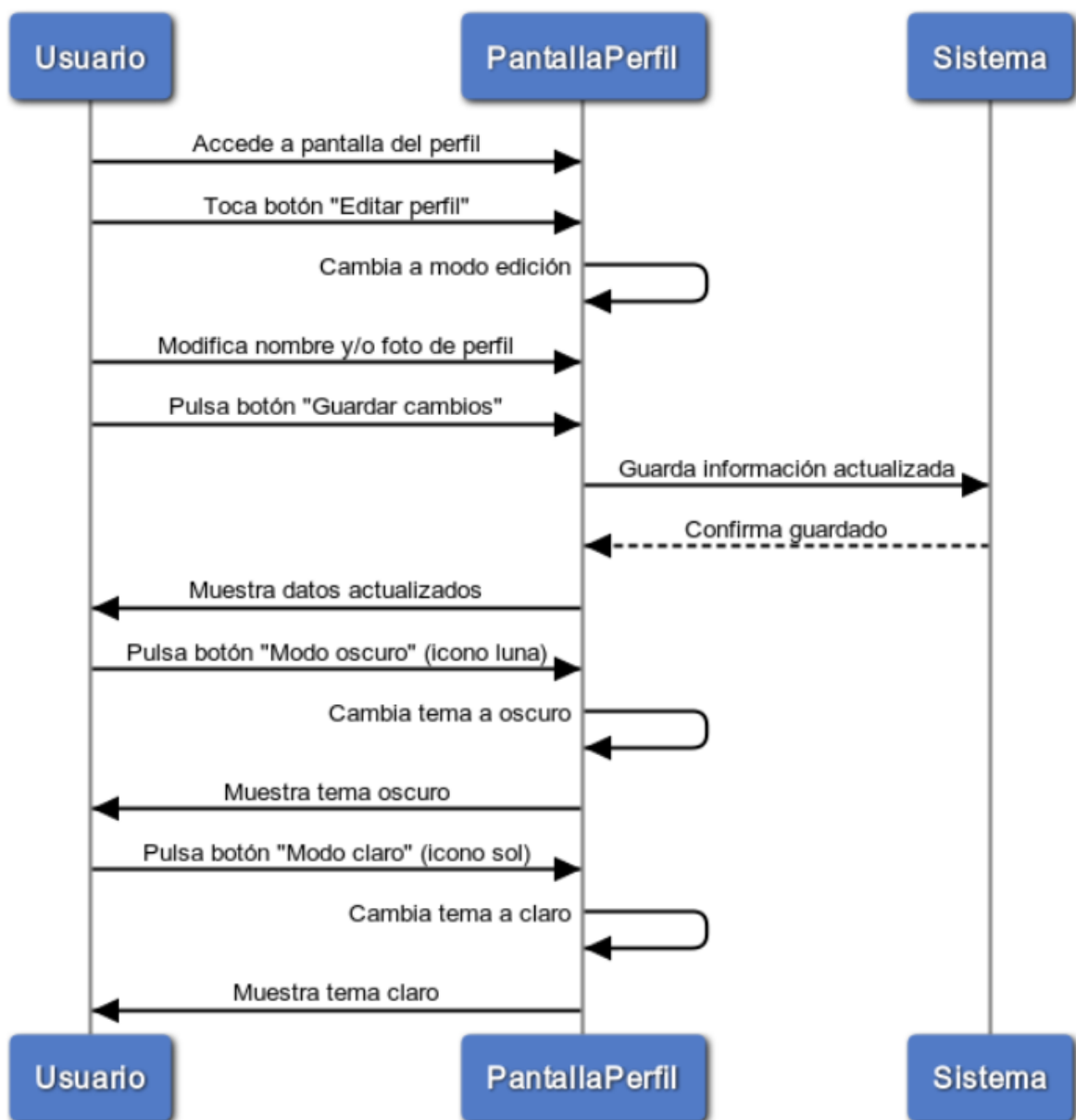


Figura 7 - Diagrama secuencia – Actualizar perfil

11. Análisis de la seguridad

11.1 Amenazas

ID	Amenaza	Descripción	Impacto
A-01	Acceso físico no autorizado	Cualquier persona con acceso al dispositivo puede abrir la app y ver los datos.	Pérdida de privacidad financiera del usuario; exposición de información sensible personal.
A-02	Manipulación o eliminación de datos	El usuario o un tercero pueden editar o borrar registros sin restricciones.	Distorsión de la información registrada, pérdida de trazabilidad y errores en el control financiero.
A-03	Pérdida total de datos	Al no haber copias de seguridad, una desinstalación borra todos los datos.	Pérdida irreversible de toda la información financiera almacenada por el usuario.
A-04	Lectura directa de la base de datos	Cualquier app con permisos o acceso root puede leer directamente la base de datos SQLite.	Riesgo de robo de datos si el dispositivo está comprometido o rooteado.

11.2 Controles

ID	Amenaza	Descripción
C-01	Almacenamiento interno de Android	La base de datos se guarda en almacenamiento interno, inaccesible por otras apps sin root.
C-02	Validación de entradas	Se valida que los campos introducidos cumplan el formato esperado.
C-03	Ausencia de funciones de exportación	La aplicación no permite exportar los datos ni acceder a ellos desde otras apps, lo que reduce el riesgo de fuga o exposición no deseada.

12. Identificación y análisis de soluciones posibles

12.1 Alternativas analizadas

Mi primera opción fue desarrollar MyFinance utilizando Kotlin y Jetpack Compose para la interfaz y SQLite para almacenar los datos localmente. Esto me asegura que mis datos no dependan de servicios externos para almacenarlos y me asegura la privacidad ellos.

La segunda opción que valoré fue desarrollar la aplicación en flutter ya que compila a código nativo tanto en Android como en iOS gracias a Dart. Sin embargo, al no conocer tanto este lenguaje y que el TFG será únicamente para Android, descarte esta opción.

12.2 Criterios de selección

Para decidir qué alternativa seguir en el desarrollo de MyFinance, se valoraron principalmente cuatro factores clave: la privacidad de los datos, el rendimiento de la aplicación, el enfoque del Trabajo de Fin de Grado y el tiempo disponible para su desarrollo.

La privacidad de los datos era un aspecto prioritario. Al elegir SQLite como base de datos local, se garantiza que los datos del usuario no salgan del dispositivo, eliminando así cualquier dependencia de servicios en la nube o terceros.

Otro criterio importante fue el enfoque del TFG. El desarrollo nativo en Android es coherente con los conocimientos adquiridos durante el ciclo formativo de Desarrollo de Aplicaciones Multiplataforma (DAM), lo que permite demostrar con claridad las competencias adquiridas en este ámbito.

Por último, el tiempo disponible para el desarrollo también influyó en la decisión. Elegí una tecnología que ya conocía (Kotlin y Jetpack Compose) y que había estudiado durante el curso porque me iba a resultar más cómodo que invertir tiempo en aprender un nuevo lenguaje como Dart, cuyo uso no era necesario para los objetivos del proyecto.

13. Solución propuesta

Esta sección detalla la alternativa final seleccionada **desarrollo nativo** y su plan de ejecución, dividiéndola en tres bloques clave: descripción general, fases de desarrollo e implementación y validación.

13.1 Descripción de la solución

Finalmente he decidido desarrollar la aplicación en Kotlin utilizando Jetpack Compose. Almacenaré los datos localmente utilizando Room (SQLite), y así garantizaré el funcionamiento de la aplicación sin conexión. El usuario podrá registrar transacciones, visualizar un historial con filtros para el tipo de transacción y gráficos con estadísticas. También incluye una calculadora donde los usuarios pueden hacer simulaciones de interés compuesto y una pantalla perfil donde registrar y modificar tanto un nombre de usuario como una foto de perfil, en esta pantalla también se podrá alternar entre modo oscuro y claro

13.2 Fases del desarrollo

Organicé el desarrollo en varias fases bien diferenciadas. Lo primero que hice fue planificar el diseño de la aplicación. Para ello, definí claramente las funcionalidades que debía ofrecer y realicé bocetos de cada pantalla, esto fue muy importante porque me ayudó a coger ideas y visualizar que quería hacer realmente.

El siguiente paso fue construir la base de la aplicación, en esta fase diseñé la estructura interna para guardar y recuperar datos de manera eficiente gracias a Room. También desarrollé los componentes compartidos entre todas las pantallas, es decir la cabecera (header) y la barra de navegación.

A continuación, y con los componentes comunes ya implementados, lo siguiente fue crear cada pantalla para poder moverme entre ellas y diseñé la pantalla de Inicio, para poder ver los datos inicialmente y poder mostrar el balance general, de ingresos y gastos y cuando ya tuve esto, implemente la pantalla de transacciones. Al implementar la pantalla de transacciones verifiqué que se actualizara correctamente el balance de la app e implementé la pantalla historial.

Con todo esto, añadí las pantallas de interés compuesto y perfil, estas pantallas no están relacionadas con el resto así que su implementación fue más fluida.

Finalmente, probé cada funcionalidad para asegurar que todo sea intuitivo y libre de errores, ajuste detalles de diseño e implemente el modo claro/oscuro

13.3 Implementación y validación

Durante la implantación de la aplicación, llevé a cabo varias pruebas, para la funcionalidad, verifiqué que todas las operaciones funcionaran correctamente, tanto los cambios entre pantallas como las funcionalidades de cada pantalla fueron probadas en múltiples escenarios para verificar que funcionaran correctamente.

Posteriormente realicé pruebas de usabilidad, para ello le instalé la aplicación a miembros de mi familia y comprobé como interactuaban con ella, me recomendaron algún visual y así hice. También comprobé que la aplicación tuviera un buen rendimiento y efectivamente tarda muy poco en responder ante cualquier cosa gracias al uso eficiente de Room.

Gracias a esto último, con Room tanto la privacidad de los datos como el funcionamiento de la app sin conexión es posible, permitiendo al usuario utilizarla en cualquier circunstancia.

14. Diseño de la solución

En esta sección describo cómo he estructurado y organizado la aplicación para garantizar claridad, escalabilidad y un mantenimiento fácil.

14.1 Arquitectura de capas

La aplicación se organiza en tres capas principales. La primera de ellas es la capa de datos, ésta capa se encarga del almacenamiento de la información. Para ello utilizo Room (SQLite), que me permite gestionar los datos de manera local sin necesidad de conexión a internet.

La segunda es la de negocio, ésta capa contiene la lógica de la aplicación. Está compuesta principalmente por ViewModels que actúan como intermediarios entre la base de datos y la interfaz. Por último, está la capa de presentación, ésta está diseñada con Jetpack Compose, cada pantalla de la app reacciona automáticamente a los cambios de datos.

14.2 Componentes reutilizables

Para evitar duplicar código, creé componentes comunes y reutilizables que utilizo en todas las pantallas. El primero de ellos es la cabecera (Header), esta es una barra superior que contiene el logo y el título de la aplicación y el nombre de la pantalla es la que estás. El segundo componente es la barra de navegación, ésta se encuentra en la parte de inferior de todas las pantallas de la aplicación y se encarga de permitir al usuario moverse entre las pantallas.

14.3 Flujo de navegación

Usé un único NavHost que gestiona las rutas, al pulsar en un icono de la barra de navegación, llama al NavController de NavHost y cambia de pantalla de forma fluida, manteniendo el estado de los datos gracias el ViewModel.

14.4 Gestión de estados y reactividad

Cada ViewModel expone su información mediante StateFlow. Cuando se añade una transacción, se cambia el tema de la aplicación o cualquier dato de la base de datos cambia, la interfaz se actualiza automáticamente con los datos correctos.

15. Diseño detallado

En esta sección menciono cómo estructuro internamente cada módulo y cómo interactúan sus componentes para ofrecer la funcionalidad esperada.

En el modelo de datos, tengo las transacciones, las categorías y el usuario. Cada transacción contiene los campos; id, tipo, monto, fecha, categoría y descripción. Por otro lado, las categorías también son entidades con su propio id, nombre y tipo. Por último, el usuario se almacena con su id, nombre y foto de perfil.

Los repositorios actúan como una capa intermedia entre los ViewModels y la base de datos. Éstos se encargan de proporcionar funciones para agregar o modificar datos. Los ViewModels, son el punto de conexión entre los datos y la interfaz, al cargar cada pantalla son los responsables de mostrar la información correctamente actualizada

La interfaz esta diseñada de manera independiente, cada pantalla tiene su propia estructura y cumplen cada una con sus tareas independientes. Todas cuentan con la misma barra de navegación y cabecera, ya que son componentes únicos para todas.

16. Tecnologías utilizadas

16.1 Entorno de desarrollo

He utilizado Android Studio como IDE de mi aplicación principalmente porque es el que hemos utilizado durante el curso y me siento muy cómodo con él, además cuenta con una integración nativa con Kotlin y Jetpack Compose.

16.2 Lenguaje de desarrollo

He desarrollado la aplicación en Kotlin por un motivo similar al del punto anterior, es el que hemos estado utilizando durante el año y me he sentido cómodo desarrollando con ello.

16.3 Sistema operativo

La aplicación está dirigida únicamente a dispositivos Android.

16.4 Frameworks y bibliotecas

Para diseñar la interfaz he utilizado Jetpack Compose, es un framework de Kotlin que me ha ayudado a crear la interfaz de forma reactiva, haciendo que, al haber un cambio en la base de datos, los cambios se reflejen al instante.

Para el almacenamiento he utilizado Room (SQLite), debido a su comodidad para mantener los datos de manera local.

16.5 Control de versiones

He trabajado con Github para el control de las versiones de mi aplicación, principalmente porque es la que siempre suelo usar y me resulta muy cómoda.

17. Desarrollo de solución propuesta

En este apartado, se detallan los elementos clave del desarrollo, los problemas y obstáculos hallados, además de las medidas adoptadas para superarlos y las especificidades de la resolución final:

17.1 Proceso de desarrollo

Dividí el proyecto en puntos semanales: cada semana elegía una pantalla o función (home, historial, formulario, calculadora, perfil), asegurando que no me quedaba estancado.

17.2 Problemas y dificultades

El mayor problema que encontré fue la sincronización de datos, me llevó mucho tiempo conseguir que al añadir una transacción se guardará bien y actualizara correctamente tanto el historial como los gráficos de estadísticas.

17.3 Decisiones tomadas

Separé la lógica de la gestión de datos y la interfaz para que cada parte tenga su responsabilidad clara y el código no se convierta en un enredo.

Para ello utilicé Room (SQLite) para que la base de datos local envíe automáticamente las actualizaciones.

17.4 Particularidades de la solución final

Al hacerlo de esta manera, me aseguro de que la aplicación es 100% offline, toda transacción, cambio en el perfil o cambio de tema de la aplicación, se guarda de manera local y siempre vuelve a aparecer igual incluso después de cerrar la aplicación.

18. Implantación

La etapa de implementación fue muy importante en el desarrollo de la aplicación, en ella pasé de verla en mi entorno de desarrollo a verla funcionar en un dispositivo real.

18.1 Instalación en dispositivos de prueba

Primero me descargué el APK desde Android Studio y lo instalé en el móvil de mi padre, un Samsung Galaxy. Así pude probar directamente la app en un dispositivo real y asegurarme de que todo funcionaba correctamente.

18.2 Pruebas en entorno real

En el móvil real probé a fondo todas las pantallas: desde la pantalla de inicio hasta el historial, el formulario de transacciones, la calculadora y el perfil. Verifiqué que cada vista se cargara sin errores, que los gráficos de ingresos, gastos y estadísticas se actualizarán al instante tras añadir transacciones, y que el desplazamiento y los filtros del historial respondieran con fluidez.

También comprobé que alternaba correctamente entre los modos claro y oscuro y edité el perfil varias veces, confirmando que esos ajustes se guardaran y reflejaran de inmediato, incluso sin conexión a Internet o tras cerrar y volver a abrir la aplicación.

18.3 Pantalla inicio

La pantalla inicio muestra claramente el balance actual de la cuenta, justo debajo, aparecen tres tarjetas más pequeñas que resumen los ingresos, gastos y el ahorro acumulado. Esta pantalla también cuenta con dos gráficos, el primero muestra la distribución de gastos y el segundo gráfico muestra la distribución de ingresos.

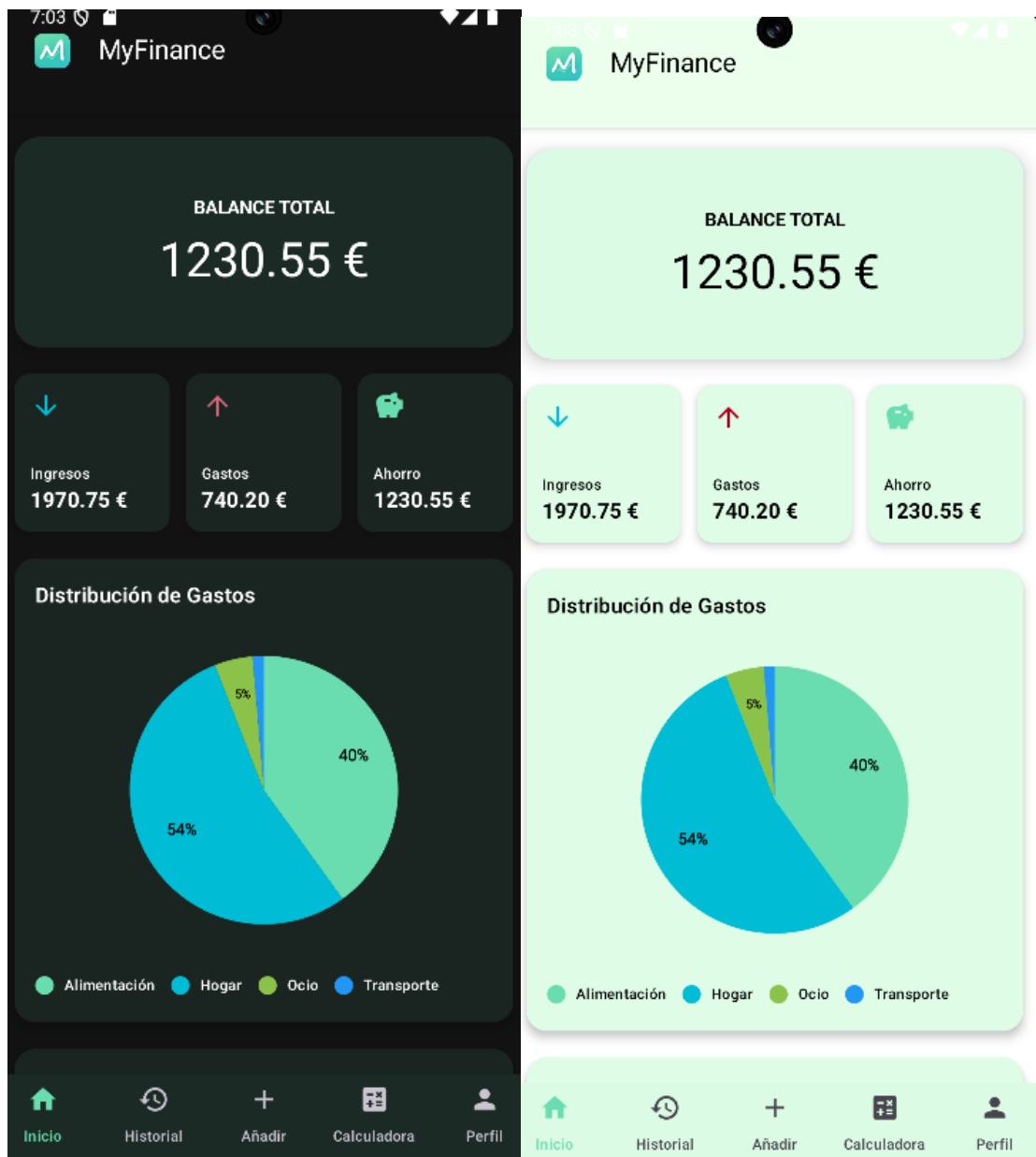


Figura 8 - Pantalla Inicio

18.4 Pantalla Historial

En la pantalla historial podemos visualizar rápidamente las últimas operaciones que hemos realizado de una manera muy sencilla. Cada operación esta diferenciada con el icono de la categoría a la que pertenece, también aparece el tipo de transacción que es, si es un gasto aparecerá en rojo y si es un ingreso en verde.

Esta pantalla también cuenta con los botones de “Todos”, “Ingresos” y “Gasto”, éstos sirven para filtrar al instante la lista y centrarnos solo en los movimientos que nos interesen.

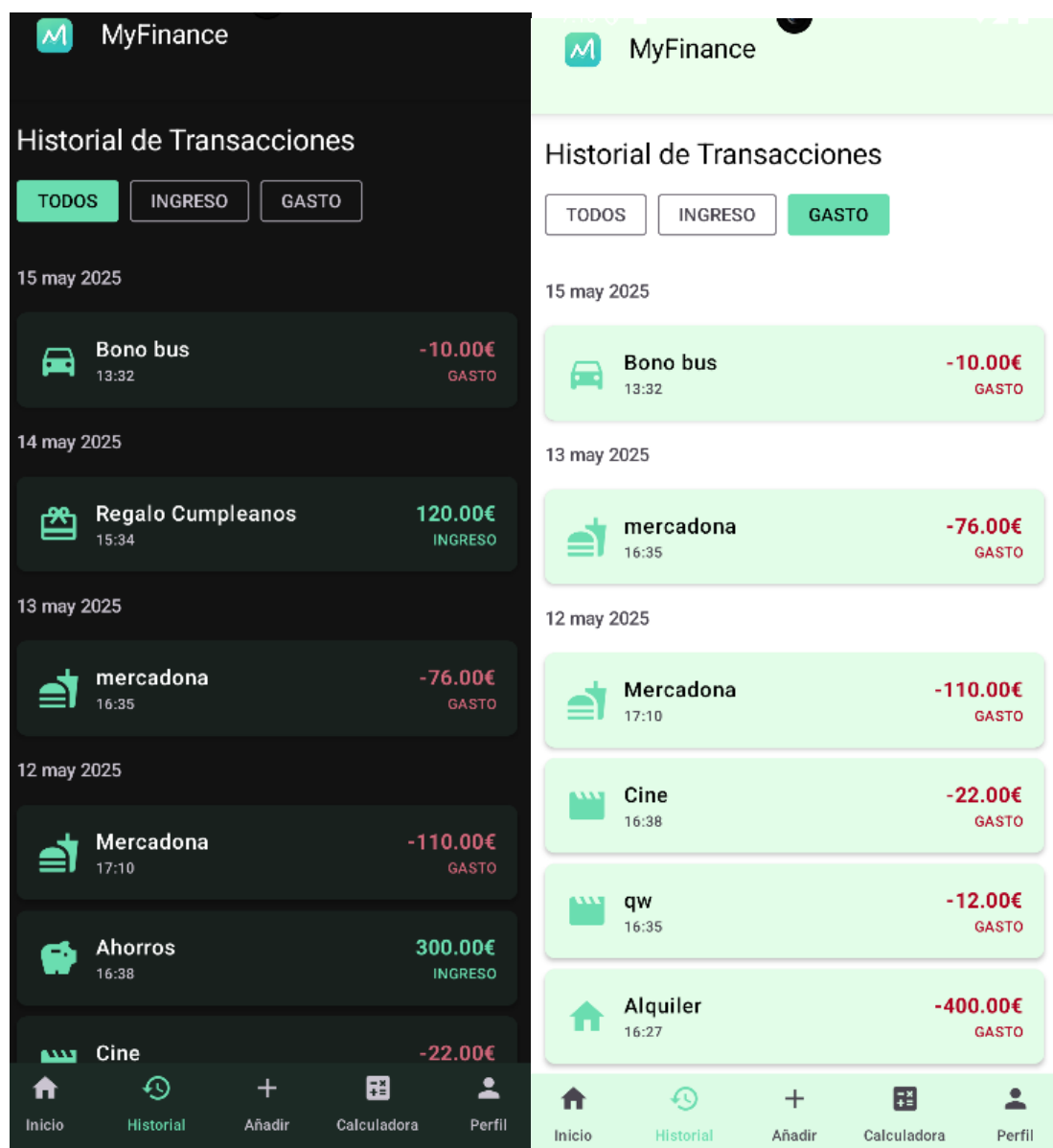


Figura 9 - Pantalla Historial

18.5 Pantalla Transacciones

En la pantalla de transacciones podemos registrar cualquier ingreso o gasto de forma rápida e intuitiva. Primero debemos elegir entre el tipo de transacción tocando en “Gasto” o “Ingreso”, al elegir uno, alternan las categorías y por ello también los iconos que aparecen. Después, podremos darle una breve descripción a la transacción, para así diferenciarla en el historial. El siguiente paso será elegir una categoría presionando en el icono que queramos y por último indicar el monto total de la transacción y así poder añadirla

Todos los campos deben estar rellenos antes de añadir la transacción, si no fuera así, la aplicación no te permite añadirla ya que el botón de “Añadir transacción” aparecerá deshabilitado, y solo lo habilitará cuando todos los campos estén rellenos.

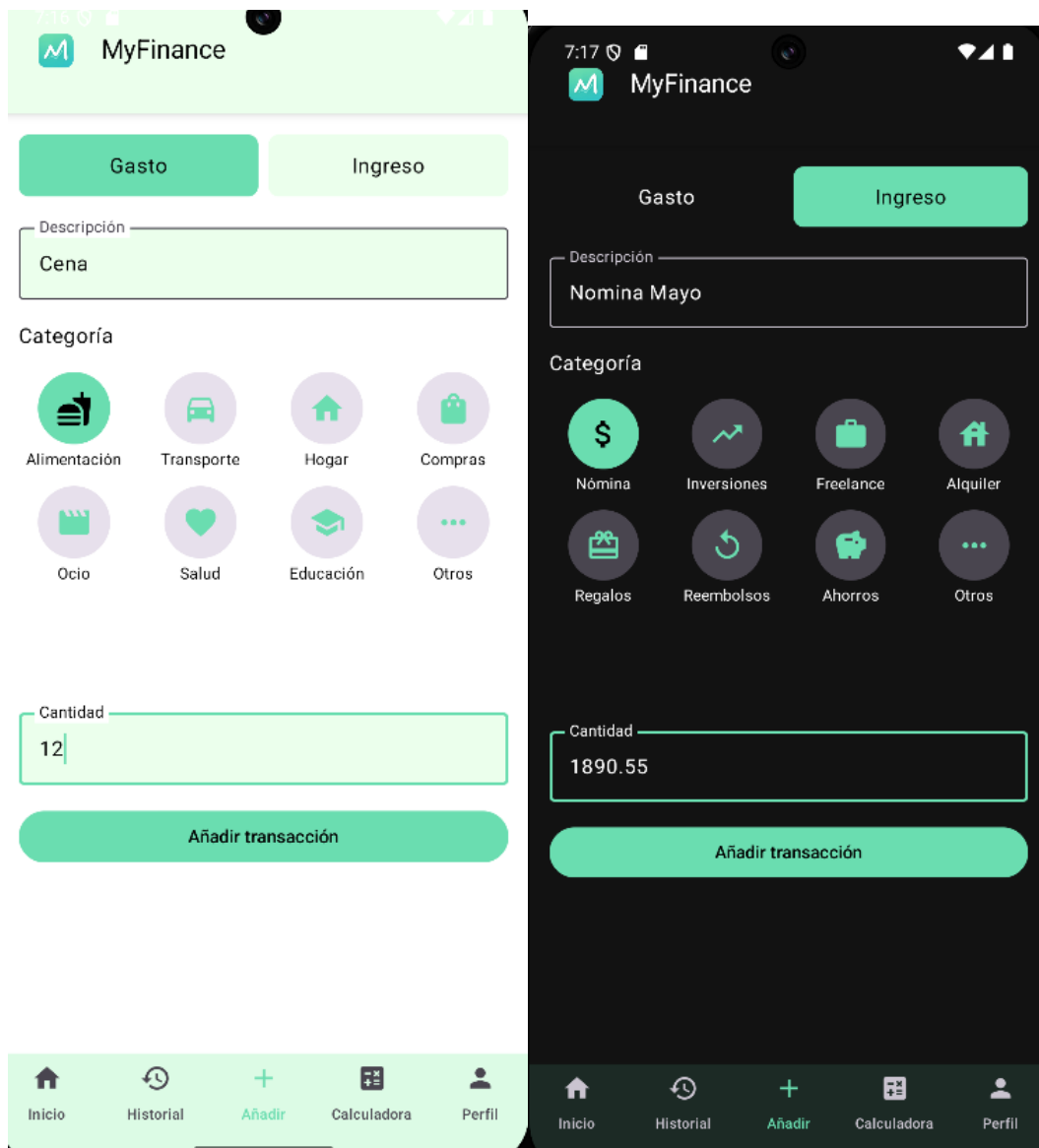


Figura 10 - Pantalla Transacción

18.6 Pantalla calculadora

En la pantalla de la calculadora de interés compuesto, podemos introducir nuestro saldo inicial, el importe que queremos aportar cada mes, la tasa anual y el número de años, y al pulsar el botón de “Calcular”, obtenemos al momento el total acumulado y un desglose claro de cuánto proviene del capital inicial, de los depósitos periódicos y de los intereses generados. El resultado aparece en una tarjeta que incluye un gráfico en el que podemos diferenciar de donde vendría el dinero final claramente. También cuenta con botón de “Restablecer” para limpiar los campos y probar nuevos valores sin perder la fluidez de la experiencia.

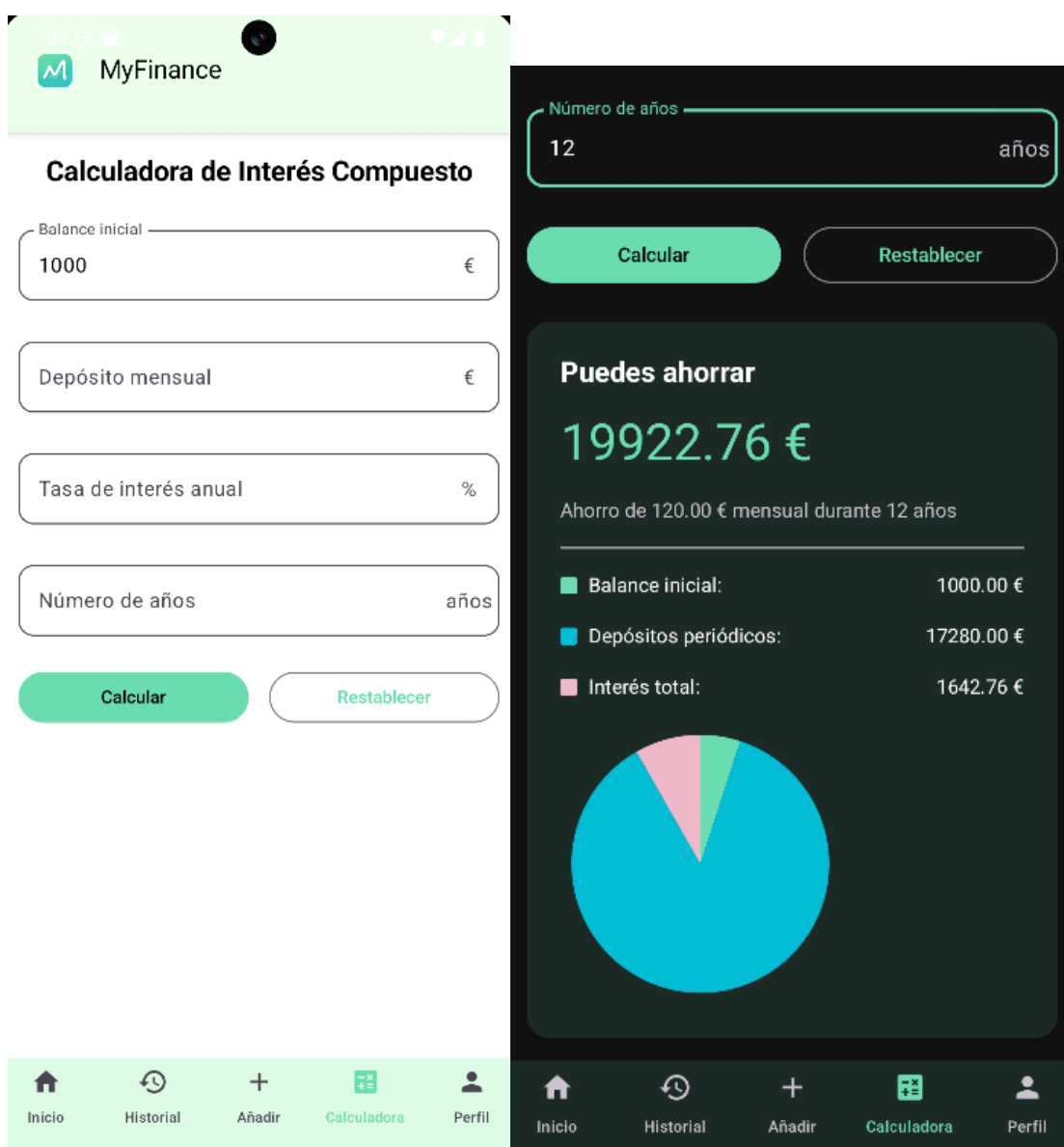


Figura 11 - Pantalla Calculadora

18.7 Pantalla perfil

En la pantalla de Perfil puedo ver mi avatar y mi nombre en grande, junto a un botón “Editar perfil” que me permite cambiar mi nombre y subir una foto nueva de forma muy sencilla. Debajo, hay dos botones con iconos de sol y luna para alternar entre modo claro y oscuro; al tocar uno, el tema de toda la app cambia al instante y se guarda mi preferencia.

Al pulsar “Editar perfil” se despliega un campo de texto donde puedo modificar mi nombre y, si lo deseo, elegir otra imagen para mi avatar. Tras escribir el nuevo nombre o seleccionar una foto, doy a “Guardar cambios” y vuelvo automáticamente a la vista principal de Perfil con los datos actualizados. Todo permanece sincronizado incluso si cierro y vuelvo a abrir la app, gracias al almacenamiento local.

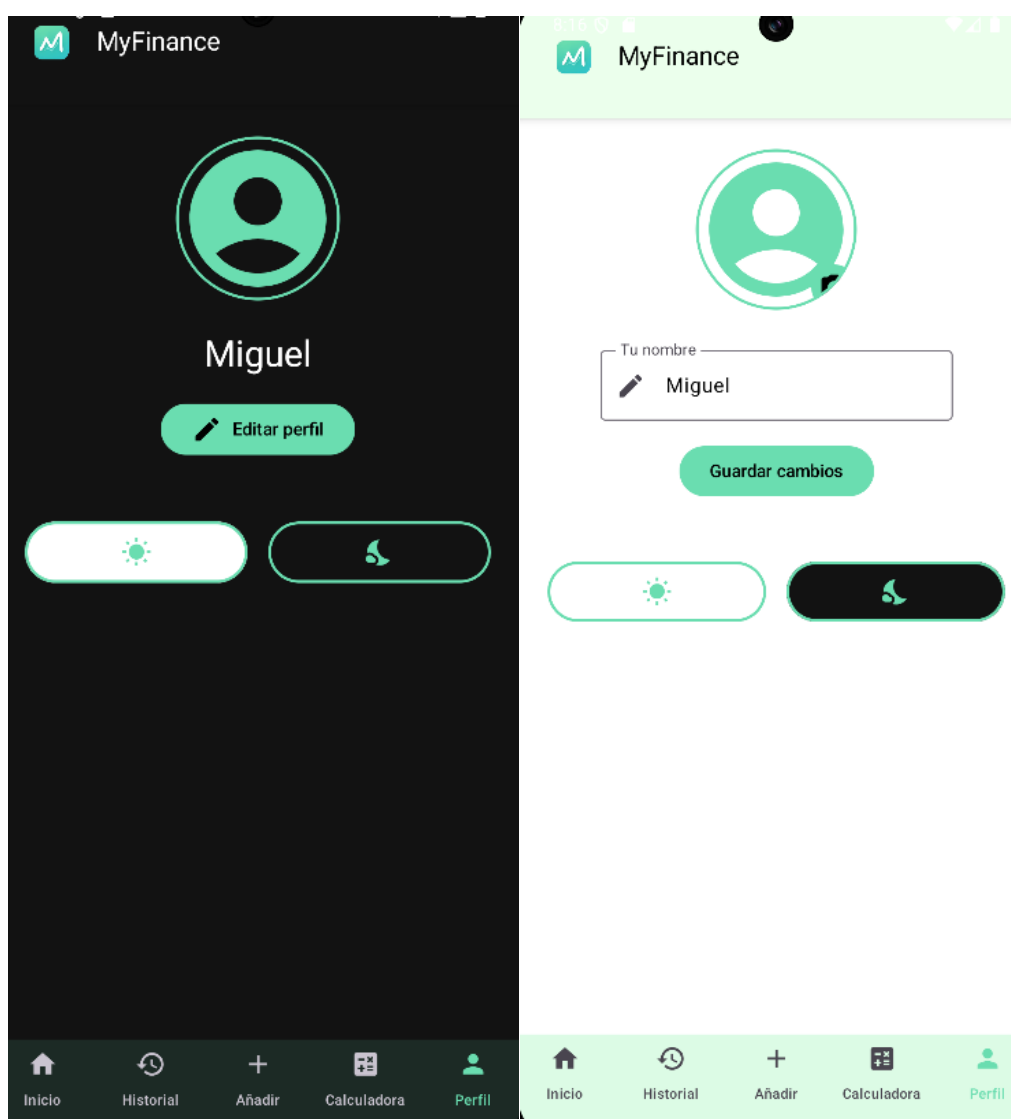


Figura 12 - Pantalla Perfil

19. Plan de pruebas

En este apartado se detalla el plan de pruebas llevado a cabo para asegurar que la aplicación funcione correctamente y cumpla con las expectativas técnicas. Las pruebas son una fase crítica del desarrollo, ya que permiten identificar y corregir errores, mejorar la eficiencia y garantizar que el producto final esté listo para su despliegue en un entorno de producción.

A continuación, se presenta una tabla con las pruebas realizadas, detallando las condiciones iniciales y los resultados obtenidos. Cada prueba se diseñó y ejecutó meticulosamente para asegurar que MyFinance no solo funcione correctamente, sino que también proporcione una experiencia satisfactoria y eficiente a los usuarios finales.

PRUEBA	DESCRIPCIÓN	CONDICIONES	RESULTADO
P1	Conexión de la app con la base de datos local	App instalada en dispositivo/emulador sin datos previos	Se inicia sin errores
P2	Carga del balance total al iniciar la aplicación	Base de datos con transacciones existentes	Se muestra el total correcto en las tarjetas de inicio
P3	Añadir una transacción	Formulario completo (tipo, categoría, descripción, monto, fecha)	La transacción se guarda en la base de datos
P4	Actualización historial tras añadir una nueva transacción	Haber ejecutado P3; pantalla Historial abierta	La nueva transacción aparece al inicio de la lista
P5	Actualización del balance tras añadir una nueva transacción	Haber ejecutado P3; pantalla Inicio visible	El balance total refleja el cambio inmediatamente
P6	Actualización gráficos tras añadir una nueva transacción	Haber ejecutado P3; gráfico de pastel en pantalla Inicio	El gráfico incorpora la nueva categoría y porcentaje actualizado
P7	Filtrar historial por gastos o ingresos	Historial con múltiples transacciones de ambos tipos	Al seleccionar “Gasto” o “Ingreso” solo se muestran los correspondientes

P8	Funcionamiento offline	Desactivar conexión de red y reiniciar la app	Todas las pantallas funcionan y muestran datos locales sin error
P9	Cambio tema claro/oscuro en todas las pantallas	Tema actual claro u oscuro; app en cualquier sección	La UI cambia de tema al instante en todas las pantallas
P10	No permitir añadir transacción sin rellenar todos los campos	Formulario con uno o más campos vacíos	El botón “Añadir” permanece deshabilitado y muestra validación
P11	Funcionamiento calculadora interés compuesto	Campos de la calculadora completos con valores válidos	Se muestra el total acumulado y desglose correctos
P12	Reestablecer campos en la calculadora de interés compuesto	Haber realizado un cálculo previo	Al pulsar “Restablecer” todos los campos vuelven a estar en blanco
P13	Permitir actualizar foto de perfil	Pantalla Perfil abierta; avatar seleccionado	Se carga un selector de imágenes y el nuevo avatar se muestra al guardar
P14	Permitir actualizar nombre de usuario	Pantalla Perfil en modo edición; campo nombre editable	El nombre cambia instantáneamente y persiste al reiniciar la app

20. Conclusiones

La realización de MyFinance me ha permitido consolidar una aplicación Android nativa enfocada en la gestión de finanzas personales completamente offline, integrando de manera coherente funciones de registro de transacciones, visualización de estadísticas y cálculo de ahorro con interés compuesto. El objetivo principal es ofrecer una herramienta clara y fiable para usuarios sin depender de servicios en la nube y creo que se ha cumplido satisfactoriamente, con una interfaz en modos claro y oscuro.

Este trabajo refuerza mi capacidad para llevar una idea desde el boceto hasta un producto funcional, equilibrando diseño, rendimiento y usabilidad.

21. Relación con los estudios cursados

21.1 Base de datos

El empleo de Room (SQLite) para el almacenamiento y recuperación de información es una aplicación directa de lo visto en los módulos de bases de datos y acceso a datos de DAM. La gestión de datos en local, junto con la persistencia de preferencias de usuario, refleja el dominio de conceptos como la persistencia de datos, las transacciones seguras y la optimización de consultas que adquirí durante los estudios.

21.2 Programación

La estructura de MyFinance se implementó con Kotlin en Android Studio. La organización de la lógica de negocio y el diseño del código aplican los conceptos de programación orientada a objetos y los principios de modularidad y separación de responsabilidades que estudié en las asignaturas de programación.

21.3 Desarrollo de interfaces

El diseño de la interfaz con Jetpack Compose y Material3 refleja directamente lo aprendido en la asignatura de desarrollo de interfaces de DAM. Gracias a ello, pude aplicar principios de usabilidad y accesibilidad para crear componentes reutilizables (tarjetas, botones, barras de navegación) y garantizar una experiencia visual coherente y sencilla de usar, tanto en modo claro como oscuro.

22. Referencias

Google. (n.d.). *Android Lifecycle: LiveData and ViewModel*.

<https://developer.android.com/topic/libraries/architecture/lifecycle>

Google. (n.d.). *Guía de arquitectura para apps Android (MVVM y capas)*.

<https://developer.android.com/topic/architecture>

Google. (n.d.). *Jetpack Compose documentation*.

<https://developer.android.com/jetpack/compose>

Google. (n.d.). *Material Design*. <https://material.io/design>

Google. (n.d.). *Material Design 3 guidelines*. <https://m3.material.io>

Google. (n.d.). *Room Persistence Library documentation*.

<https://developer.android.com/topic/libraries/architecture/room>

Google. (n.d.). *Android Studio releases (Meerkat | 2024.3.2)*.

<https://developer.android.com/studio/releases>

JetBrains. (n.d.). *Kotlin documentation*. <https://kotlinlang.org/docs>

Android Developers. (n.d.-a). *Documentación oficial de Android*.

<https://developer.android.com/docs>

Android Developers. (n.d.-b). *Navigation component*.

<https://developer.android.com/guide/navigation>

Jahn, P. (n.d.). *MPAndroidChart: A powerful Android chart library*. GitHub.

<https://github.com/PhilJay/MPAndroidChart>

23. Anexos

23.1 Estructura del proyecto

23.1.1 Módulo raíz

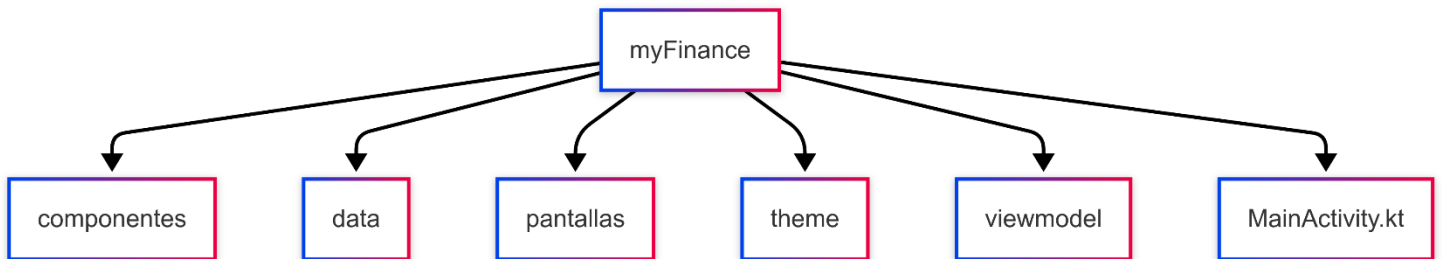


Figura 13 - Módulo Raíz

23.1.2 Módulo componentes

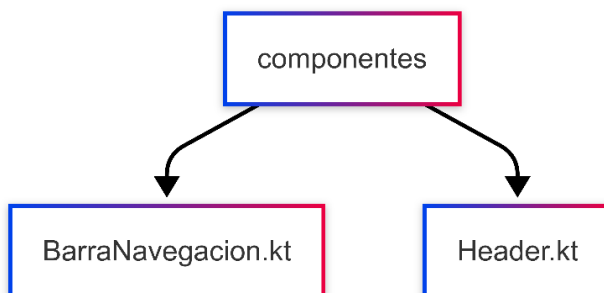


Figura 14 - Módulo components

23.1.3 Módulo data

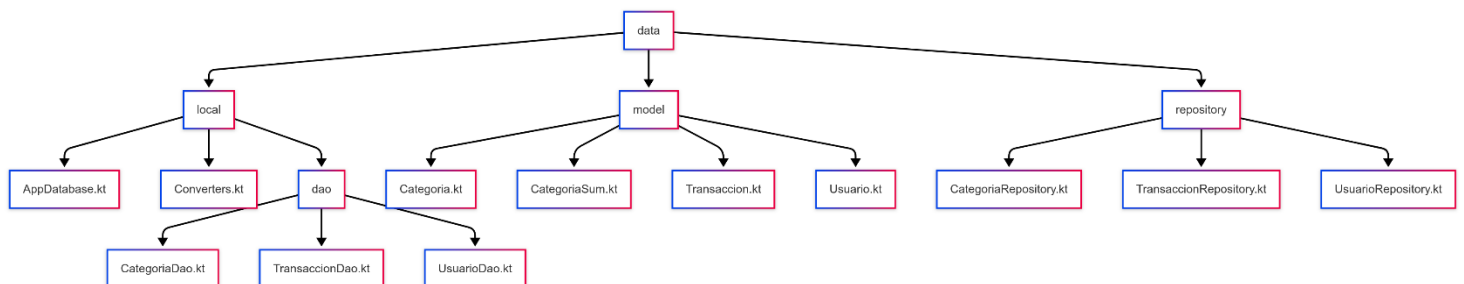


Figura 15 - Módulo data

23.1.4 Módulo pantallas

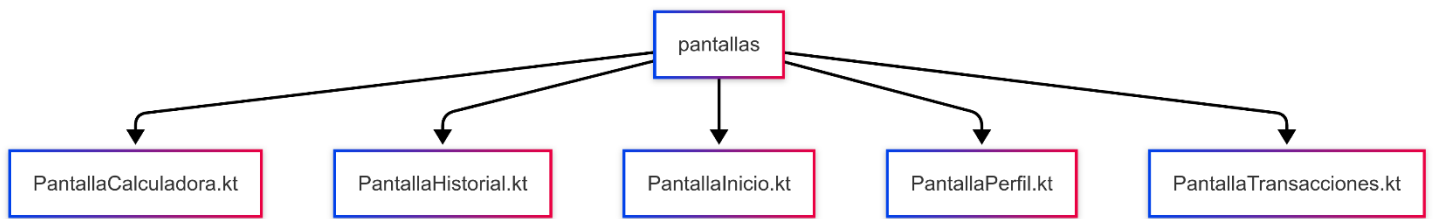


Figura 16 - Módulo pantallas

23.1.5 Módulo theme

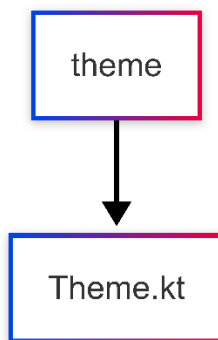


Figura 17 - Módulo theme

23.1.6 Módulo viewmodel

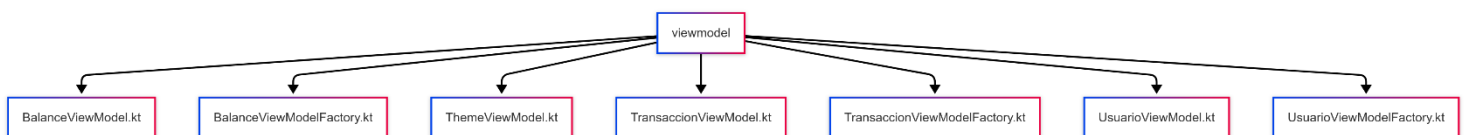


Figura 18 - Módulo viewmodel

23.2 Manual de usuario

23.2.1 ¿Qué es MyFinance?

MyFinance es una app para Android que asiste al usuario en la administración de sus finanzas personales de forma sencilla e intuitiva. El objetivo principal de la aplicación es facilitar el registro y clasificación de transacciones, ayudando al usuario a mantener un control total de sus finanzas.

La aplicación se ha creado en Kotlin usando Jetpack Compose, y solo es válida en dispositivos Android. La utilización de Room (SQLite) permite el almacenamiento local de todos los datos.

23.2.2 Requisitos mínimos

Para poder instalar y ejecutar MyFinance el dispositivo debe tener al menos 100 MB disponibles para la base de datos local y ser un dispositivo con sistema operativo Android 8.0 o superior.

23.2.3 Instalación

Para instalar correctamente MyFinance puedes descargar el apk desde la web o desde github. En el dispositivo ve a ajustes y luego seguridad, allí activa “Fuentes desconocidas”. Elige el apk de MyFinance y sigue al asistente. Una vez instalada, abre MyFinance y empieza a utilizar la aplicación.

23.2.4 Primero pasos

Al iniciar MyFinance por primera vez, se accede directamente a la pantalla de inicio, allí podremos ver las tarjetas que indican el balance actual de la cuenta y el total de ingresos y gastos. Obviamente todos los datos estarán iniciados a 0, al estar la base de datos vacía, ahora puedes empezar a añadir transacciones y disfrutar de la aplicación.

23.2.5 Navegación general

La navegación de la aplicación está organizada en una barra de navegación que se encuentra en la parte inferior de ella, esta barra de navegación permite al usuario moverse entre las pantallas de la aplicación.

23.2.6 Uso de la aplicación

Añadir transacciones

Para añadir transacciones, el usuario deberá ir a la pantalla de transacciones, una vez allí, deberá rellenar todos los campos requeridos; descripción, tipo, categoría y un monto. Si no están todos los campos rellenos el botón de guardar estará deshabilitado, una vez el usuario rellene los campos este botón se habilitará.

Consultar historial

El usuario podrá consultar el historial de transacciones desde la pantalla del historial, allí se encuentran todas las transacciones ordenadas por fecha, de más reciente a más antigua. Las transacciones indican con un icono la categoría a la que pertenecen, el usuario podrá filtrar por el tipo de transacción: ingresos, gastos o todas.

Visualizar estadísticas

En la pantalla de inicio aparecen varios gráficos en los que el usuario puede analizar la proporción de gastos e ingresos por categorías. Esta información se actualiza automáticamente al registrar una nueva transacción.

Simular ahorros con interés compuesto

La aplicación también cuenta con una calculadora de interés compuesto en la que el usuario puede simular escenarios de cuánto dinero podría ahorrar en un periodo determinado. El usuario deberá rellenar los campos antes de realizar el cálculo y una vez completado, aparecerá un gráfico con el crecimiento estimado de la cuenta.

Editar perfil y modo visual

En la pantalla de perfil, el usuario puede cambiar su nombre y su foto de perfil. Además, esta pantalla cuenta con dos botones que permiten alternar entre el modo oscuro y el modo claro.

23.2.7 Seguridad y privacidad

MyFinance está diseñada para almacenar los datos localmente utilizando Room (SQLite), la aplicación no realiza copias en la nube ni accede a servicios externos protegiendo al usuario de fugas o pérdidas de información a través de terceros. No se requieren permisos especiales y la app puede utilizarse perfectamente sin conexión a internet.

Buenas prácticas de uso

Se recomienda revisar el historial para mantener el control de los movimientos. Para simulaciones con la calculadora, es aconsejable guardar los resultados manualmente, ya que la aplicación no guarda estos registros.

23.2.8 Preguntas frecuentes (FAQ)

¿Necesito conexión a internet para usar MyFinance?

La respuesta es **no**, la aplicación funciona completamente offline.

¿Dónde se guardan mis datos?

Toda la información se almacena en la memoria del dispositivo.

¿Qué ocurre si desinstalo la aplicación?

Toda la información registrada se perderá, ya que no existe copia de seguridad.

23.2.9 Contacto

Correo: miguel.reyesgomez1@gmail.com

Repositorio público de Github: <https://github.com/miguelreyess1/TFG-GS-DAM>