Instituto Tecnológico Superior de Turismo y Patrimonio Yavirac

INFORME DE AVANCE TÉCNICO Y FUNCIONAL

Expenses Tracker

Fase Practica

Segundo Semestre

Estudiante: Melany Rea

Carrera: Desarrollo de Software

Tutor a cargo: Ing. Diego Cando

ÍNDICE

1. Introducción

Este informe reúne todos los avances logrados hasta la fecha en el desarrollo de la aplicación móvil “Gestor de Gastos”. El objetivo principal es facilitar la administración financiera personal y grupal, permitiendo a los usuarios registrar, visualizar y compartir gastos e ingresos de forma fácil y segura.
El documento repasa los principales hitos semanales, los problemas superados y las decisiones técnicas adoptadas, además de detallar la arquitectura final de la base de datos y las principales configuraciones del proyecto.

2. Cronograma de Avances

2.1 Semana 1: Primeros pasos y prototipo

La semana comenzó con la exploración de Kotlin y Jetpack Compose, centrándome en aprender a crear interfaces modernas. Como ejercicio inicial, desarrollé una pequeña app llamada “Gestor Gastos”, donde se podían registrar gastos básicos con una simple descripción y monto.
Intenté integrar Supabase como backend utilizando la librería KTOR, logrando las primeras conexiones y guardado de datos, aunque con varias limitaciones en funcionalidad.

2.2 Semana 2: Rediseño y retos técnicos

Al revisar los requisitos reales, decidí rehacer el proyecto desde cero en Jetpack Compose. Implementé pantallas para login, registro, home, agregar gasto, historial y filtros.
En esta etapa, Supabase resultó poco práctico: las librerías no eran compatibles, la autenticación (goTrue) daba errores constantes y no se podía mantener una sesión de usuario estable. Probé varias versiones, borré y recreé el proyecto, pero no hubo avances funcionales reales.

Finalmente, opté por buscar una solución alternativa para no perder más tiempo en esta integración.

2.3 Semana 3: Migración a Firebase y consolidación

Esta semana el enfoque fue migrar todo el backend a Firebase, aprovechando Firestore y Auth.
La integración fue mucho más sencilla y estable. Configuré las dependencias necesarias y el archivo google-services.json en el proyecto.
Con esto, logré implementar el registro y autenticación de usuarios, así como el guardado y consulta de gastos por usuario.

También ajusté las reglas de seguridad de la base de datos para evitar accesos no autorizados.
Durante las pruebas, surgieron algunos errores por falta de permisos y manejo de corutinas, pero se solucionaron revisando la documentación y haciendo pruebas controladas.

2.4 Semana 4: Mejoras en interfaz y experiencia de usuario

Con la base funcional ya establecida, dediqué la semana a pulir la interfaz y mejorar la experiencia de usuario:

- Agregué una gráfica de dona con la librería MPAndroidChart para mostrar visualmente la relación entre ingresos y gastos.
- Simplifiqué el proceso de agregar ingresos/gastos permitiendo seleccionarlos desde la misma pantalla.
- Incorporé un resumen financiero en el home, mostrando ingresos, egresos y el balance actual.
- Añadí una lista de últimas transacciones, que se pueden editar de forma rápida.

- Realicé mejoras visuales en márgenes, colores y tipografías, asegurando coherencia en toda la app.
- Corregí detalles como cortes de texto y formatos de fecha, y ajusté la lógica para mostrar sólo los controles necesarios (por ejemplo, el checkbox de “gasto recurrente” solo cuando aplica).

2.5 Semana 5: Billeteras compartidas y gastos programados

Billeteras compartidas

En esta etapa incorporé la funcionalidad de crear billeteras individuales o compartidas.
El usuario puede invitar a otros por email y elegir si solo pueden consultar los movimientos (“Solo lectura”) o también agregar/modificar datos (“Lectura y edición”).
Gestioné los permisos en la base de datos usando los
campos sharedWith y accessibleTo.
La interfaz permite compartir, editar o eliminar billeteras, y muestra claramente el tipo de permiso asignado a cada usuario.
Me enfrenté a retos como la sincronización en tiempo real de los usuarios con acceso y la restricción de acciones según el permiso (por ejemplo, deshabilitar el botón de agregar gasto si el usuario solo tiene permiso de lectura).

Gastos programados

También desarrollé el módulo de gastos programados, permitiendo al usuario planificar futuros pagos asociados a una billetera.
Cada gasto programado puede tener estado pendiente, pagado o vencido, se pueden editar y marcar como pagados registrando el monto real.
Implementé un gráfico de dona para visualizar el estado de los gastos programados y filtros para alternar entre pendientes o vencidos.
Hubo dificultades para refrescar correctamente las gráficas y filtrar los estados tras cada cambio, pero se resolvieron ajustando la lógica de actualización y los listeners.

2.6 Semana 6: Transacciones, seguridad y correcciones finales

Transacciones y estructura de datos

Esta semana añadí la colección transactions para registrar cada movimiento realizado en una billetera, ya sea ingreso o egreso, con detalles de usuario, fecha y categoría.
Usé el modelo WalletWithCounts para mostrar en la interfaz cuántos gastos programados están pendientes o pagados por billetera.

Seguridad y reglas en la base

Me enfoqué en reforzar la seguridad, revisando las reglas de Firebase para que cada usuario sólo pueda acceder a lo que le corresponde.
Validé que los permisos de lectura y edición funcionen correctamente tanto en la app como en la consola de Firebase.

Mejoras y correcciones

Hice ajustes en formularios, márgenes y navegación para que todo sea más intuitivo.
Solucioné bugs que me reporté y consulté en días anteriores, como la actualización de listas tras editar/eliminar datos, el refresco de la gráfica y la navegación entre pantalla

---

3. Estructura de la base de datos

A continuación se describen las colecciones principales creadas en Firebase Firestore:

- users: Almacena los datos básicos de los usuarios registrados.
- expenses: Guarda los gastos e ingresos de cada usuario.
- wallets: Billeteras personales y compartidas; contiene información sobre permisos y usuarios con acceso.
- scheduled_expenses: Gastos programados asociados a cada billetera.
- transactions: Registro detallado de transacciones (ingresos y egresos) por billetera.
- app_settings: Configuración general de la aplicación.

4. Aspectos técnicos relevantes

- El archivo build.gradle.kts del módulo principal está configurado para incluir:
    - Soporte para Kotlin, Jetpack Compose y viewBinding.
    - Dependencias de Firebase para Auth, Firestore y Analytics, usando la BOM oficial para evitar conflictos de versiones.
    - MPAndroidChart para las gráficas.
    - Room, preparado para una posible futura persistencia local.
    - Material Components para un diseño moderno.
    - Corutinas y lifecycle para operaciones asíncronas y sincronización en tiempo real.
    - Compatibilidad con minSdk 24 y targetSdk 34, asegurando funcionamiento en la mayoría de dispositivos recientes.

Fragmento de dependencias principales:

Kotlin

```
implementation(platform("com.google.firebase:firebase-bom:32.7.2"))

implementation("com.google.firebase:firebase-analytics")

implementation("com.google.firebase:firebase-auth")

implementation("com.google.firebase:firebase-firestore")

implementation("androidx.compose.material3:material3")

implementation("com.github.PhilJay:MPAndroidChart:v3.1.0")

implementation("androidx.room:room-runtime:2.6.1")

kapt("androidx.room:room-compiler:2.6.1")

implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.8.0")

implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.8.0")
```

5. Estado actual

La aplicación permite el registro y autenticación de usuarios, registro y edición de gastos e ingresos, visualización de balances y breakdowns por categoría, creación y gestión de billeteras compartidas con diferentes tipos de permisos, manejo de gastos programados y registro detallado de transacciones, todo con una interfaz clara y segura.

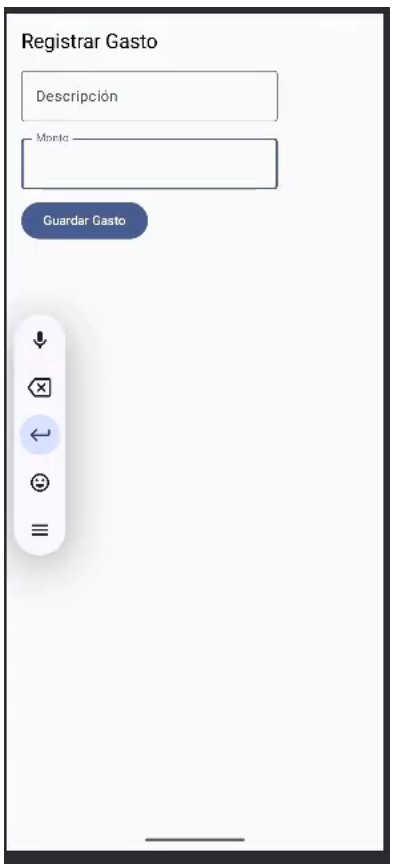
Siguientes pasos:
Como mejoras futuras se plantea:

- Optimizar aún más la sincronización de datos y la experiencia visual.
- Añadir notificaciones para gastos programados y cambios en billeteras compartidas.
- Implementar una base de datos local (Room) para funcionamiento offline.
- Realizar pruebas con usuarios y seguir refinando la experiencia según sus sugerencias.

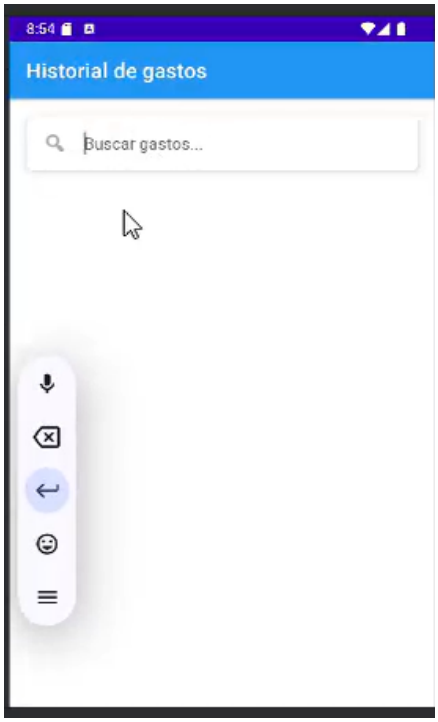
6. Anexos

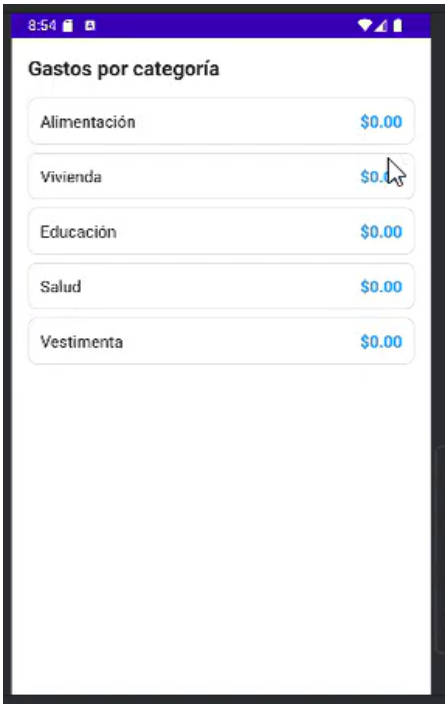
- **Capturas de pantalla de la app desde los inicios hasta la actualidad**

  App de prueba

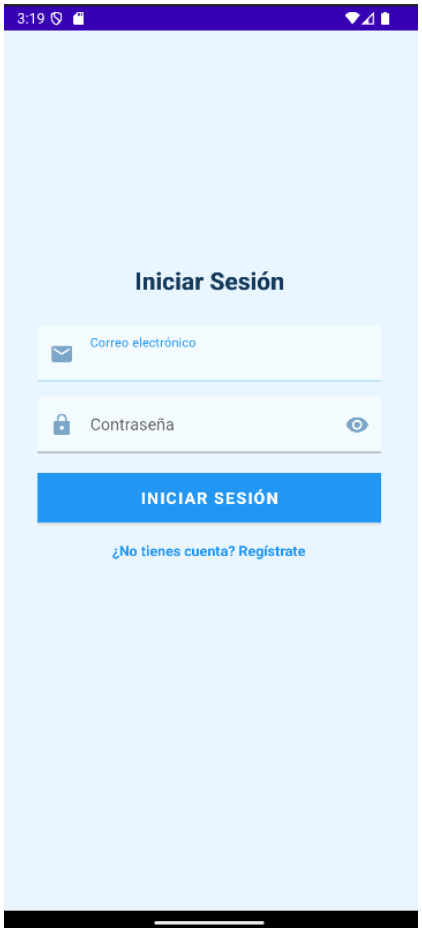


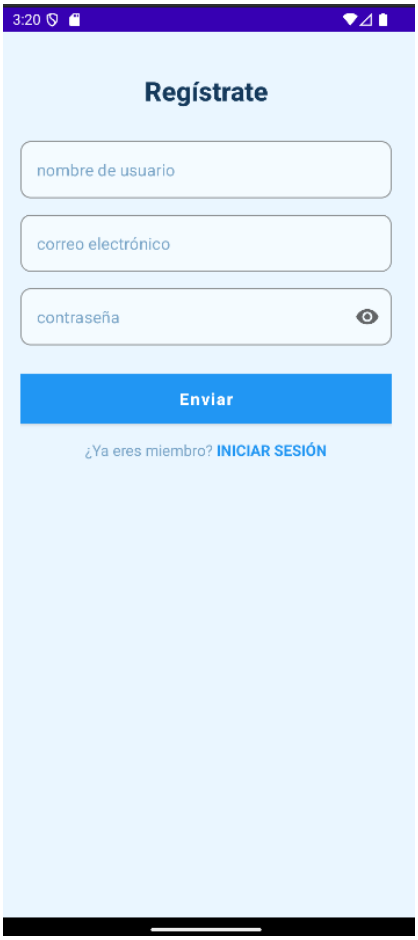
Inicios de la Applicacion antes



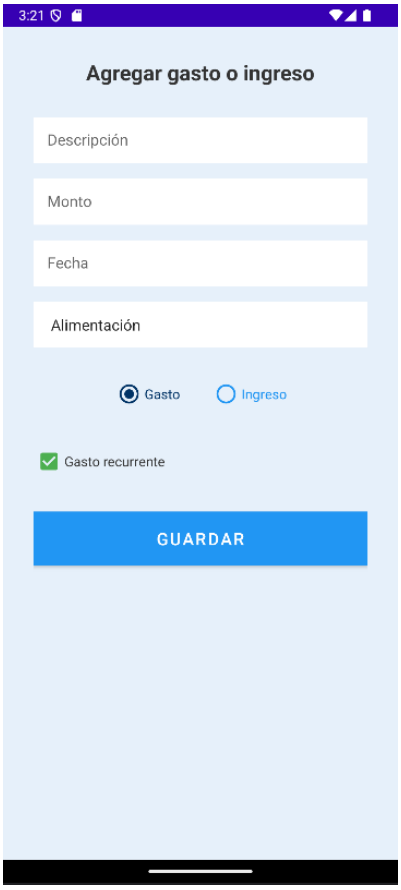


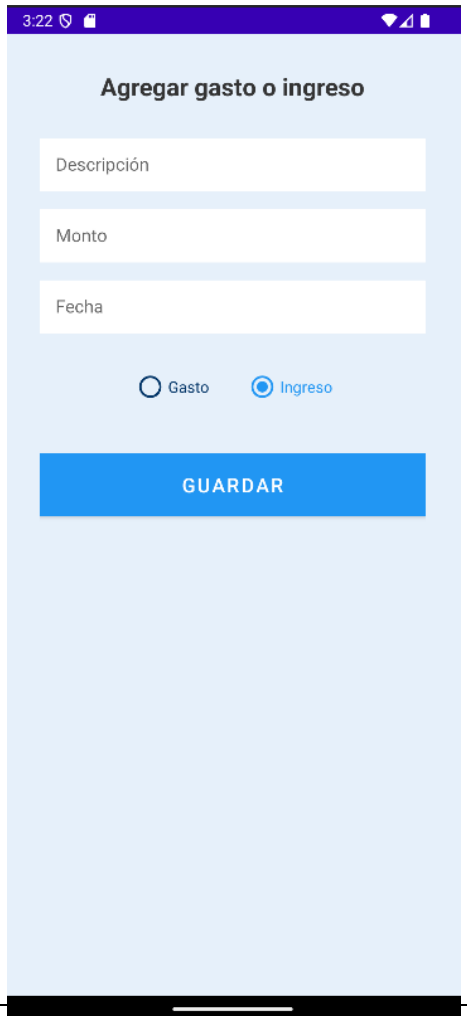
Después / Aplicación actual



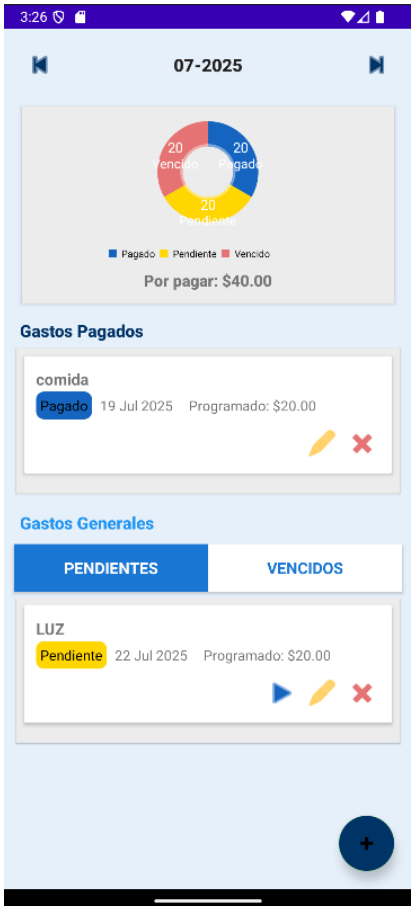



3:18
Mi billetera personal
50.0
Ingresos
87.0
Gastos
Ingresos
Gastos
DATOS FINANCIEROS
Ingresos: $50.00 USD
Gastos: $87.00 USD
Balance: $-37.00 USD
AGREGAR NUEVO GASTO/INGRESO
BILLETERAS
Últimos gastos
luz eelectrica
Vivienda
$15.00
19/07/2025

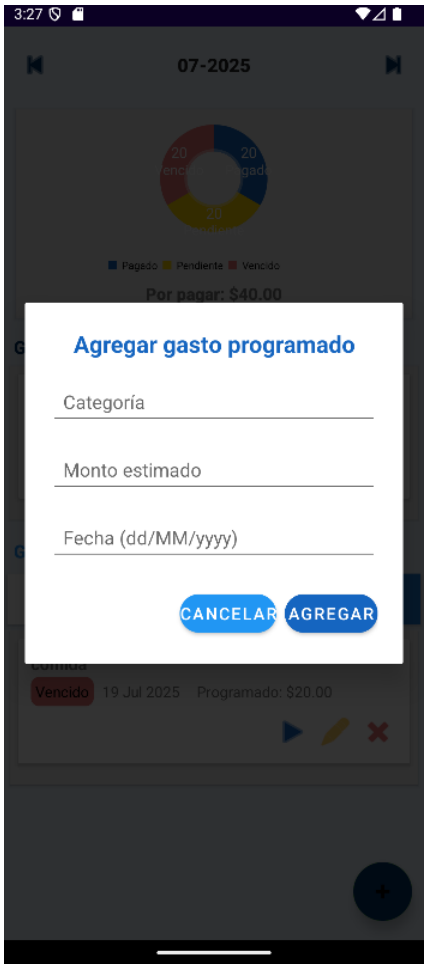

3:21
Agregar gasto o ingreso
Descripción
Monto
Fecha
Alimentación
Gasto
Ingreso
Gasto recurrente
GUARDAR


3:22
Agregar gasto o ingreso
Descripción
Monto
Fecha
Gasto
Ingreso
GUARDAR


3:24
Mis Billeteras
GASTOS CASA
Pendientes por pagar: 2
Pagados: 1
COMPARTIR
comida
Pendientes por pagar: 2
Pagados: 0
COMPARTIR
gastos
Pendientes por pagar: 0
Pagados: 1
COMPARTIR

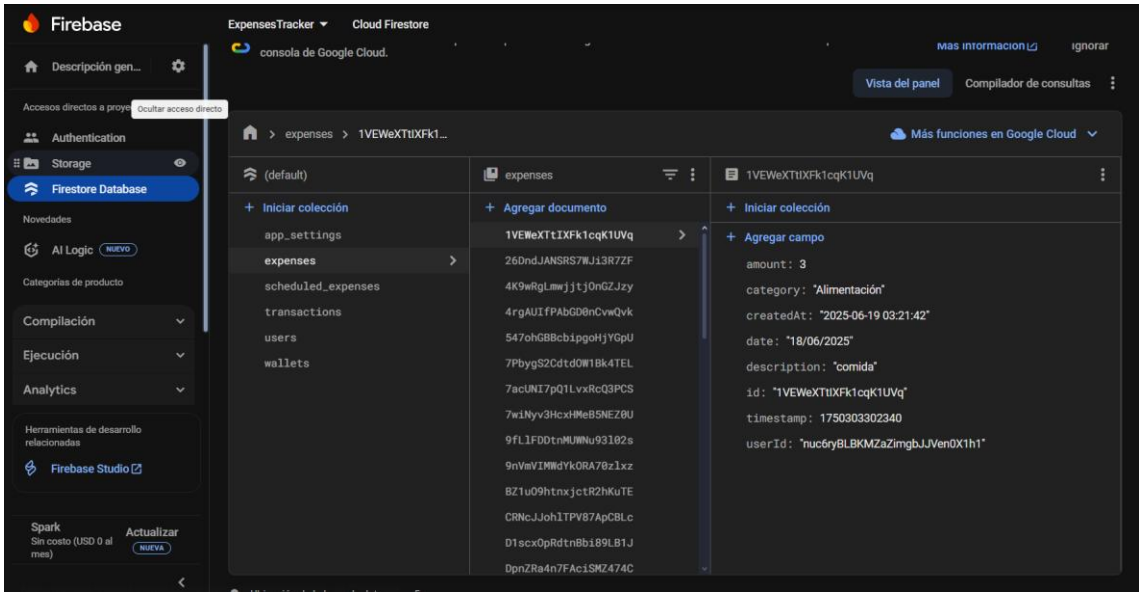

3:26
07-2025
20
20
20
Pagado
Pendiente
Vencido
Por pagar: $40.00
Gastos Pagados
comida
Pagado
19 Jul 2025
Programado: $20.00
Gastos Generales
PENDIENTES
VENCIDOS
LUZ
Pendiente
22 Jul 2025
Programado: $20.00

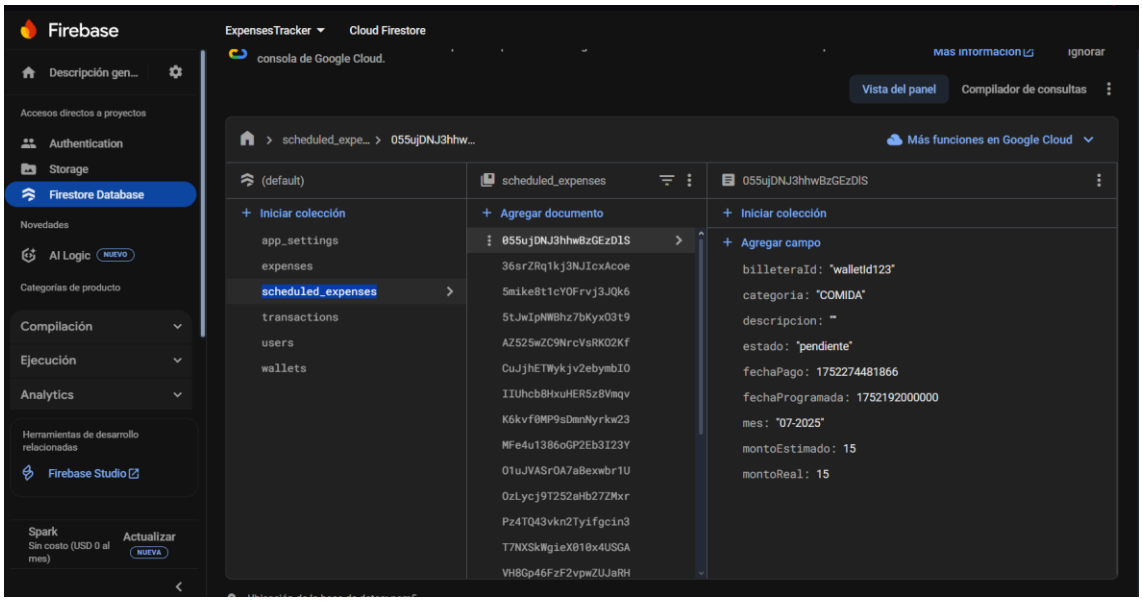

3:26
07-2025
20
20
20
Pagado
Pendiente
Vencido
Por pagar: $40.00
Gastos Pagados
comida
Pagado
19 Jul 2025
Programado: $20.00
Gastos Generales
PENDIENTES
VENCIDOS
comida
Vencido
19 Jul 2025
Programado: $20.00

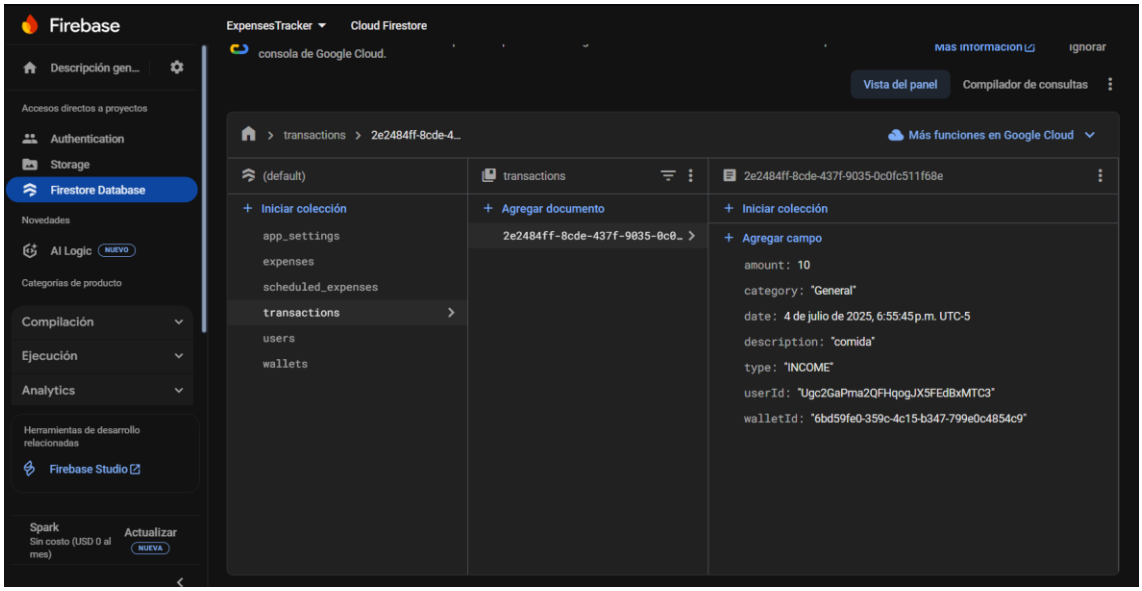

3:27
07-2025
Pagado
Pendiente
Vencido
Por pagar: $40.00
Agregar gasto programado
Categoría
Monto estimado
Fecha (dd/MM/yyyy)
CANCELAR
AGREGAR
Vencido
19 Jul 2025
Programado: $20.00

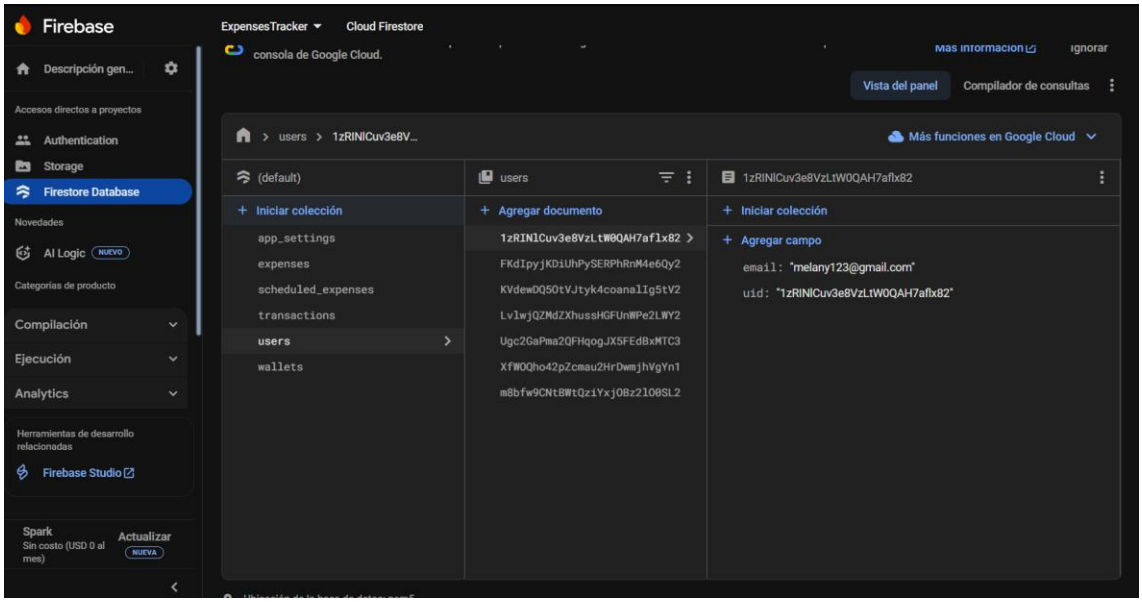

3:27
09-2025
Por pagar: $0.00
Gastos Pagados
Gastos Generales
PENDIENTES
VENCIDOS
comida
Vencido
19 Jul 2025
Programado: $20.00

Firebase

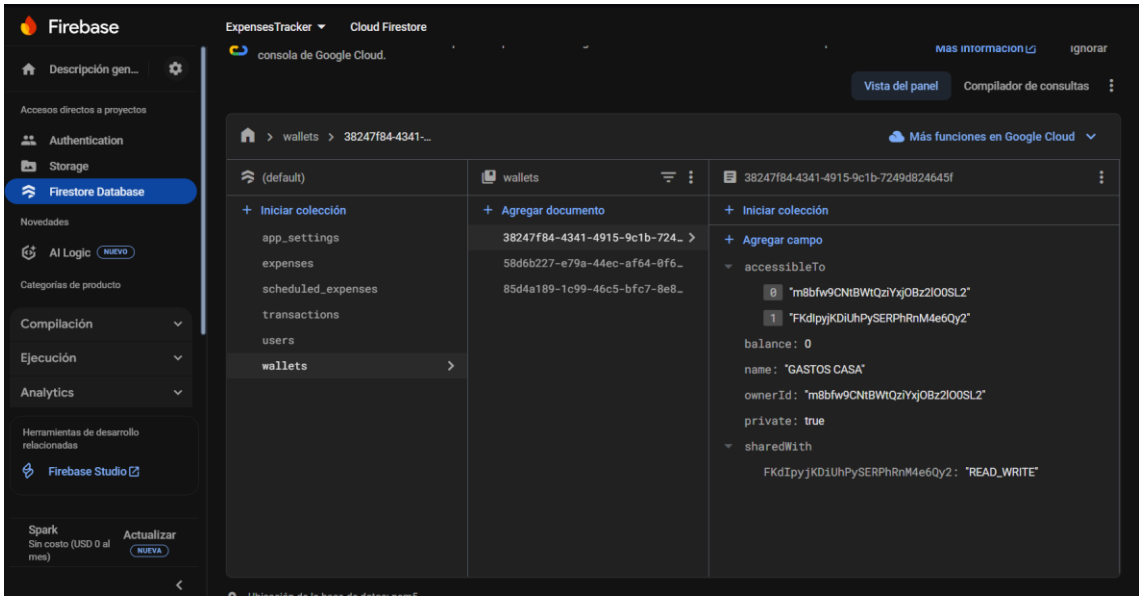

Firebase
ExpensesTracker
Cloud Firestore
consola de Google Cloud.
Descripción gen...
Accesos directos a proyectos
Ocultar acceso directo
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Herramientas de desarrollo relacionadas
Firebase Studio
Spark
Sin costo (USD 0 al mes)
Actualizar
NUEVA
Más información
Ignorar
Vista del panel
Compilador de consultas
expenses > 1VEWeXTtIXFk1...
Más funciones en Google Cloud
(default)
expenses
1VEWeXTtIXFk1cqK1UVq
Iniciar colección
Agregar documento
Agregar campo
app_settings
expenses
scheduled_expenses
transactions
users
wallets
1VEWeXTtIXFk1cqK1UVq
26DndJANSRS7WJi3R7ZF
4K9wRgLmwjjtjOnGZJzy
4rgAUIfPAbGD0nCvwQvk
547ohGBBcbipgoHjYGpU
7PbygS2CdtdOW1Bk4TEL
7acUNI7pQ1LvxRcQ3PCS
7wiNyv3HcxHMeB5NEZ0U
9fL1FDDtnMUWNu93l02s
9nVmVIMWdYkORA70zlxz
BZ1uO9htnxjctR2hKuTE
CRNcJJohlTPV87ApCBLc
D1scxOpRdtn8bi89LB1J
DpnZRa4n7FAciSMZ474C
amount: 3
category: "Alimentación"
createdAt: "2025-06-19 03:21:42"
date: "18/06/2025"
description: "comida"
id: "1VEWeXTtIXFk1cqK1UVq"
timestamp: 1750303302340
userId: "nuc6ryBLBKMZaZimgbJJVen0X1h1"

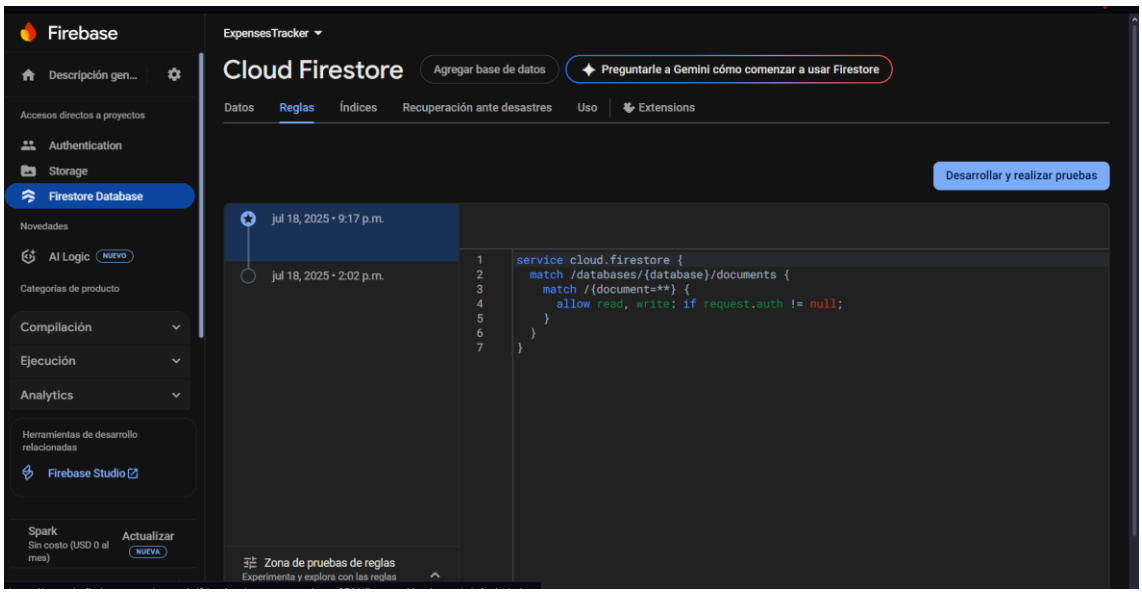

Firebase
ExpensesTracker
Cloud Firestore
consola de Google Cloud.
Descripción gen...
Accesos directos a proyectos
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Herramientas de desarrollo relacionadas
Firebase Studio
Spark
Sin costo (USD 0 al mes)
Actualizar
NUEVA
Más información
Ignorar
Vista del panel
Compilador de consultas
scheduled_expe... > 055ujDNJ3hhw...
Más funciones en Google Cloud
(default)
scheduled_expenses
055ujDNJ3hhwBzGEzDlS
Iniciar colección
Agregar documento
Agregar campo
app_settings
expenses
scheduled_expenses
transactions
users
wallets
055ujDNJ3hhwBzGEzDlS
36srZRq1kj3NJIcxAcoe
5mike8t1cYOFrvj3JQk6
5tJwIpNWBhz7bKyxO3t9
AZ525wZC9NrcVsRKO2Kf
CuJjhETWykjv2ebymbIO
IIUhcb8HxuHER5z8Vmqv
K6kvf0MP9sDmnNyrkw23
MFe4u1386oGP2Eb3I23Y
O1uJVASrOA7aBexwbr1U
OzLycj9T252aHb27ZMxr
Pz4TQ43vkn2Tyifgcin3
T7NXSkWgieX010x4USGA
VH8Gp46FzF2vpwZUJaRH
billeteraId: "walletId123"
categoria: "COMIDA"
descripcion: ""
estado: "pendiente"
fechaPago: 1752274481866
fechaProgramada: 1752192000000
mes: "07-2025"
montoEstimado: 15
montoReal: 15


Firebase
ExpensesTracker
Cloud Firestore
consola de Google Cloud.
Descripción gen...
Accesos directos a proyectos
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Herramientas de desarrollo relacionadas
Firebase Studio
Spark
Sin costo (USD 0 al mes)
Actualizar
NUEVA
Vista del panel
Compilador de consultas
transactions > 2e2484ff-8cde-4...
Más funciones en Google Cloud
(default)
transactions
2e2484ff-8cde-437f-9035-0c0fc511f68e
Iniciar colección
Agregar documento
Agregar campo
app_settings
expenses
scheduled_expenses
transactions
users
wallets
2e2484ff-8cde-437f-9035-0c0...
amount: 10
category: "General"
date: 4 de julio de 2025, 6:55:45 p.m. UTC-5
description: "comida"
type: "INCOME"
userId: "Ugc2GaPma2QFHqogJX5FEdBxMTC3"
walletId: "6bd59fe0-359c-4c15-b347-799e0c4854c9"


Firebase
ExpensesTracker
Cloud Firestore
consola de Google Cloud.
Descripción gen...
Accesos directos a proyectos
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Herramientas de desarrollo relacionadas
Firebase Studio
Spark
Sin costo (USD 0 al mes)
Actualizar
NUEVA
Vista del panel
Compilador de consultas
users > 1zRINlCuv3e8V...
Más funciones en Google Cloud
(default)
users
1zRINlCuv3e8VzLtW0QAH7aflx82
Iniciar colección
Agregar documento
Agregar campo
app_settings
expenses
scheduled_expenses
transactions
users
wallets
1zRINlCuv3e8VzLtW0QAH7aflx82
FKdIpyjKDiUhPySERPhRnM4e6Qy2
KVdewDQ5OtVJtyk4coanaIlg5tV2
LvlwjQZMdZXhussHGFUnWPe2LWY2
Ugc2GaPma2QFHqogJX5FEdBxMTC3
XfWOQho42pZcmau2HrDwmjhVgYn1
m8bfw9CNtBWtQziYxjOBz2lO0SL2
email: "melany123@gmail.com"
uid: "1zRINlCuv3e8VzLtW0QAH7aflx82"


Firebase
ExpensesTracker
Descripción gen...
Accesos directos a proyectos
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Authentication
Usuarios
Método de acceso
Plantillas
Uso
Configuración
Extensions
Las siguientes funciones de autenticación dejarán de estar disponibles cuando se cierre Firebase Dynamic Links el 25 de agosto de 2025: la autenticación a través de vínculos de correo electrónico en apps para dispositivos móviles, así como la compatibilidad con OAuth de Cordova para apps web.
Buscar por dirección de correo electrónico, número de teléfono o UID de usuario
Agregar usuario
Identificador
Proveedores
Fecha de creación
Fecha de acceso
UID del usuario
erickperalta@gmail.com
16 jul 2025
18 jul 2025
FKdIpyjKDiUhPySERPhRnM4e...
vanessarea@gmail.com
16 jul 2025
18 jul 2025
KVdewDQ5OtVJtyk4coanallg5...
melanyrea@gmail.com
16 jul 2025
20 jul 2025
m8bfw9CNtBWtQziYxjOBz2lO...


Firebase
ExpensesTracker
Cloud Firestore
consola de Google Cloud.
Más información
Ignorar
Vista del panel
Compilador de consultas
Descripción gen...
Accesos directos a proyectos
Authentication
Storage
Firestore Database
Novedades
AI Logic
NUEVO
Categorías de producto
Compilación
Ejecución
Analytics
Herramientas de desarrollo relacionadas
Firebase Studio
Spark
Sin costo (USD 0 al mes)
Actualizar
NUEVA
wallets
38247f84-4341-...
Más funciones en Google Cloud
(default)
Iniciar colección
app_settings
expenses
scheduled_expenses
transactions
users
wallets
Agregar documento
38247f84-4341-4915-9c1b-724...
58d6b227-e79a-44ec-af64-0f6...
85d4a189-1c99-46c5-bfc7-8e8...
38247f84-4341-4915-9c1b-7249d824645f
Iniciar colección
Agregar campo
accessibleTo
0 "m8bfw9CNtBWtQziYxjOBz2lO0SL2"
1 "FKdIpyjKDiUhPySERPhRnM4e6Qy2"
balance: 0
name: "GASTOS CASA"
ownerId: "m8bfw9CNtBWtQziYxjOBz2lO0SL2"
private: true
sharedWith
FKdIpyjKDiUhPySERPhRnM4e6Qy2: "READ_WRITE"


Firebase
ExpensesTracker
Cloud Firestore
Agregar base de datos
Preguntarle a Gemini cómo comenzar a usar Firestore
Datos
Reglas
Índices
Recuperación ante desastres
Uso
Extensions
Desarrollar y realizar pruebas
jul 18, 2025 • 9:17 p.m.
jul 18, 2025 • 2:02 p.m.
service cloud.firestore {
match /databases/{database}/documents {
match /{document=**} {
allow read, write: if request.auth != null;
}
}
}
Zona de pruebas de reglas
Experimenta y explora con las reglas