

Spend Wisely:

Una app para controlar
tu dinero



Alumno: Jacobo Felipe Ruiz Nina

Ciclo: Desarrollo de Aplicaciones Multiplataforma

Curso 2021-2022

Centro: IES La Vereda

Tutora: Carmen Martínez

Indice

1) Introducción	
1.1) Módulos implicados	3
1.2) Breve descripción del proyecto	4
2) Diseño	
2.1) Diagrama Entidad-Relación (ER)	5
2.2) Flujo de la aplicación (Mockups)	6
3) Recursos utilizados	
3.1) Herramientas Hardware	7
3.2) Herramientas Software	7
3.3) Complementos utilizados	8
4) Plan de trabajo	
4.1) Diagrama de Gant	10
4.2) Objetivos	11
5) Manual de usuario	
5.1) Pantalla Gastos	11
5.2) Pantalla Cuentas	16
5.3) Pantalla Resumen	18
6) Enlaces consultados	19
7) Mi repositorio GitHub	19

1) Introducción

1.1) Módulos implicados

Dentro de los módulos implicados en el proyecto podemos encontrar:

Sistemas informáticos

- El proyecto corre bajo Android, un sistema operativo libre.

Bases de datos

- Se ha creado un diagrama ER en el que se muestran las tablas utilizadas en la Base de datos del proyecto.
- Uso de DML y DDL para la gestión de las tablas.

Programación

- Se han creado distintos objetos a lo largo del desarrollo de la aplicación.
- Uso de estructuras de control: if, for, when (switch), etc.
- Se ha optimizado el código lo mejor posible para su mejor lectura y comprensión.

Lenguaje de marcas

- Se ha utilizado ficheros XML para la definición de las pantallas de la aplicación.

Entornos de desarrollo

- Las distintas versiones de la aplicación se subirán a GitHub.

Acceso a datos

- La aplicación conecta con una base de datos local que se crea al instalar la aplicación en el dispositivo Android.
- La aplicación utiliza una herramienta de Google para la gestión de Bases de Datos llamada Room, la cual funciona como un ORM.

Desarrollo de Interfaces

- La aplicación ha sido creada con Android Studio, IDE que utiliza XML's para la creación de elementos visuales de la aplicación.
- Se han utilizado distintos componentes en la aplicación: botones, botones flotantes, listas personalizables (recycler view), etc.

Programación multimedia y dispositivos móviles

- Desarrollo de la aplicación para dispositivos móviles.
- Toda la interfaz gráfica desarrollada con Android Studio.
- Persistencia de datos.
- Acceso a la cámara del dispositivo y poder tomar fotos con ella.

Programación de servicios y procesos

- Se utilizan llamadas asíncronas para realizar cambios en la Base de datos.

Sistemas de gestión empresarial

- Se implementará una ERP para gestionar el CRM y las ventas de la aplicación. En esta ocasión se utilizará la ERP de Odoo.

1.2) Breve descripción del proyecto

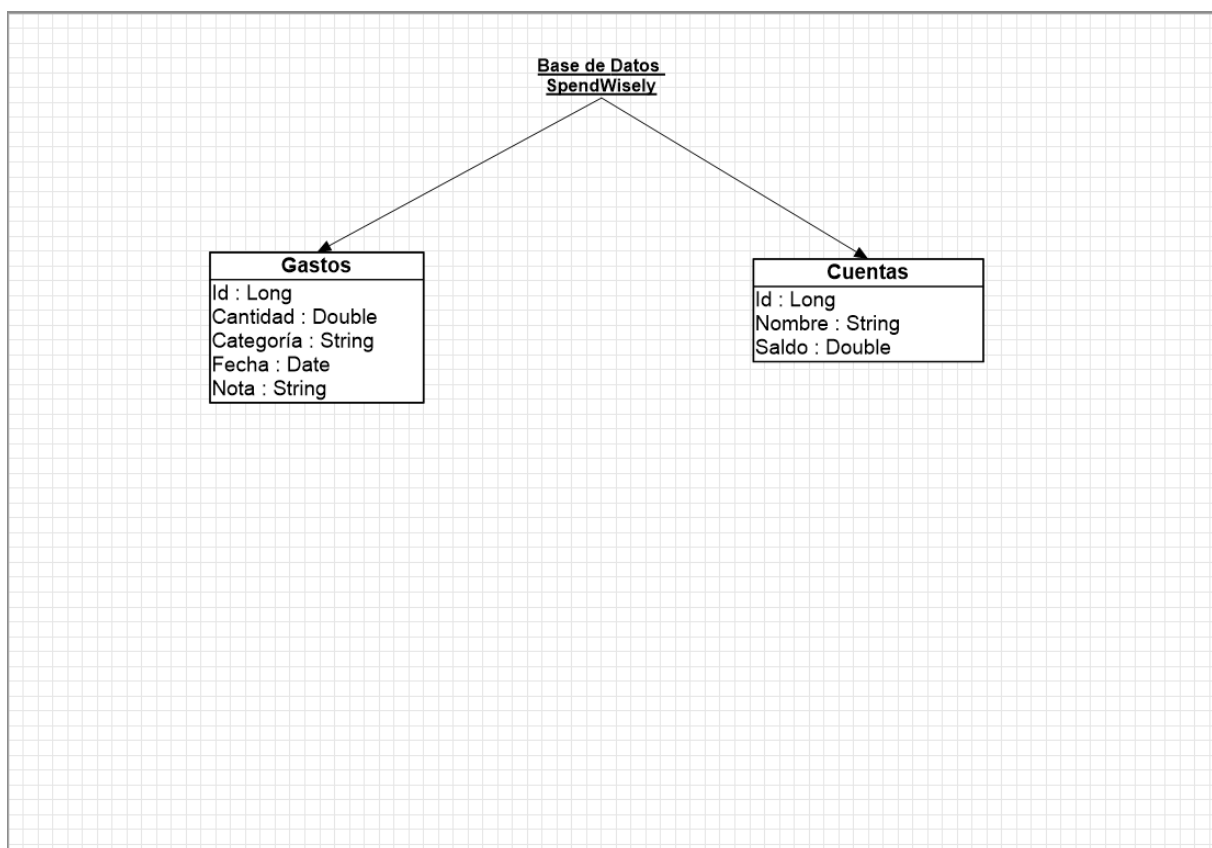
Spend Wisely es una aplicación desarrollada con el propósito de que el usuario sea capaz de controlar los gastos que hace a diario, ya que como bien dice el nombre de la aplicación, la meta es que el usuario gaste inteligentemente y así no se quede sin dinero.

2) Diseño

2.1) Diagrama Entidad-Relación (ER)

Tras un estudio previo, se determinó que con 2 tablas era suficiente para el correcto funcionamiento de la aplicación.

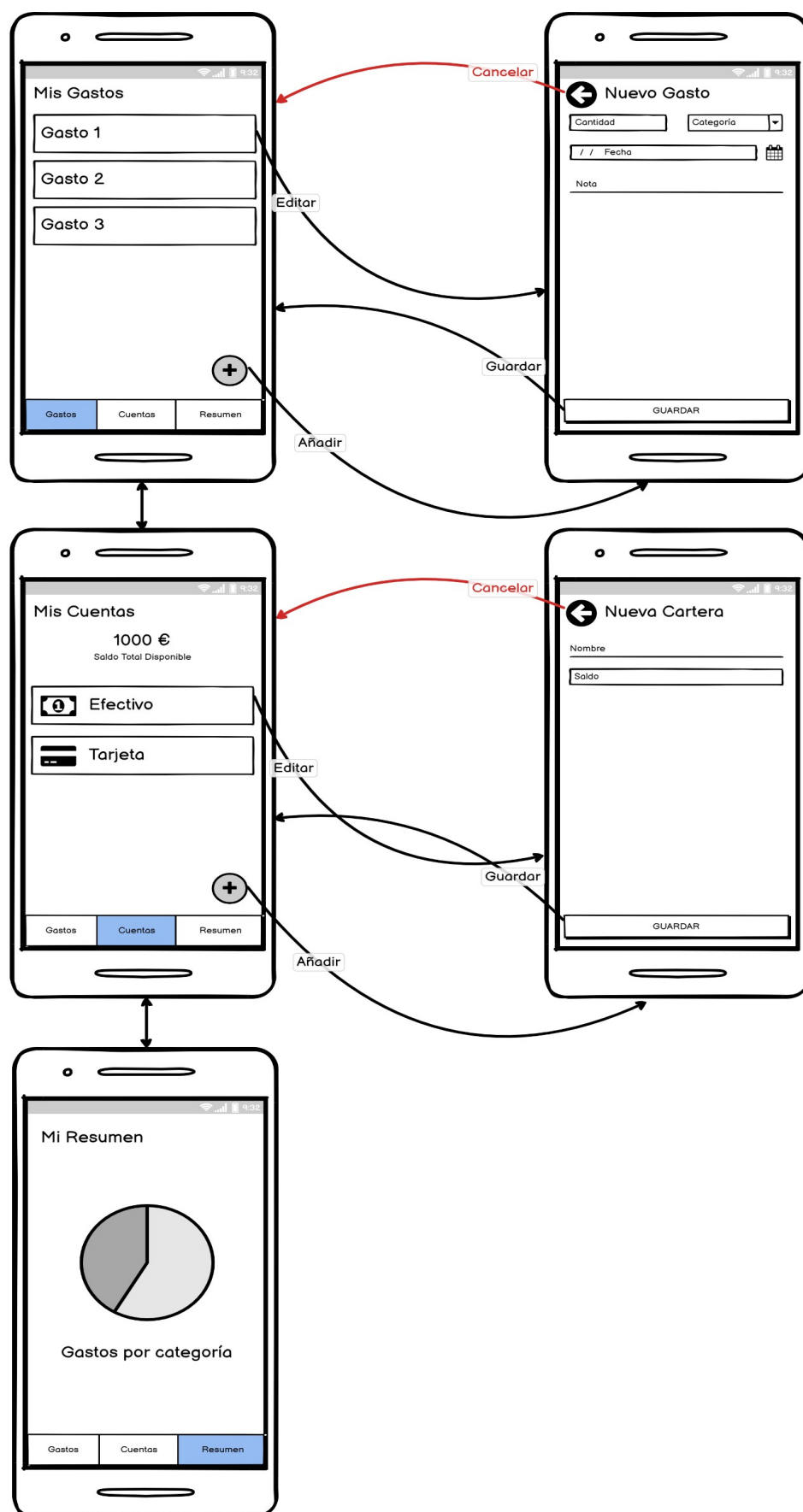
Una tabla de gastos en la cual se guardarán los gastos que vaya introduciendo el usuario y una tabla de cuentas donde se guardará el dinero que tenga el usuario disponible.



2.2) Flujo de la aplicación (Mockups)

El usuario lo primero que podrá ver será la pantalla de sus gastos y podrá navegar a través de todas las pantallas mediante un menú que tendrá en la parte inferior de la aplicación.

Pantalla inicial



3) Recursos utilizados

3.1) Herramientas Hardware

- **Un ordenador que sea capaz de poder ejecutar el Android Studio y una máquina virtual con Android a la vez.**

A continuación dejo los requisitos mínimos que pide Android Studio para un ordenador con Windows:

Windows

- Microsoft Windows 7/8/10 (32-64 bits).
- 8 GB de RAM recomendados (2 GB mínimo).
- 4 GB de espacio en disco duro (2 GB mínimo).
- Pantalla con resolución mínima de 1280×800 píxeles.
- Procesador Intel.
- Java 8.

Imagen 1

3.2) Herramientas Software

- **Android Studio:** En mi caso he utilizado la última versión de Android Studio disponible actualmente, la cual es: Android Studio Chipmunk 2021.2.1.

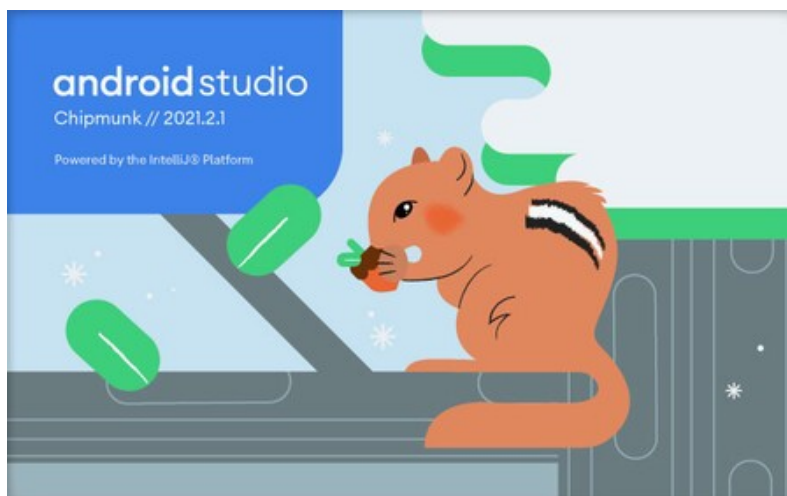


Imagen 2

- **Una máquina virtual con Android instalado:** Utilizando Android Studio como IDE para el desarrollo de la aplicación trae incorporado un gestor de máquinas virtuales con Android. En caso de no utilizar Android Studio, se tendrá que utilizar una máquina virtual con Android por separado.

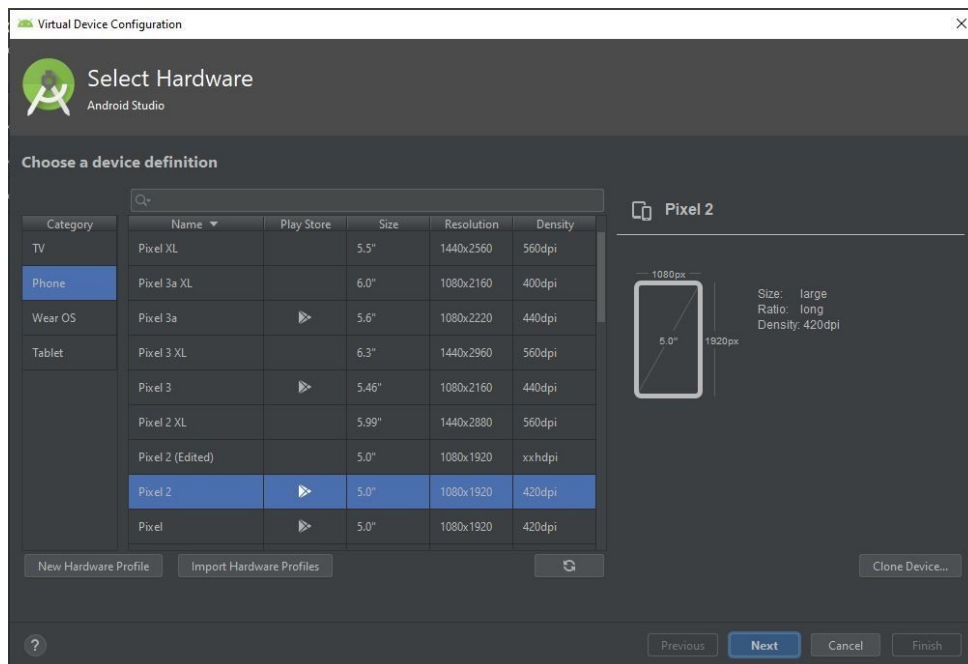


Imagen 3

3.3) Complementos utilizados

- **Room:** Dependencia creada por Google y la que recomienda utilizar para la persistencia de datos dentro de una aplicación Android. Utiliza debajo de toda la funcionalidad SQLite.

Para poder utilizar esta dependencia dentro de nuestro proyecto debemos de importarla en el build.gradle de nuestra app:

```
def roomVersion = "2.4.2"
dependencies {
    implementation("androidx.room:room-ktx:$roomVersion")
    kapt("androidx.room:room-compiler:$roomVersion")
}
```


- **Coroutines:** Dependencia que nos permitirá realizar co-rutinas dentro de la aplicación para realizar el CRUD de la Base de datos sin bloquear la interfaz del usuario.

Para poder utilizar esta dependencia dentro de nuestro proyecto debemos de importarla en el build.gradle de nuestra app:

```
def coroutines = "1.6.0"
dependencies {
    api "org.jetbrains.kotlinx:kotlinx-coroutines-core:$coroutines"
    api "org.jetbrains.kotlinx:kotlinx-coroutines-android:$coroutines"
}
```

- **LifeCycleVersion:** Dependencia que nos permitirá mostrar las modificaciones a la Base de datos de forma inmediata en la interfaz del usuario.

Para poder utilizar esta dependencia dentro de nuestro proyecto debemos de importarla en el build.gradle de nuestra app:

```
def lifecycleVersion = "2.4.1"
dependencies {
    implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:$lifecycleVersion"
    implementation "androidx.lifecycle:lifecycle-livedata-ktx:$lifecycleVersion"
    implementation "androidx.lifecycle:lifecycle-common-java8:$lifecycleVersion"
}
```

- **MPAndroidChart:** Dependencia que nos ayudará a poder introducir gráficos dentro de nuestra aplicación.



Imagen 4

Para poder utilizar esta dependencia dentro de nuestro proyecto debemos de importarla en el build.gradle de nuestra app:

```
dependencies {
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'
}
```

4) Plan de trabajo

4.1) Diagrama de Gant

El desarrollo del proyecto se ha dividido en 5 etapas:

- Diseño de la base de datos
- Diseño de la aplicación (Mockups)
- Desarrollo de la aplicación
- Testing de la aplicación
- Elaboración de la memoria del proyecto
- Últimos cambios del proyecto

[illegible][illegible]

Imagen 5

4.2) Objetivos

- Que el usuario pueda controlar su dinero de una forma correcta.
- Una aplicación que consiga almacenar registros del usuario (gastos).
- Una aplicación intuitiva y que el usuario sea capaz de usar con facilidad.
- Una aplicación agradable a la vista.
- Una aplicación que no requiera de muchos recursos para su funcionamiento.
- Que el usuario pueda ver de una forma gráfica los gastos que realiza.

5) Manual de usuario

5.1) Pantalla Gastos

Al abrir la aplicación lo primero que veremos será la pantalla de gastos:

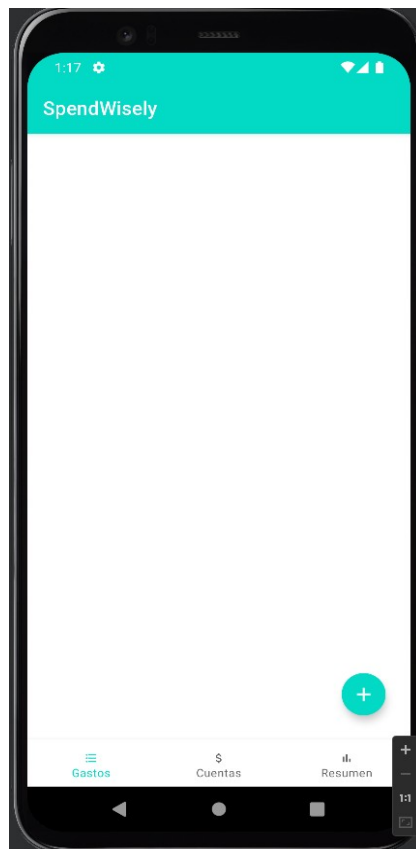


Imagen 6

En la cual podremos ver todos los gastos que vayamos introduciendo.

Como podemos ver en la imagen, en la parte inferior tenemos un menú desde el cual podremos navegar entre las tres pantallas disponibles, las cuales iremos viendo en este manual.

Crear nuevo gasto

Desde la pantalla de Gastos tendremos que dar un tap en el botón flotante con un + de icono. Acto seguido nos llevará a esta pantalla:



Imagen 7

Desde aquí podremos crear un nuevo gasto, para ello llenaremos los campos y posteriormente presionaremos en el botón GUARDAR. Si introducimos bien los datos nos saldría un mensaje diciéndonos que se ha creado el gasto satisfactoriamente, caso contrario tendremos que modificar los datos.

El resultado correcto sería el siguiente:



Imagen 8

Editar gasto

Para ello, desde la pantalla de Gastos tendremos que dar un tap en el gasto que queramos modificar. Se abrirá una pantalla para editar el gasto:



Imagen 9

Modificamos los datos y presionamos en guardar. Si hemos llenado bien los campos, nos aparecerá un mensaje diciendo que se han aplicado los cambios y veremos nuestro cambios reflejados:

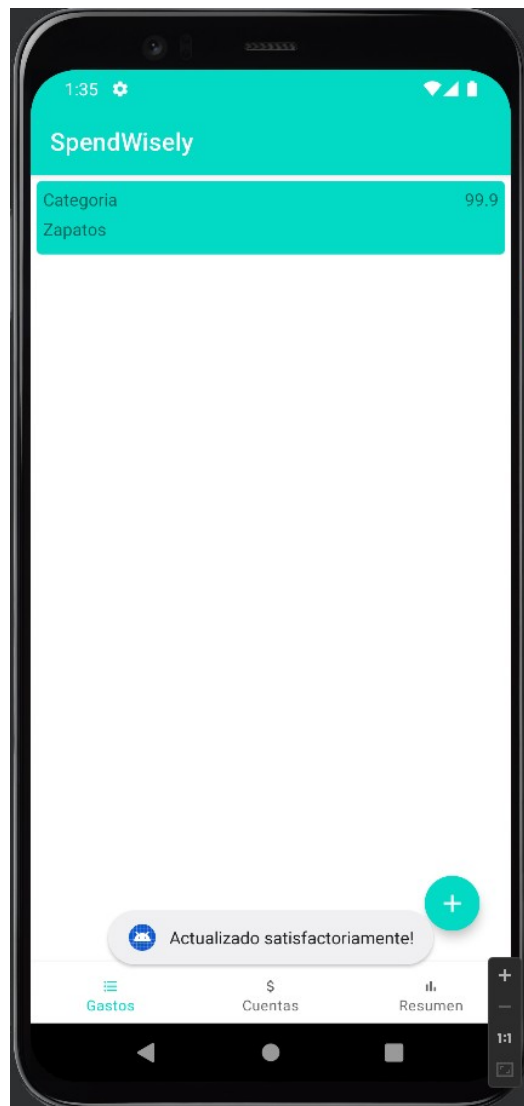


Imagen 10

Eliminar un gasto

Para ello mantendremos el tap encima de uno de los gastos y nos aparecerá un diálogo que nos pedirá confirmar el cambio, presionamos que sí y vemos cómo el gasto desaparece.

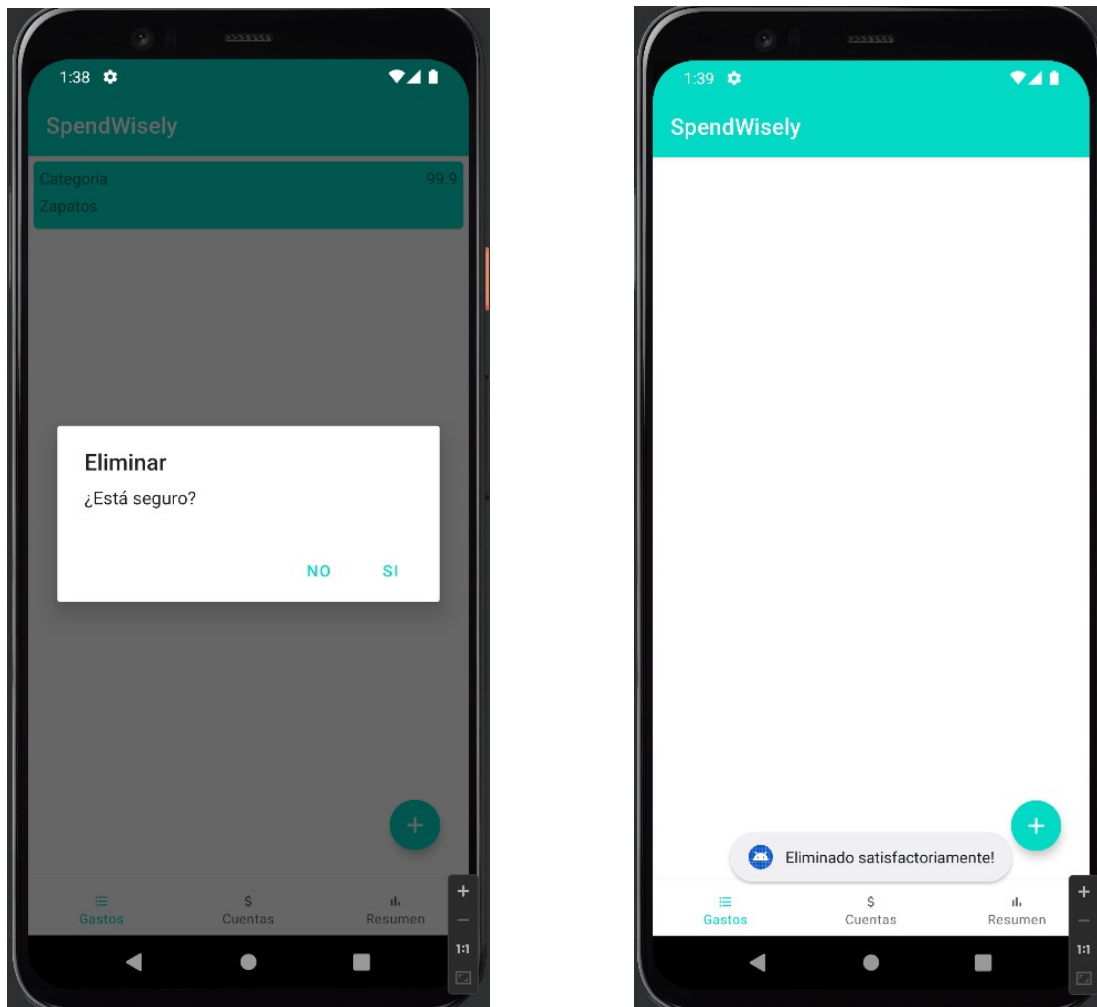


Imagen 11

5.2) Pantalla Cuentas

Desde aquí podremos ver el dinero del que disponemos, además de poder añadir nuevas cuentas de dinero (efectivo, tarjetas).

Cabe destacar que la funcionalidad es muy parecida a la de la pantalla de Gastos.

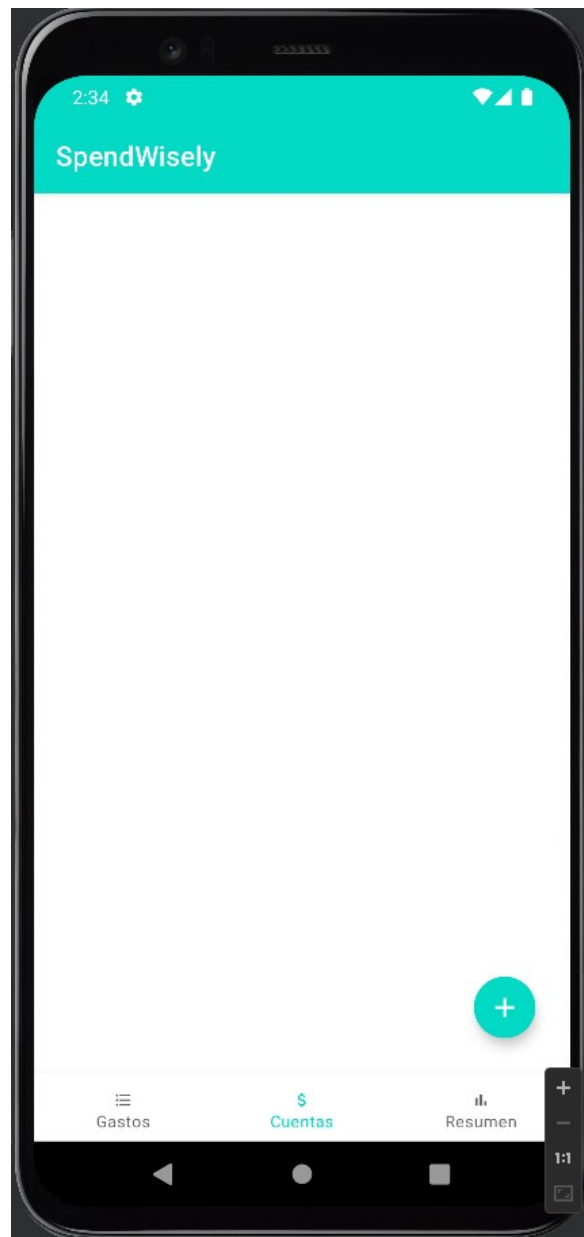


Imagen 12

Añadir billetera

Para ello tendremos que dar un tap en el botón flotante con un + de icono. Dentro de la nueva pantalla deberemos de llenar los campos y darle al botón de GUARDAR.

Si tenemos todos los campos bien, nos saldrá un mensaje diciendo que se ha creado la nueva billetera satisfactoriamente.

Editar Billetera

Para ello, desde la pantalla de Cuentas tendremos que dar un tap en el gasto que queramos modificar. Se abrirá una pantalla para editar el gasto:

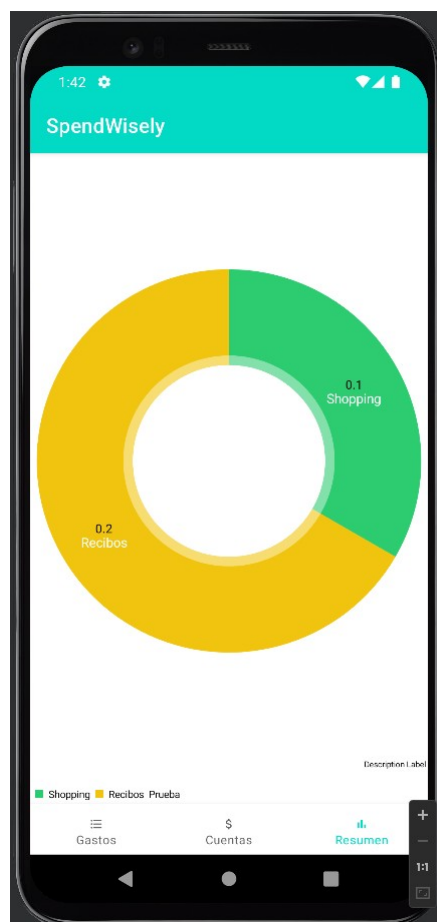
Una vez tengamos hechos los cambios presionaremos en GUARDAR y los cambios se verán reflejados.

Eliminar Billetera

Para ello mantendremos el tap encima de uno de las cuentas y nos aparecerá un diálogo que nos pedirá confirmar el cambio, presionamos que sí y vemos cómo la billetera desaparece.

5.3) Pantalla Resumen

Aquí es donde podremos ver un gráfico en el que se verán reflejados nuestros gastos por categoría:



Este gráfico se actualizará automáticamente de acuerdo a la cantidad de gastos que tengamos en la pantalla de gastos.

6) Enlaces consultados

- <https://developer.android.com/codelabs/android-room-with-a-view-kotlin?hl=es-419#0> (Crear una base de datos room)

- <https://github.com/PhilJay/MPAndroidChart> (MPAndroidChart GitHub)

- Apuntes del aules

7) Mi repositorio GitHub

<https://github.com/Jeikob28/SpendWisely.git>