Construcción de aplicaciones móviles

MisFinanzas

Nicolas Rodriguez Castillo

Problema

Presentación de solución

Una aplicación móvil diseñada para ayudar a los usuarios a gestionar sus finanzas personales de manera efectiva.

MisFinanzas

Objetivos de la aplicación

- Facilitar el registro y seguimiento de ingresos y gastos.
- Proveer herramientas de análisis para una mejor toma de decisiones financieras.
- Ofrecer una interfaz amigable y accesible para todos los usuarios.

Funcionalidades clave

Registro de Transacciones:

•Ingreso manual de ingresos y gastos con categorización.

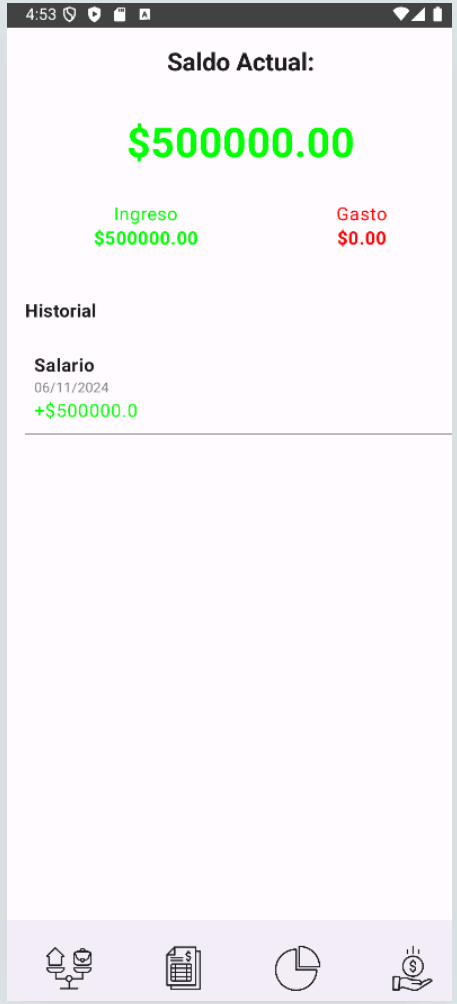
Visualización de Saldo:

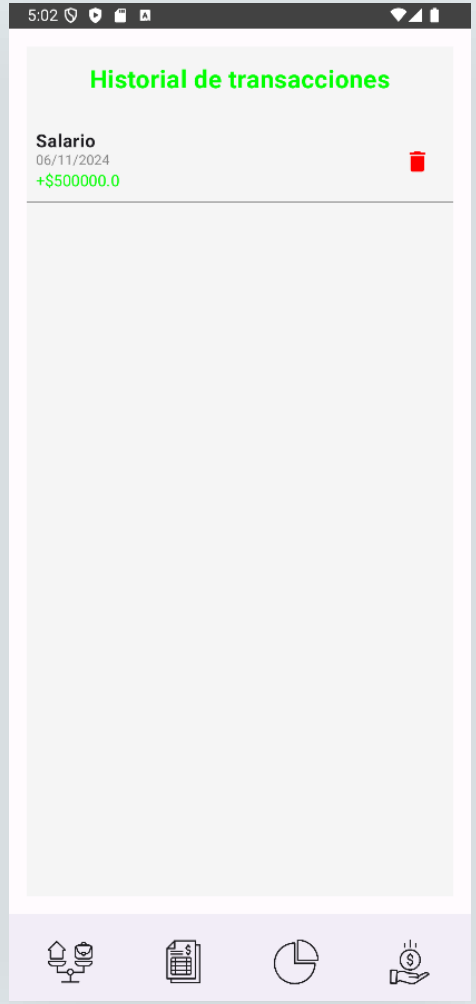
•Historial de transacciones recientes con información sobre la fecha, categoría, tipo de transacción y valor de la transaccion.

Análisis de Gastos:

•Gráficos que muestran la distribución de gastos e ingresos.

Funcionalidades MisFinanzas





Beneficios de la aplicacion

CONTROL FINANCIERO: Permite a los usuarios tener un control detallado de sus finanzas, reduciendo el riesgo de deudas innecesarias.

TOMA DE DECISIONES INFORMADA: Proporciona datos y análisis que ayudan a los usuarios a tomar decisiones financieras más acertadas.

ACCESIBILIDAD: Disponible en dispositivos móviles, permitiendo el acceso a la información financiera en cualquier momento y lugar.

Tecnologías usadas

Plataforma de Desarrollo

Android Studio:

- La aplicación está desarrollada para el sistema operativo Android, aprovechando su amplia base de usuarios y compatibilidad con dispositivos móviles.

Lenguajes y Frameworks

Kotlin:

- Lenguaje de programación principal utilizado por su concisión, seguridad y compatibilidad con Android.

Jetpack Compose:

- Framework de UI moderno para construir interfaces de usuario de manera declarativa y eficiente.

Tecnologías usadas

Arquitectura y Patrones de Diseño

MVVM (Model-View-ViewModel):

- Patrón de arquitectura que separa la lógica de la interfaz de usuario de la lógica de negocio, facilitando el mantenimiento y la escalabilidad.

Repository Pattern:

- Gestión de datos centralizada que maneja la fuente de datos, ya sea local o remota.

Persistencia de Datos

Room Database:

- Biblioteca de persistencia de datos que proporciona una capa de abstracción sobre SQLite, facilitando el almacenamiento y recuperación de datos.

Tecnologías usadas

Gestión de Navegación

Jetpack Navigation:

- Componente para gestionar la navegación entre diferentes pantallas de la aplicación de manera sencilla y eficiente.

Dependencias y Bibliotecas Adicionales

Coroutines y Flow:

- Manejo asíncrono de tareas y flujos de datos reactivos para una experiencia de usuario fluida.

Testing y Calidad de Software

JUnit:

- Herramientas para pruebas unitarias, garantizando la calidad y fiabilidad de la aplicación.

Demostración de la aplicacion