Aplicación para el seguimiento de la gestión financiera

Presentado por:

- Israel Hernández
- Daniela Pozo
- Gabriel Vásconez
- Michael (Melissa) Molina

Ciudad: Quito
Fecha: 13/07/2025

Resumen

En este trabajo se desarrolló una aplicación móvil para control de gastos llamada Kaputt, diseñada para apoyar a estudiantes universitarios en la gestión de sus finanzas personales. El objetivo principal fue facilitar el registro de ingresos y gastos, el cálculo automático del balance financiero, la visualización gráfica de gastos, ingresos y la comparativa con el balance llamada “general” además de la promoción de hábitos de ahorro a través de metas y retos financieros. La metodología que se utilizó fue un modelo ágil: Kanban y control de versiones en GitHub. Se realizaron entrevistas y levantamiento de requerimientos para adaptar la aplicación a las necesidades reales del usuario, donde lograremos una aplicación funcional, intuitiva y adaptada a estudiantes universitarios, que no solo ayuda a administrar los recursos económicos, sino que también fomenta buenos hábito. Así, Kaputt se convierte en una herramienta efectiva para mejorar la organización financiera en estudiantes.

Índice de contenidos

Índice de tablas

Índice de figuras

1. Introducción

La gestión financiera es un aspecto básico en la vida que empieza a aprenderse conforme un individuo empieza a independizarse, recibiendo un ingreso monetario fijo y administrándolo en sus propios gastos personales. Dentro de estos gastos se encuentra, alimentación diaria, alquiler de vivienda, transporte y en el caso de un estudiante materiales necesarios. Conforme el gasto se vuelve más grande, también lo hace la necesidad de tener una herramienta que funcione como un apoyo gestor para los mismos.

Frente a esto la tecnología nos presenta soluciones que pueden contribuir no solo a entender los gastos, sino también aprender de educación financiera, es por ello que nuestro equipo se ha propuesto a desarrollar una aplicación gestora de recursos para los estudiantes a través de nuestra visión y perspectiva como estudiantes de la universidad católica, pues somos capaces de entender que al tener una vida ajetreada es más sencillo el organizarse mediante el uso de aplicaciones.

1.1 Justificación

El proceso de independizarse, sobre todo financieramente, coincide en muchas ocasiones con el ingreso a la universidad, donde es el propio estudiante quien maneja sus gastos, en transporte, alimentación, gastos sociales, entre muchos más. A pesar de esto muchos carecen de la formación básica necesaria para afrontar este reto, lo que puede desembocar en desorganización e imposibilidad de cubrir gastos esenciales durante el mes.

A pesar de que existen aplicaciones destinadas a la administración financiera, ninguna esta enfocada a la gestión financiera de estudiantes, volviéndose complejas, poco intuitivas e ineficientes a la hora de cumplir su propósito. Por ello el desarrollo de una aplicación enfocada principalmente en estudiantes, representa un reto imperativo a ser resuelto.

1.2 Planteamiento del trabajo

El proyecto pretende ofrecer, además de una solución tecnológica funcional e intuitiva, la capacidad de fomentar hábitos financieros saludables que impacten positivamente la vida académica del estudiante. Para lograr esto se desarrollará una aplicación móvil en Android con enfoques en los movimientos económicos diarios, el establecimiento de un

presupuesto mensual o semanal, visualización de gráficos y reportes que permitan al usuario comprender mejor su comportamiento financiero.

2. Contexto y estado del arte

-Introducción

En la actualidad el desarrollo de aplicaciones de gestión financiera ha cobrado gran importancia debido al constante y creciente interés que presentan las personas por tener un mejor control sobre sus gastos y ser capaces de sobrellevar de mejor manera las condiciones del mundo actual. Esta situación se vuelve todavía mas intensa en estudiantes universitarios, que constituyen un grupo mas vulnerable a cometer gastos irracionales que comprometan su situación financiera al empezar, muchas veces, a hacerse cargo de sus propios gastos personales.

No obstante, muchas de las aplicaciones financieras del mercado no están diseñadas para manejar los ingresos variables de un estudiante, así como sus distintas necesidades específicas. Por ello resulta indispensable el desarrollar una aplicación capaz de solucionar estos problemas mediante el uso de interfaces claras y amigables con el usuario y funciones útiles para nosotros, como estudiantes universitarios.

-Desarrollo

Entre las aplicaciones mas prominentes en el mercado se encuentran Mobills, Mint, Wally, MoneyBox y Tricount, de las cuales compararemos sus funcionalidades con las propuestas para nuestra propia aplicación

Funcionalidad	**Nuestra App**	**Mobills**	**Mint**	**Wally**	**Moneybox**	**Tricount**
Registro de ingresos y egresos	Si	Si	Si	Si	No	Si
Gráficas de análisis	Si	Si	Si	Si	No	No
Identificación del mayor gasto individual	Si	No	No	No	No	No
Área de mayor gasto	Si	Si	Si	Si	No	No
Función de ahorro con metas	Si	Si	No	No	Si	No
Retos financieros motivacionales	Si	No	No	No	No	No
Registro por categorías	Si	Si	Si	Si	No	No
Descripciones personalizadas	Si	Si	No	Si	No	No
Interfaz amigable para estudiantes	Si	No	No	Si	Si	Si

Como se puede observar nuestra aplicación tiene un conjunto de funcionalidades que le pueden resultar útiles al estudiante universitario, no solo para manjar sus gastos, sino también para aprender sobre educación financiera.

-Conclusión

Las herramientas que encontramos para la creación, organización e independencia financiera son varias, pero cada una tiene algo diferente con propósitos de por medio dirigidos a una población en específico. Encontramos apps como Mobills, Mint, Wally, Moneybox y Tricount las cuales no se adaptan a las necesidades específicas de estudiantes universitarios, quienes necesitan soluciones un poco más simples y enfocadas a la educación para fomentar sus hábitos financieros.

En la comparación realizada hallamos que nuestra aplicación comparte características de otras para la gestión eficiente de gastos e ingresos. Tenemos registros de cada uno de ellos, categorización y análisis mediante los gráficos, Sin embargo, también incorporamos elementos motivacionales y retos financieros que no se encuentran comúnmente en las aplicaciones analizadas. Se consideró estos aspectos fundamentales para una mayor motivación y compromiso al organizarse con la aplicación buscando integrar características simples que manejen adecuadamente el desempeño responsable de los recursos económicos por parte de los estudiantes.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Creación de una app móvil enfocada en ayudar a los usuarios –principalmente estudiantes– a administrar los gastos que pueden realizar, ya sea en plazos semanales o mensuales, por medio de interfaces intuitivas a las que se puedan adaptar fácilmente, donde se mostrará su información sobre sus gastos realizados de manera visual y fácil de comprender.

3.2. Objetivos específicos

A) Enseñar sobre gestión financiera, especialmente a los estudiantes universitarios que esta empezando a manejar sus recursos monetarios por su cuenta, por medio de herramientas fáciles de entender y utilizar.
B) Dar flexibilidad al usuario sobre los tipos de gastos que puede documentar, teniendo una variedad de opciones apropiadas ya acertadas; como lo sería para gastos del hogar, transporte, alimentación, salud física, salud mental, entre otros.
C) Implementar diferentes tipos de gráficos intuitivos y amigables para el usuario, donde el mismo pueda reconocer y analizar los gatos que ha realizado, mostrando de manera clara aquellas áreas en donde los gastos han sido mayores

3.2. Metodología del trabajo

Kanban es una metodología ágil de gestión de proyectos y tareas que permite visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia (o *flujo*). Su nombre viene del japonés "kan" (visual) y "ban" (tarjeta o señal), es decir: tarjeta visual.

Principios básicos de Kanban

1. **Visualiza el trabajo**

   Usa un tablero dividido en columnas que representan diferentes etapas del proceso (por ejemplo: *Por hacer*, *En progreso*, *Hecho*).

   Cada tarea es una tarjeta que se mueve entre las columnas.

2. **Limita el trabajo en curso (WIP)**

   Se establece un número máximo de tareas que pueden estar en una columna.

   Esto evita la multitarea excesiva y mejora la concentración.

3. **Gestiona el flujo**

   Se observa cómo fluye el trabajo desde el inicio hasta el final, buscando cuellos de botella y mejoras.

4. **Políticas de proceso explícitas**

   Las reglas deben ser claras. Por ejemplo: "Solo puede haber 3 tareas en progreso".

5. **Usa bucles de retroalimentación**

   Reuniones regulares y análisis del flujo de trabajo ayudan a mejorar continuamente.

6. **Mejora continuamente**

   Se parte del proceso actual y se busca cómo mejorar de forma evolutiva y colaborativa.

4. Desarrollo específico de la contribución

4.1. Tipo 2. Planificación de un proyecto de desarrollo de software

4.1.1. Identificación de requisitos

Para la identificación de requisitos funcionales y no funcionales del sistema se siguió el siguió un proceso que incluyó la selección de tecnologías adecuadas, organización del equipo para un desarrollo eficiente y la implementación de mecanismos de evaluación continua.

4.1.1.1 Tecnologías utilizadas durante el proceso de desarrollo

- Android Studio: Para el entorno principal de desarrollo usamos Android Studio, por su compatibilidad con dispositivos con este sistema de desarrollo, permitir la simulación en emuladores integrados sin necesidad adicional, permitir el trabajo con java y su diseñador intuitivo.
- PowerDesigner: PowerDesigner es un software que permite el modelado de bases de datos usando diferentes tipos de notación, en nuestro caso usamos

Baker, la herramienta permite hacer tanto el modelado conceptual, lógico y físico de la base de datos, en ocasiones incluso proporcionando el script necesario para el levantamiento de las bases de datos, sin embargo en nuestro trabajo nos limitamos hasta el modelado lógico.

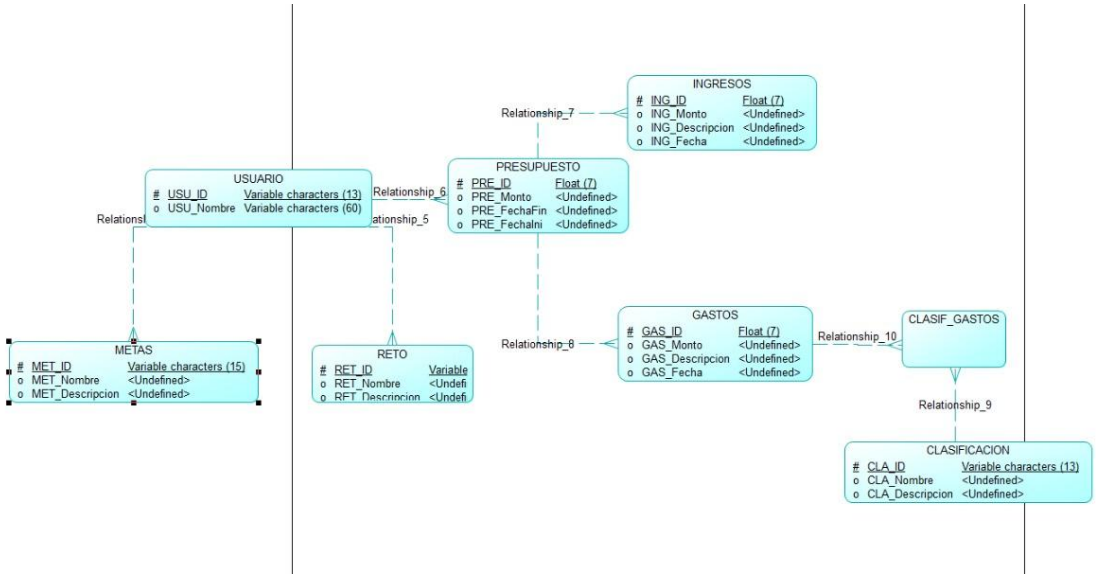


Figura 1. Modelo Entidad-Relacion

- Java y SQLite: Ambas alternativas se usaron por su integración nativa con Android, mientras Java nos permite hacer una codificación robusta y escalable, SQLite nos garantiza un almacenamiento local eficiente sin la necesidad de conexión a internet
- Github: Es una plataforma en donde los miembros de equipo tras hacer una cuenta pueden acceder a todos los documentos dentro de un repositorio comun, además cuenta con la opción de manejar proyectos ayudándonos por un roadmap y un tablero Kanban

4.1.1.2 Organización del proceso de desarrollo

El uso de la metodología Kanban permitió una gestión visual y flexible de las tareas con la implementación de un tablero (Figura 2) dividido en columnas. Backlog, siendo tareas que debemos hacer, pero que todavía no podemos iniciar, en “Ready” se encuentran las tareas que podemos empezar a hacer, en “In progress” encontramos las tareas que cada miembro esta realizando actualmente, al terminar una tarea esta se mueve finalmente a la columna Done, en donde se encuentran todas las tareas

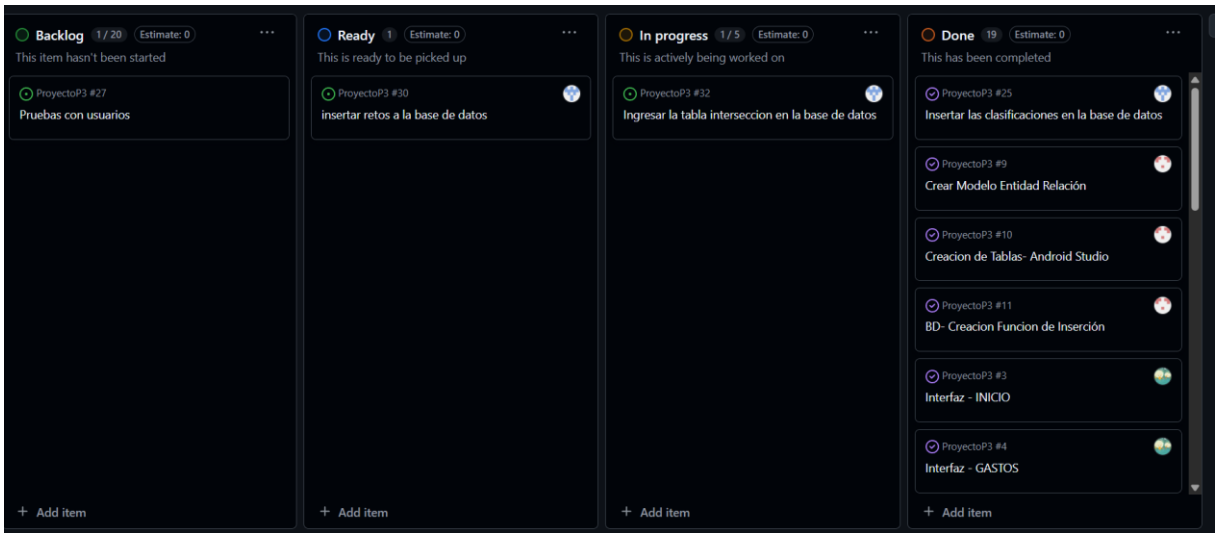


Figura 2. Tablero Kanban

Conforme fuimos realizando tareas, la retroalimentación en partes faltantes o mejoras en ciertos aspectos como fue por ejemplo la base de datos, se hicieron de forma eficiente y ágil manteniendo un flujo de trabajo constante y estable

4.1.1.2 Participantes y técnicas de levantamiento

Para la recopilación de información y así entender los requerimientos necesarios se hicieron consultas a estudiantes universitarios de distintas universidades entre las que están la Pontificia Universidad Catolica del Ecuador, la Universidad Politécnica Nacional, la Universidad Politecnica Salesiana y la Universidad Catolica del Ecuador, entre los 18 y 25 años.

Para ello se usaron entrevistas semiestructuradas en donde se hizo énfasis en hábitos financieros, gastos principales y facilidad con la cual se desenvuelven económicamente en su vida diaria y si sufren de problemas de manejo monetario o no.

4.1.1.3 Planificación del proceso

Durante el proceso de desarrollo de la aplicación, se utilizó GitHub como el principal instrumento de seguimiento y evaluación del proyecto. Esta plataforma nos permitió no solo trabajar con la metodología Kanban, que consiste en un modelo que nos proporciona un seguimiento continuo en cada etapa de nuestro proyecto, desde los requerimientos, hasta que se hicieron pruebas del proyecto, sino que también incorporamos la lista de actividades, tablero de prioridad, y un roadmap para cronogramar las actividades.

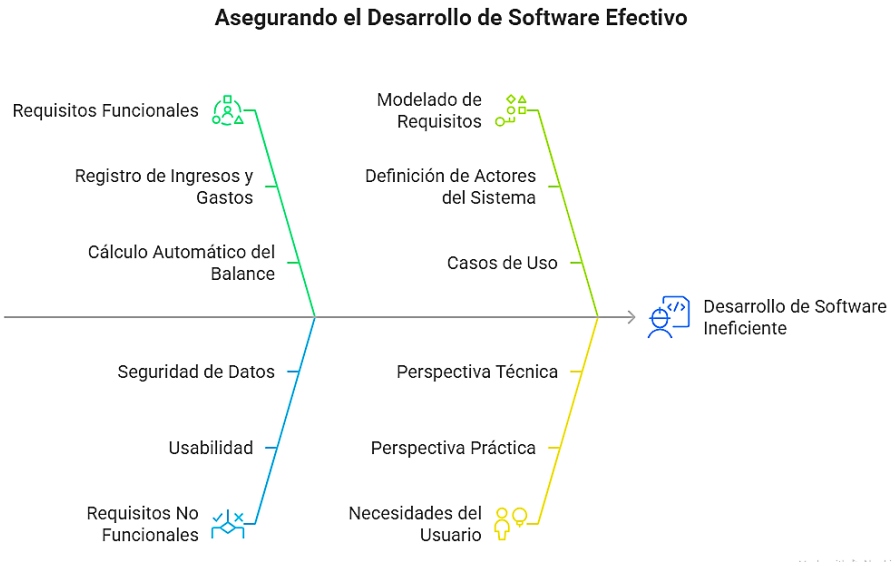
También nos permitió gestionar el control de versiones, organizar el trabajo mediante ramas (branches) para el desarrollo de nuevas funcionalidades como al hacer nuevas

interfaces, o añadir pequeños fragmentos útiles para las funcionalidades, y llevar un historial detallado de los cambios realizados en todo este tiempo.
A través de commits, se registraron avances y se documentaron correcciones, lo que facilitó el desarrollo. Además, se emplearon funcionalidades como los Issues y los Proyectos (Projects) de GitHub para anotar todas las tareas pendientes que asignamos al principio, asignar prioridades y hacer seguimiento del estado de cada componente según la metodología.
Esta herramienta también fue clave para detectar y corregir errores, colaborar en grupo por su flexibilidad al compartir enlaces y revisiones periódicas en el flujo de trabajo. Gracias a la plataforma, se pudo mantener un desarrollo organizado, transparente y flexible en cada una de sus etapas.

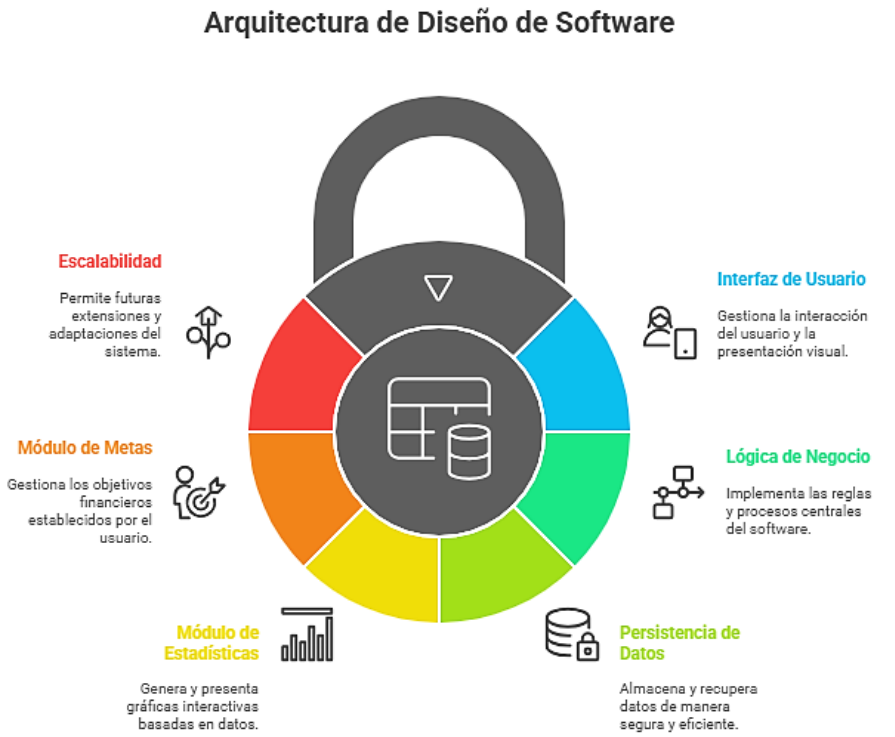
4.1.2. Descripción de la planificación del proyecto de desarrollo de software

Para la planificación del proyecto se usó un modelo incremental de desarrollo, donde con en etapas sucesivas, incorporamos poco a poco nuevas funcionalidades en cada iteración. Esto hizo más fácil el control de versiones funcionales del sistema, promoviendo la retroalimentación continua y permitiendo así realizar ajustes conforme avanzaba el proyecto. Cada “rama” fue diseñada, implementada y probada de forma separada, lo que redujo los riesgos y mejoró la rapidez al momento de trabajar en equipo.

El proyecto fue dividido en cuatro fases principales: análisis, diseño, implementación y pruebas. En la fase de análisis se definieron los requisitos funcionales y no funcionales del sistema. La fase de diseño se centró en la arquitectura de la aplicación, la estructura de datos y la interfaz de usuario. Durante la implementación se desarrollaron los módulos principales, como el registro de gastos e ingresos, cálculo de balance, visualización de gráficas y configuración de metas. Finalmente, en la fase de pruebas se garantizó las funcionalidades en varios celulares. Los hitos importantes fueron la finalización del esquema inicial con la base de datos e interfaz, la integración de funciones, y la versión final con conexiones entre pantallas.



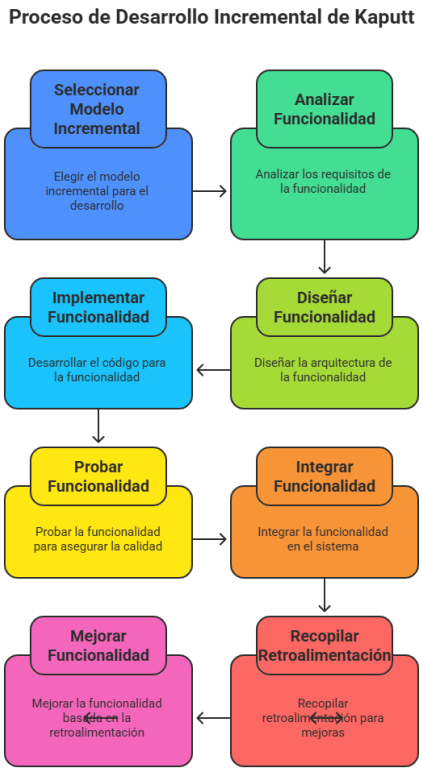
El diseño del software se basó en una arquitectura dividida en capas: interfaz de usuario, lógica de negocio y bases de datos. Esta separación permitió mantener un código organizado. Cada uno fue diseñado para cumplir una función específica; por ejemplo, el módulo de estadísticas se encarga de generar y presentar gráficas interactivas basadas en los datos del usuario, mientras que el módulo de metas gestiona los objetivos financieros establecidos por el usuario.



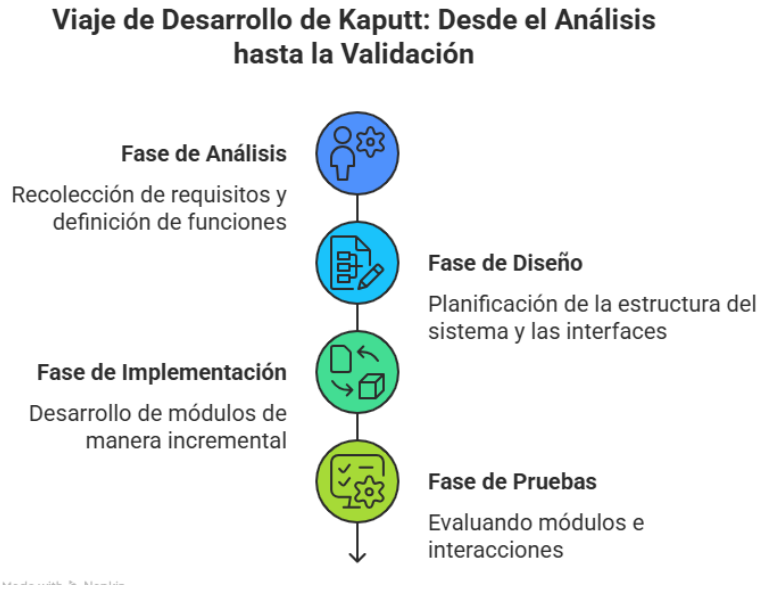
.

1. Modelo y proceso de desarrollo
   Para el desarrollo de la aplicación, se adoptó el modelo incremental de desarrollo de software, este nos ayuda a construir el sistema en ciclos sucesivos,

donde podemos agregar funcionalidades nuevas en cada iteración mientras se mantiene una versión operativa y funcional en las “Branch”. Este enfoque fue elegido por su flexibilidad, su capacidad de adaptarse a cambios durante el desarrollo, y por fomentar una mejora continua mediante la retroalimentación constante.



2. Fases de hitos del proceso



3. Estudio del dominio:

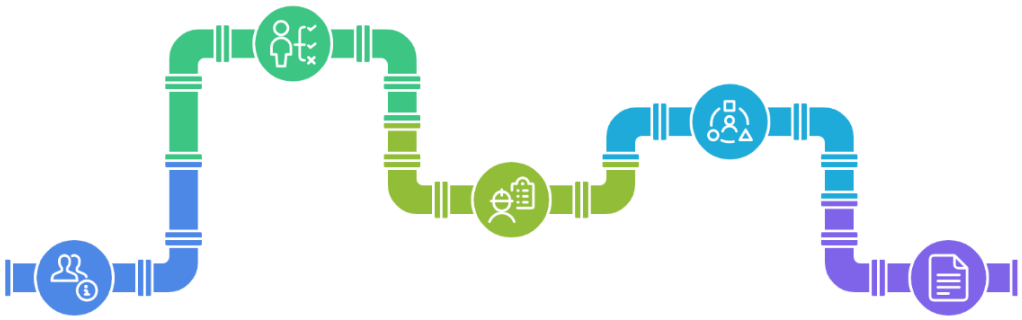
El dominio del proyecto pertenece al área de gestión financiera personal, enfocada en estudiantes universitarios que desean llevar un control detallado de sus finanzas o están aprendiendo a hacerlo. Se identificó una necesidad común: la falta de conciencia sobre

los hábitos de consumo y la dificultad para establecer metas de ahorro realistas. El estudio del dominio, a partir de las encuestas, reveló que las aplicaciones existentes suelen ser complejas o impersonales, por lo que se propuso ofrecer una alternativa más visual, simple e intuitiva, integrando funciones como retos financieros, gráficos y metas personalizadas.

4. Historias de usuario – catálogo de requisitos

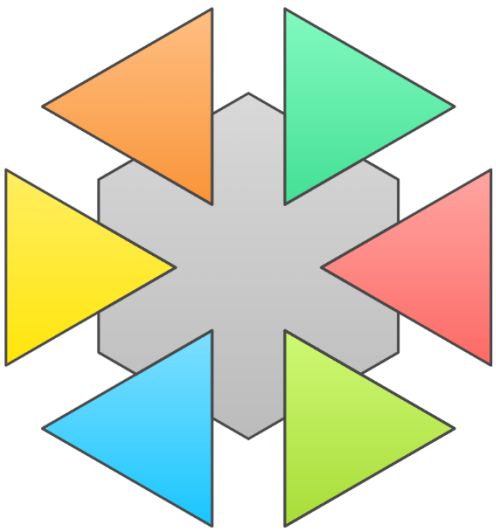
Durante el análisis de requisitos, se establecieron los requisitos funcionales, como permitir el registro de ingresos y gastos, el cálculo automático del balance, la configuración de metas de ahorro, la visualización de gráficas y la participación en retos de ahorro. Entre los requisitos no funcionales se incluyeron la usabilidad, la seguridad de los datos y el rendimiento. El modelado de requisitos permitió definir claramente los actores del sistema y los casos de uso asociados, asegurando que la aplicación cubriera las necesidades del usuario desde una perspectiva práctica y técnica.

Proceso de Análisis y Modelado de Requisitos



01	02	03	04	05
Recolección de Información e identificación	**Requisitos funcionales**	**Requisitos no funcionales**	**Modelado de requisitos**	**Documentación de Requisitos**
Recopilación de datos del dominio a través de encuestas realizadas Identificación de necesidades del usuario.	Registrar ingresos. Registrar gastos. Calcular y mostrar el balance automáticamente. Generar gráficas visuales de ingresos y egresos. Establecer metas de ahorro personalizadas. Participar en retos financieros para reducir gastos. Recibir recordatorios o alertas según comportamiento financiero.	Interfaz sencilla y amigable. Respuesta rápida del sistema. Compatibilidad con dispositivos móviles. Seguridad de los datos del usuario. Accesibilidad básica y comprensión intuitiva.	Descripciones textuales de los casos de uso. Tabla de requisitos clasificados por tipo, prioridad y descripción. Trazabilidad clara entre requisitos y funcionalidades del sistema.	Documentación de requisitos en un formato estructurado y una visión completa y estructurada que sirvió de base sólida para el diseño del sistema.

Funcionalidades de la Aplicación Kaputt



Registrar Ingresos

Permite a los usuarios ingresar sus fuentes de ingresos

Registrar Gastos

Permite a los usuarios rastrear sus gastos

Visualizar Balance

Muestra el estado financiero actual del usuario

Generar Gráficas

Crea representaciones visuales de datos financieros

Establecer Metas de Ahorro

Ayuda a los usuarios a definir y alcanzar objetivos de ahorro

Hacer Retos

Involucra a los usuarios en retos financieros interactivos

5. Modelado de casos de uso

El actor principal sería el usuario, es decir la persona que gestiona sus finanzas personales en la app.

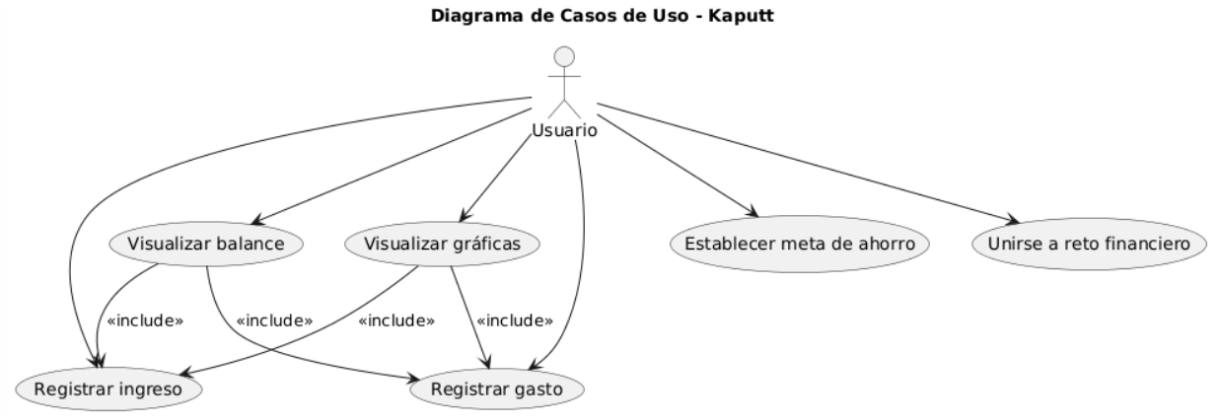
> **El usuario puede:**

- Ingresar gastos
- Ingresar ingresos
- Establecer metas de ahorro
- Hacer retos financieros
- Visualizar gráficas generadas
- Ver su balance

> **Dependencias lógicas del sistema:**

- Gastos e ingresos, permiten calcular el balance
- Gastos e ingresos, generan las gráficas
- Visualización del balance y de las gráficas son acciones que requieren previamente los registros
- Los retos y las metas son independientes, pero se relacionan con el comportamiento financiero

El propósito del diagrama es mostrar qué funciones puede ejecutar el usuario, representar gráficamente la interacción entre el usuario y el sistema y así garantizar que todas las funciones estén pensadas desde el punto de vista del usuario



6. Catálogo de requerimientos:

Requisitos funcionales:

Req	Descripción	Prioridad
RF1	El sistema debe permitir al usuario registrar ingresos con monto, fecha y categoría.	Alta
RF2	El sistema debe permitir registrar gastos con monto, fecha y categoría.	Alta
RF3	El sistema debe calcular el balance financiero total automáticamente.	Alta
RF4	El sistema debe generar gráficas de ingresos, egresos y balance por período.	Media
RF5	El sistema debe permitir al usuario visualizar su balance actual.	Alta
RF6	El sistema debe permitir establecer metas de ahorro con plazo y monto.	Media
RF7	El sistema debe permitir participar en retos financieros sugeridos.	Media

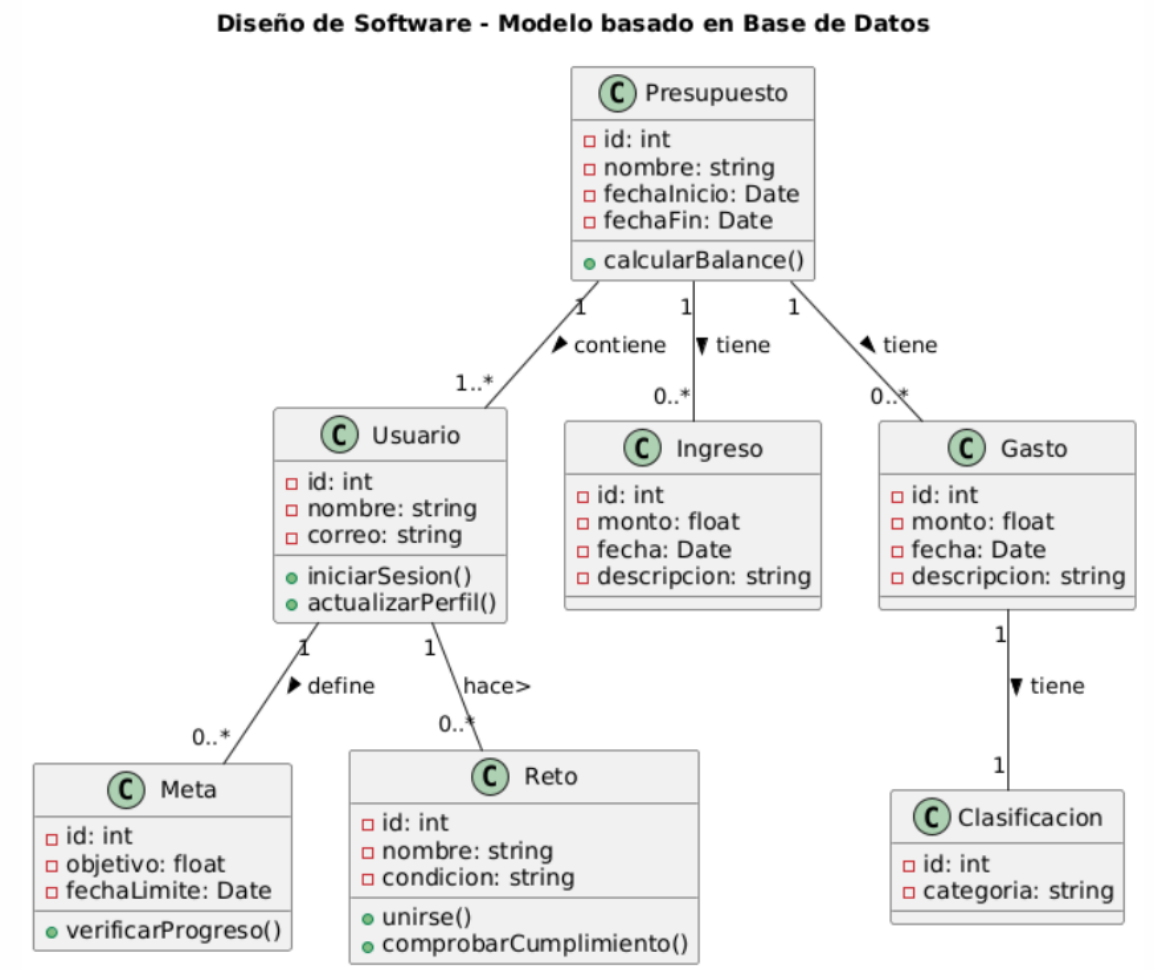
Requisitos no funcionales:

Req	Descripción	Prioridad
RNF1	La interfaz del sistema debe ser intuitiva y amigable para usuarios sin experiencia técnica.	Alta
RNF2	El tiempo de respuesta del sistema no debe superar los 2 segundos en operaciones comunes.	Alta
RNF3	El sistema debe almacenar los datos de forma segura y confidencial.	Alta
RNF4	La aplicación debe ser compatible con dispositivos móviles Android.	Alta
RNF5	El sistema debe permitir la navegación sin conexión para funciones básicas.	Baja
RNF6	El sistema debe permitir sincronización con una cuenta del usuario (en futuras versiones).	Baja

7. Diagrama de software

El diseño del software se estructuró teniendo en cuenta tres capas principales: la capa de presentación, encargada de la interfaz gráfica y la interacción directa con el usuario; la capa de lógica de negocio, donde se procesan las reglas del sistema, como el cálculo del

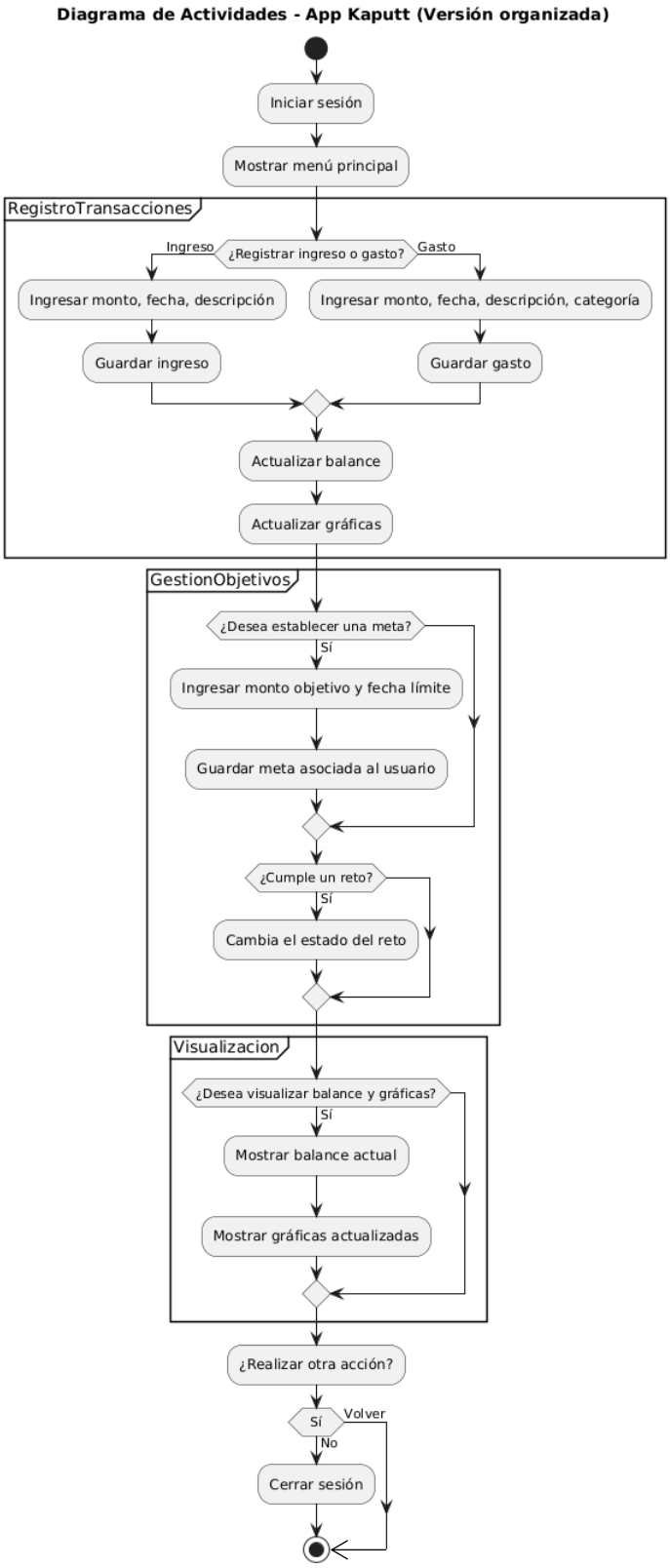
balance o la validación de los datos ingresados; y la capa de base de datos, responsable del almacenamiento de información en bases de datos. Adicionalmente, se diseñó un modelo de clases básico, en el que se definieron entidades como Usuario, presupuesto, Ingreso, Gasto, Meta, Reto y clasificación, cada una con sus atributos y métodos específicos. Las relaciones entre clases fueron definidas mediante asociaciones de composición y herencia, permitiendo representar de forma clara las dependencias lógicas entre los distintos elementos del sistema. Este modelado sirvió como puente entre los requisitos funcionales y la implementación, para que nuestro proyecto tenga coherencia y esté alineado a los requerimientos.



8. Diagrama de actividades

El diagrama de actividades de la aplicación representa la secuencia lógica de acciones que un usuario sigue para entrar y gestionar sus finanzas personales dentro del sistema. Este flujo inicia con el ingreso a la aplicación colocando su nombre y el balance actual. Posteriormente, cambiará de pantalla, para que tenga la elección de una acción principal: registrar un ingreso o gasto, establecer una meta, cumplir un reto, o visualizar su estado financiero. Si el usuario decide registrar una transacción, el sistema solicita los datos

correspondientes (monto, fecha, categoría) y luego actualiza automáticamente el balance del presupuesto. A continuación, se actualizan las gráficas financieras, que el usuario puede consultar en cualquier momento. Por otro lado, si se establece una meta o se participa en un reto, el sistema almacena dicha información y la asocia al usuario y al presupuesto activo. El flujo concluye con la posibilidad de que el usuario revise su progreso y cierre la app. Nos dará una visión clara cómo interactúan las funciones entre sí y cómo afecta las decisiones del usuario.



5. Conclusiones y trabajo futuro

5.1. Conclusiones

Este proyecto alcanzó la solución propuesta en el campo financiero. Entendemos que una herramienta amigable, accesible y adaptada podría acelerar y facilitar el ritmo de vida académico de un estudiante. Es por ello que mediante el análisis de otras aplicaciones del mercado, pudimos identificar las limitaciones comunes y diseñar nuestro proyecto que permite registrar gastos, ingresos, generar gráficas consistentes con distintas visualizaciones diarias, semanales, mensuales y anuales, ingresar metas y cumplir retos para fomentar hábitos financieros saludables. La aplicación ayuda al estudiante a crecer, ahorrar y planificar con inteligencia.

Se utilizaron tecnologías como Android Studio, para la creación de la app con interfaces, funciones y bases de datos, el lenguaje java para el desarrollo de software y SQLite para el almacenamiento de datos. Sumado a la planificación y análisis, usamos la metodología de Kanban, que ayudó a verificar y no pasar por alto errores, ayudó a corregirlos y mejorar funciones y enormemente a la división del trabajo de equipo de este proyecto. Además, incorporamos entrevistas a otros estudiantes de universidades diferentes para el levantamiento de requisitos y acercarnos más a la realidad entera que se vive día a día y así comprender más de un lenguaje financiero con el fin de crear una app que se convierte no solo en gestor económico, sino en una oportunidad que cada ahorro e ingreso registrado nos promete.

5.2. Líneas de trabajo futuro

La aplicación cumple con todos los objetivos establecidos, pero siempre hay posibles mejoras. Algunas expansiones a futuro podrían ser una sincronización de dato mejorada con ingreso de datos a la nube a través del Gmail del usuario. Además de datos en línea para que no solo acceda desde un dispositivo. Para lograr esto, se necesitaría una compatibilidad en muchas plataformas que busque una educación financiera integrada y colaboración entre usuarios. Además, sería conveniente el uso de inteligencia artificial para personalizar más retos dinámicos al usuario.

6. Bibliografía

Dybka, P. (2014, agosto 1). Barker's notation. Vertabelo Data Modeler; Vertabelo. https://vertabelo.com/blog/barkers-erd-notation/

Kanban principles and practices explained. (s/f). Kanban Software for Agile Project Management. Recuperado el 14 de julio de 2025, de https://businessmap.io/kanban-resources/getting-started/kanban-principles-practice

Martins, J. (2025, enero 19). ¿Qué es la metodología Kanban y cómo funciona? Asana. https://asana.com/es/resources/what-is-kanban

What is GitHub and why should you use it? (2023, julio 27). Coursera. https://www.coursera.org/articles/what-is-git