

Spend Wisely:

Una app para controlar
tu dinero



Alumno: Jacobo Felipe Ruiz Nina

Ciclo: Desarrollo de Aplicaciones Multiplataforma

Curso 2022-2023

Centro: IES La Vereda

Tutora: Jose Miguel Fajardo Asensi

Spend Wisely:

Una app para controlar
tu dinero



Jacobo Felipe Ruiz Nina

Índice

1) Introducción.....	4
1.1) Módulos implicados.....	4
1.2) Breve descripción del proyecto.....	6
2) Diseño.....	6
2.1) Diagrama Entidad-Relación (ER).....	6
2.2) Flujo de la aplicación (Mockups).....	7
3) Recursos utilizados.....	9
3.1) Herramientas Hardware.....	9
3.2) Herramientas Software.....	9
3.3) Complementos utilizados.....	10
4) Plan de trabajo.....	12
4.1) Diagrama de Gant.....	12
4.2) Objetivos.....	13
5) Objetivos conseguidos, descartes y mejoras.....	14
5.1) Objetivos Conseguidos.....	14
5.2) Ideas descartadas.....	14
5.3) Mejoras.....	14
6) Manual de usuario.....	15
6.1) Pantalla Transacciones.....	15
6.2) Pantalla Cuentas.....	24
6.3) Pantalla Resumen.....	26
7) Enlaces consultados y Bibliografía.....	28
8) Mi repositorio GitHub.....	28

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 Entidad-Relación en el que se muestran las tablas utilizadas en la Base de datos del proyecto.
- Uso de DML, DDL y DCL para la gestión de las tablas.

Programación

- Se han creado distintos objetos a lo largo del desarrollo de la aplicación tales como entidades para el mapeo de las tablas de la base de datos, conversores, etc.
- Uso de estructuras de control: if, for, when (switch), etc.
- Se han desarrollado distintas clases e interfaces para el correcto funcionamiento de la aplicación, así como también se ha extendido de clases ya existentes e implementado las interfaces creadas.
- Respecto a estructuras de almacenamiento, se puede destacar el uso de: Listas, Listas Mutables, 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

- Se ha utilizado el sistema de control de versiones de Git además de la creación de un repositorio en GitHub donde se subirían los cambios.

Acceso a datos

- La aplicación conecta con una base de datos local que se crea al instalar la aplicación en el dispositivo Android, desde la cual se gestionarán los datos de esta.
- 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, imágenes, textos, iconos, listas personalizables (recycler view), etc.

Programación multimedia y dispositivos móviles

- Aplicación desarrollada en específico para dispositivos móviles.
- Se controla los distintos estados de la aplicación.
- Toda la interfaz gráfica desarrollada con Android Studio.
- Persistencia de datos gracias al ORM Room.
- Acceso a la cámara del dispositivo y poder tomar fotos con ella.
- Animaciones para algunos elementos de las pantallas.

Programación de servicios y procesos

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

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 tal como dice el nombre de la aplicación, la meta es que el usuario gaste inteligentemente y así no se quede sin dinero.

Respecto a los requisitos para el desempeño correcto de la aplicación basta con que se tenga Android 10 en adelante ya que no se necesita mucha potencia para ejecutarlo.

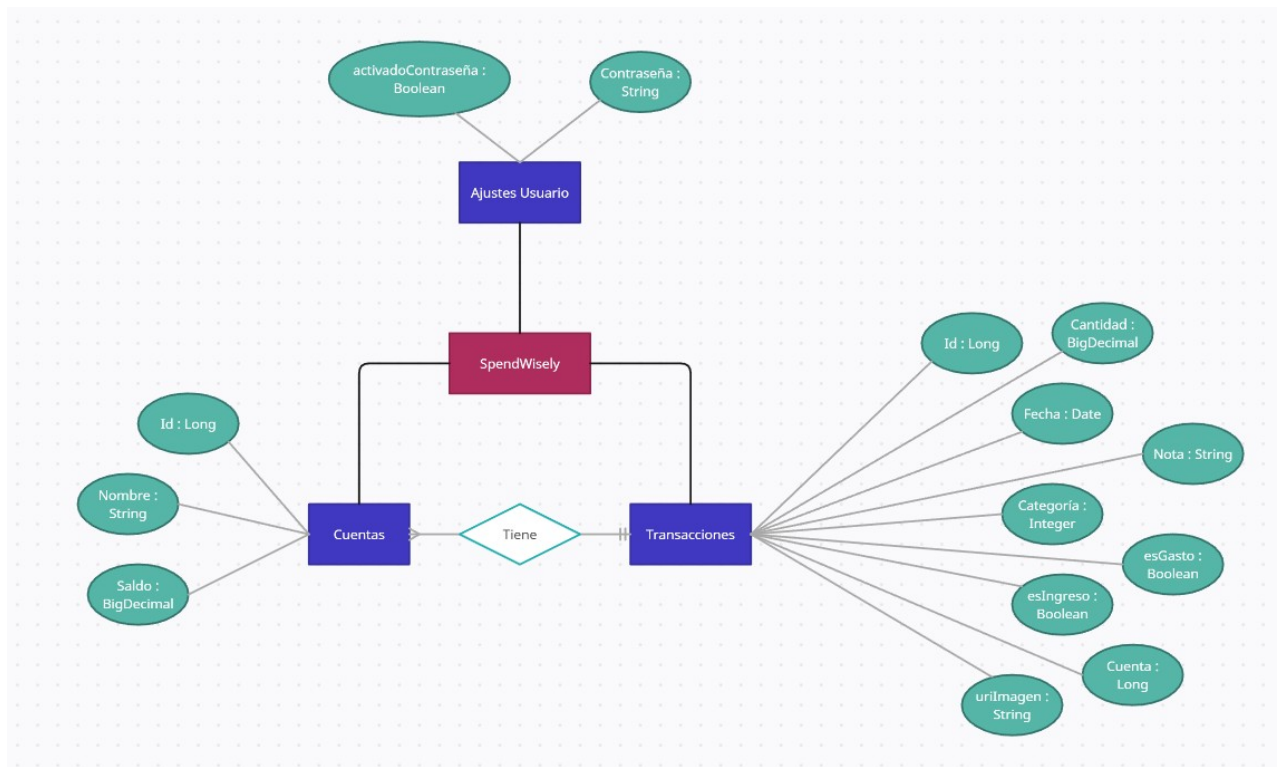
Por otra parte, el uso del proyecto será a nivel personal ya que está pensado que el usuario sea el único que acceda a esta aplicación.

2) Diseño

2.1) Diagrama Entidad-Relación (ER)

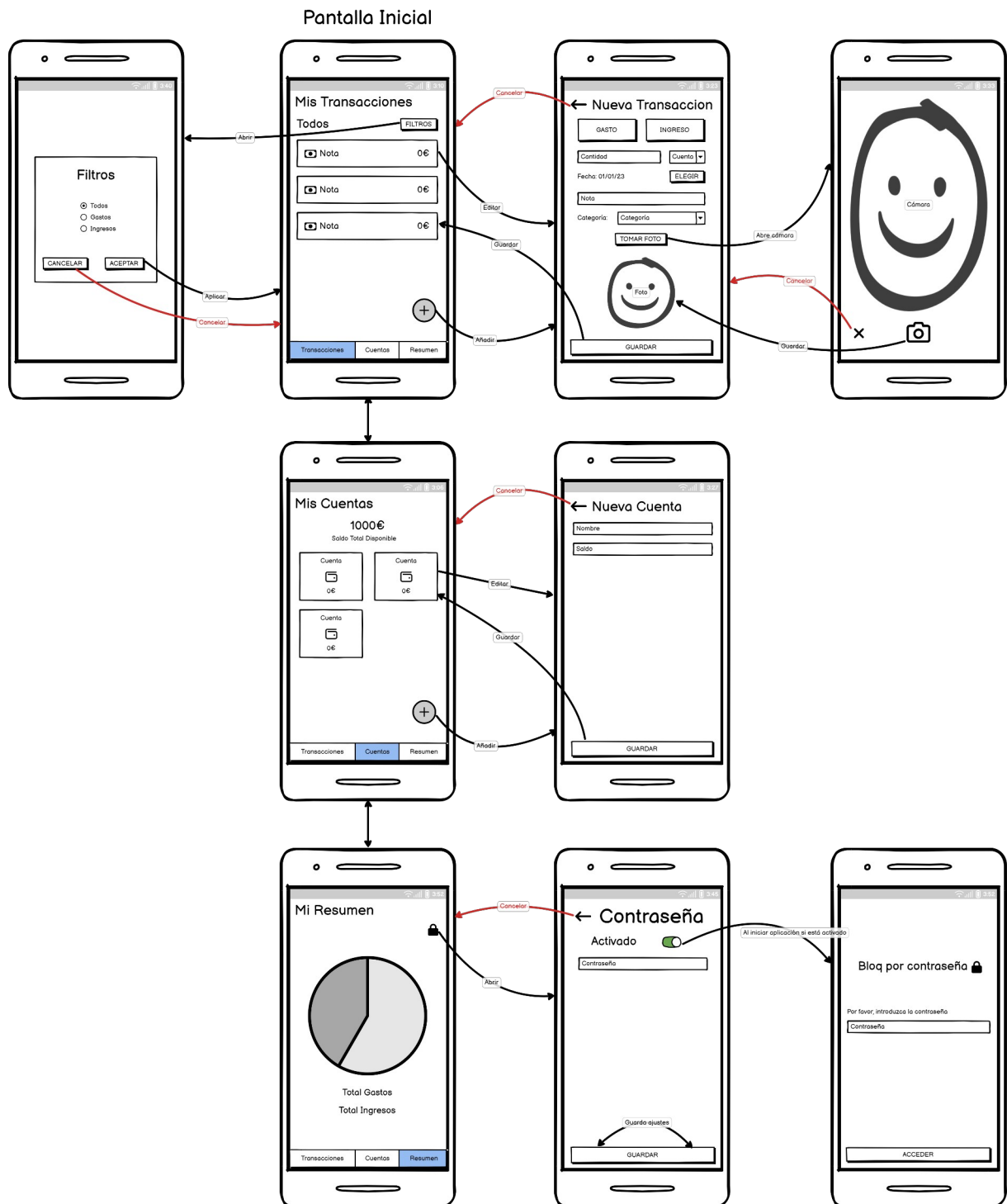
Tras un estudio previo, se determinó que con 2 tablas era suficiente para el correcto funcionamiento de la aplicación.

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 transacciones y podrá navegar a través de todas las pantallas mediante un menú que tendrá en la parte inferior de la aplicación.



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 Flamingo 2022.2.1.

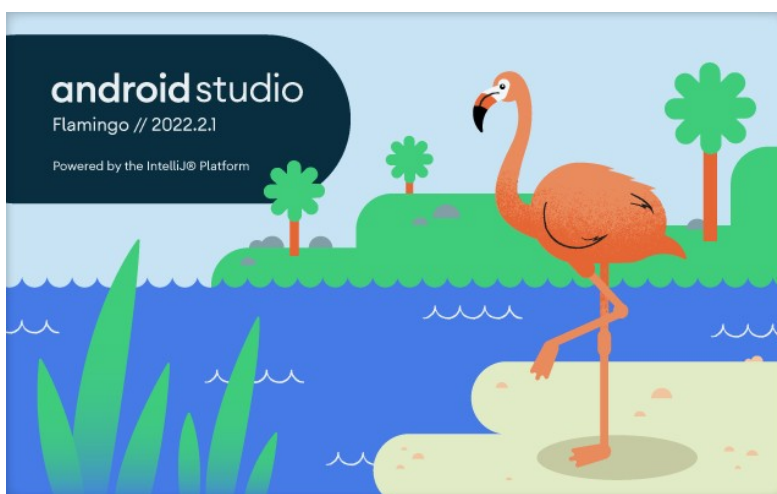


Imagen 2

- **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.

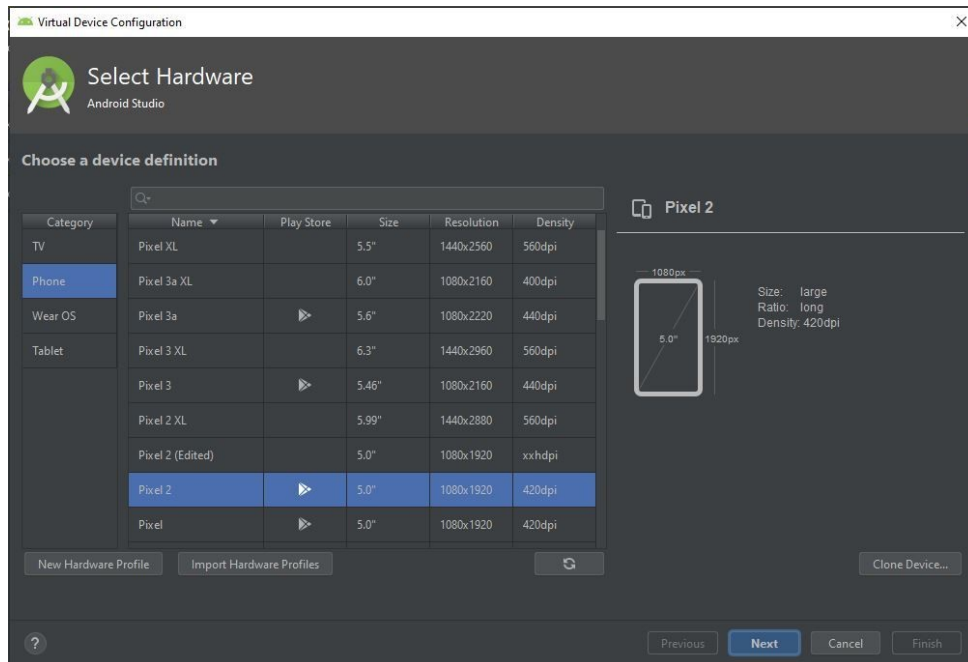


Imagen 3

3.3) Complementos utilizados

- **Material:** Dependencia de elementos visuales recomendados por Google para crear una aplicación.

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

```
dependencies {  
    implementation 'com.google.android.material:material:1.6.0'  
}
```

- **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

Actividad	Inicio	Final	04/04/23	05/04/23	06/04/23	07/04/23	08/04/23	09/04/23	10/04/23
Diseño de las bases de datos	04/04/23	10/04/23							
Diseño de la aplicación (Mockups)	11/04/23	17/04/23							
Desarrollo de la aplicación	18/04/23	15/05/23							
Testing de la aplicación	18/04/23	15/05/23							
Elaboración de memoria de proyecto	11/04/23	29/05/23							
Últimos cambios del proyecto	23/05/23	29/05/23							

11/04/23	12/04/23	13/04/23	14/04/23	15/04/23	16/04/23	17/04/23	18/04/23	...	22/05/23	23/05/23	24/05/23	25/05/23	26/05/23	27/05/23	28/05/23	29/05/23

Imagen 5

4.2) Objetivos

- Que el usuario pueda controlar su dinero de una forma correcta.
- Una aplicación que consiga almacenar registros del usuario (transacciones).
- 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) Objetivos conseguidos, descartes y mejoras

5.1) Objetivos Conseguidos

- Se logra insertar, modificar y eliminar una transacción en la aplicación junto con sus desencadenantes (ajustar saldo de las cuentas, guardarlos en base de datos, modificar gráficas, etc.)
- Se consigue filtrar las transacciones por tipo (Todas, Gastos e Ingresos).
- Se logra insertar y modificar cuentas en la aplicación y sus desencadenantes (verificar si una cuenta tiene transacciones, guardarlas en base de datos, etc.)
- Se puede acceder a la cámara del dispositivo con el fin de adjuntar una imagen a una transacción.
- Las gráficas se construyen a medida de las transacciones que se vayan creando.
- Se puede securizar la aplicación mediante una contraseña.

5.2) Ideas descartadas

- En un principio se pensaba implementar un login para el usuario pero tras analizarlo mejor se llegó a la conclusión de al ser una aplicación pensada para el uso individual y por lo tanto, de acceso personal, sería más fácil y menos tedioso para el usuario acceder mediante una contraseña, de por ejemplo 4 dígitos, que introduzca.

5.3) Mejoras

Por supuesto se puede mejorar la aplicación implementando funcionalidad tal como:

- Filtrar por periodos de tiempo (días, meses, años, etc)
- Mejorar el aspecto visual de la aplicación. (colores, iconos, etc)
- Quizá un scanner de barras para guardar el código de algunos productos dentro de las notas de las transacciones.

6) Manual de usuario

6.1) Pantalla Transacciones

Al abrir la aplicación lo primero que veremos será la pantalla de transacciones en la cual podremos ver todas las transacciones que vayamos introduciendo:

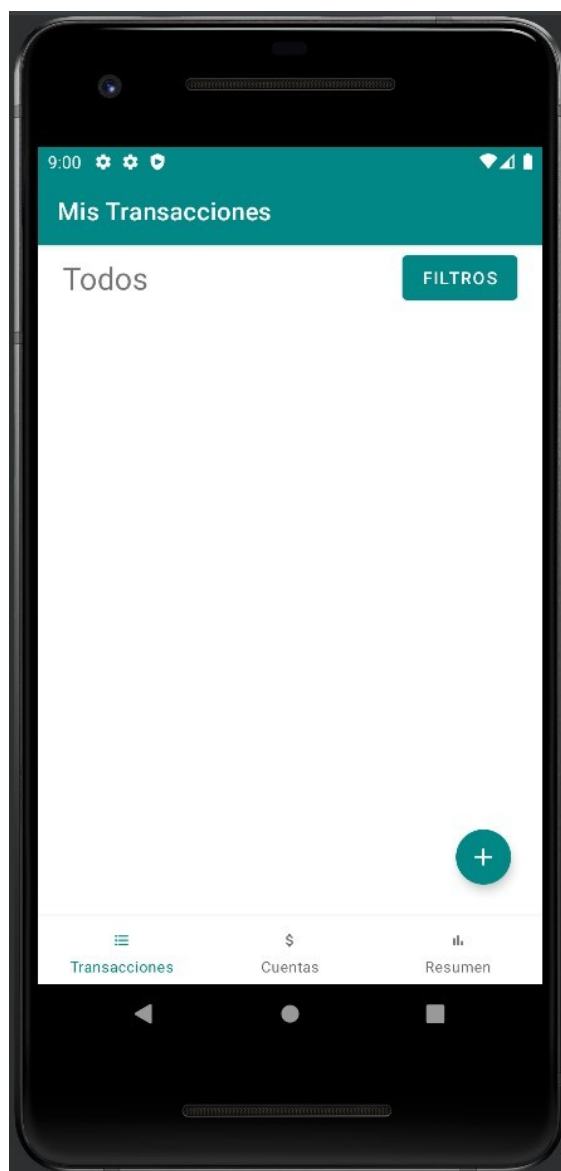


Imagen 6

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 nueva transacción

Desde la pantalla de Transacciones 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 una nueva transacción de tipo Gasto o Ingreso, 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:

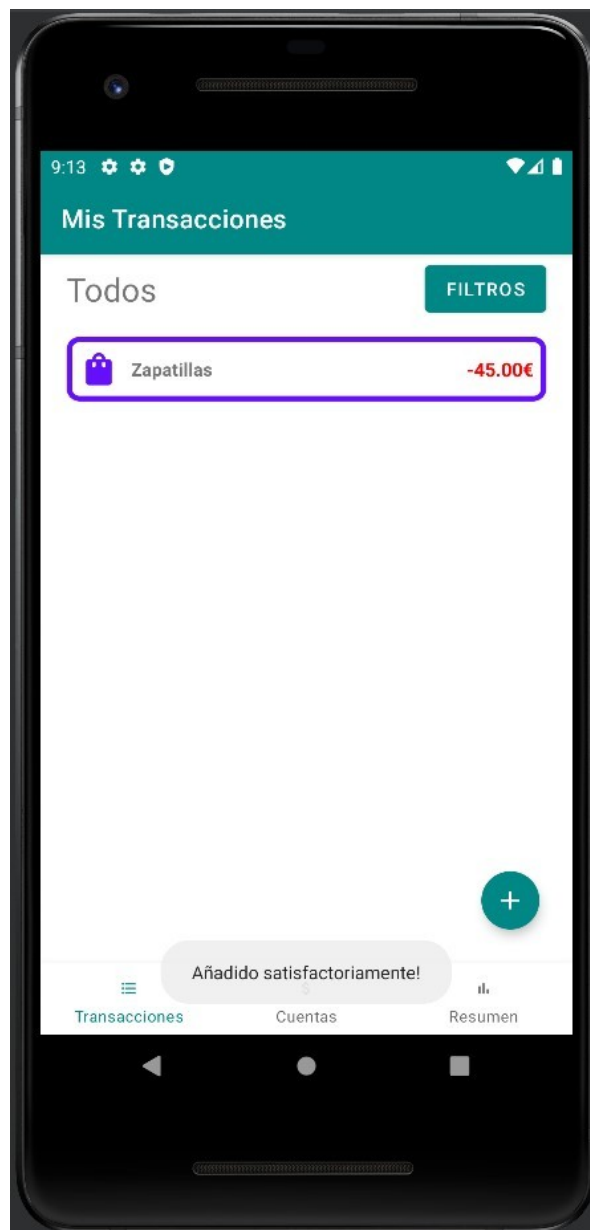


Imagen 8

Filtros

Una vez tengamos unas cuantas transacciones ya creadas, podremos aplicar filtros para poder clasificarlas mejor, para ello es tan simple como dar un tap en el botón Filtros y aplicarlos:

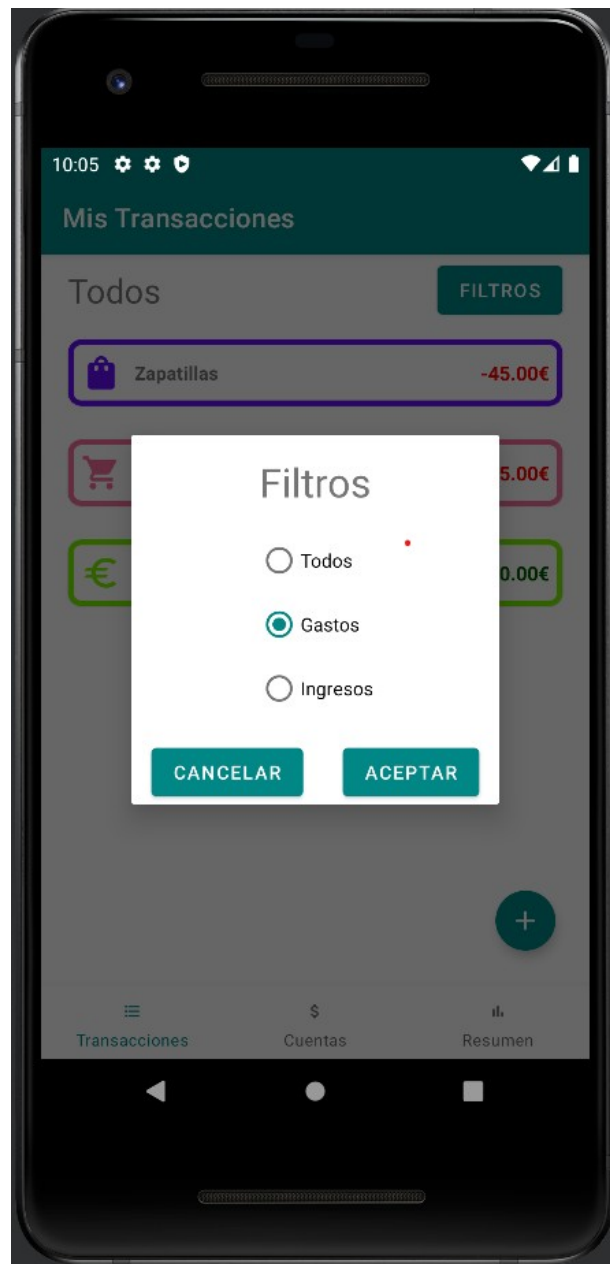


Imagen 9

Editar transacción

Para ello, desde la pantalla de Transacciones tendremos que dar un tap en la transacción que queramos modificar. Se abrirá una pantalla para editarla:



Imagen 10

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:

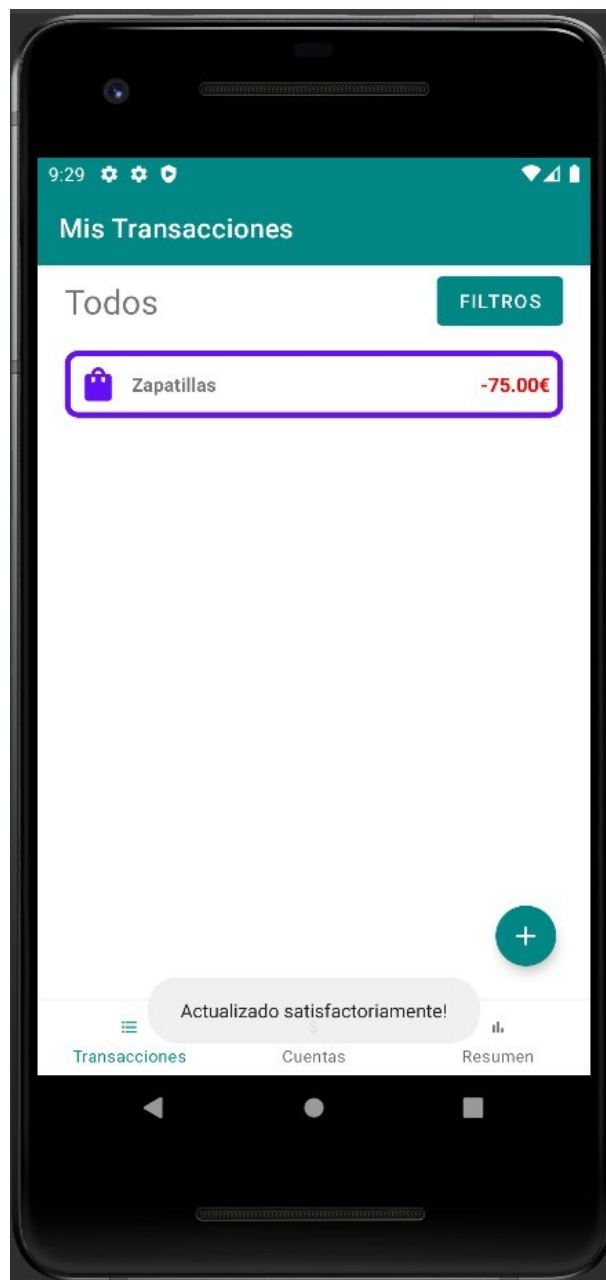


Imagen 11

Adjuntar una foto

Podemos adjuntar una foto a nuestra transacción, para ello cuando creamos o editamos una transacción debemos de dar un tap sobre el botón Tomar Foto, este nos pedirá permisos si no se los hemos dado antes y luego abrirá la cámara tal que así:

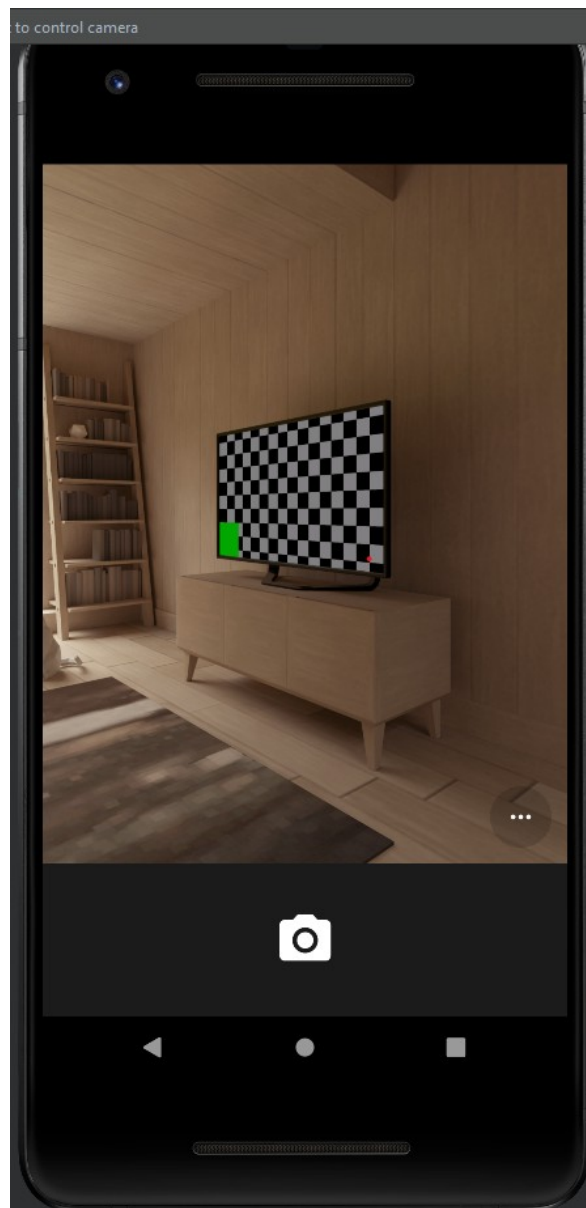


Imagen 12

Si hacemos una foto, regresaremos a la pantalla anterior y podremos ver la foto adjuntada en la parte inferior de la pantalla:

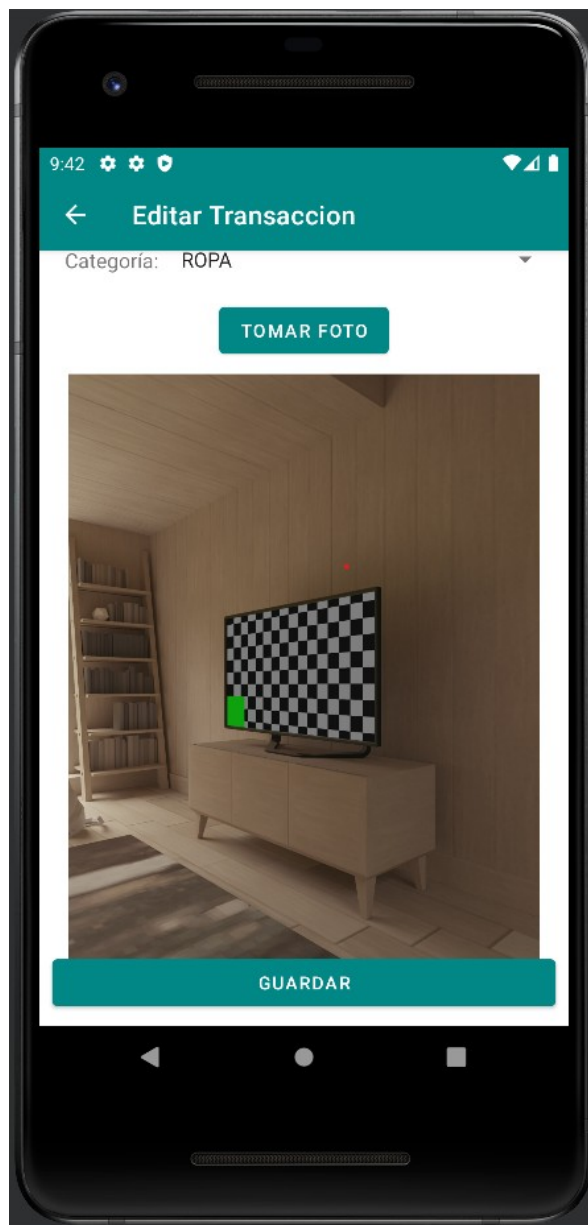


Imagen 13

Eliminar una transacción

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

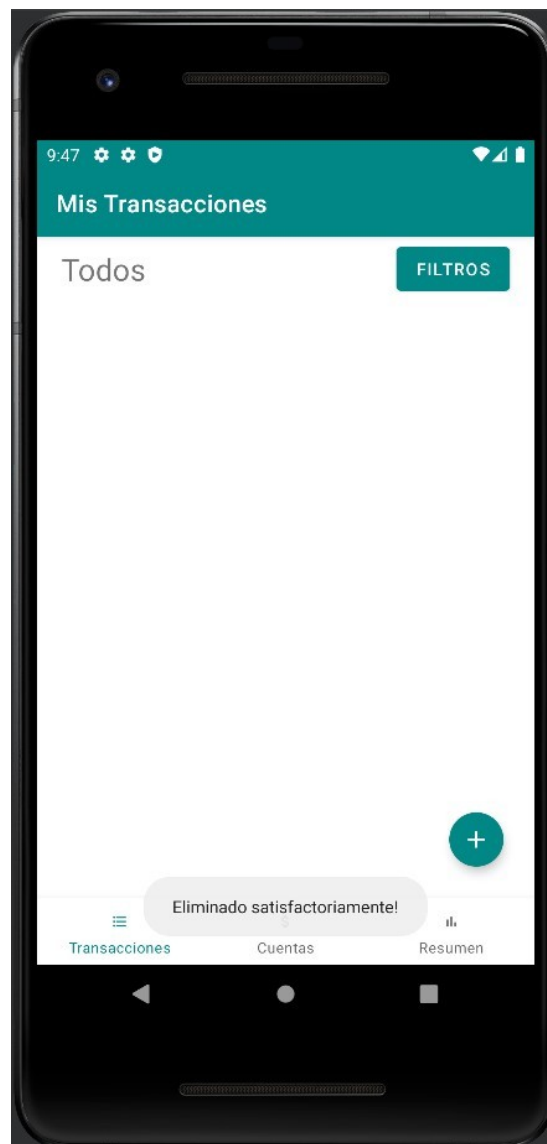


Imagen 14 y 15

6.2) Pantalla Cuentas

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

Cuando se instala la aplicación, se añade por defecto una cuenta de efectivo la cual no se puede eliminar pero sí editar.

Cabe destacar que la funcionalidad es muy parecida a la de la pantalla de Transacciones. Por lo tanto, omitiré algunas capturas.

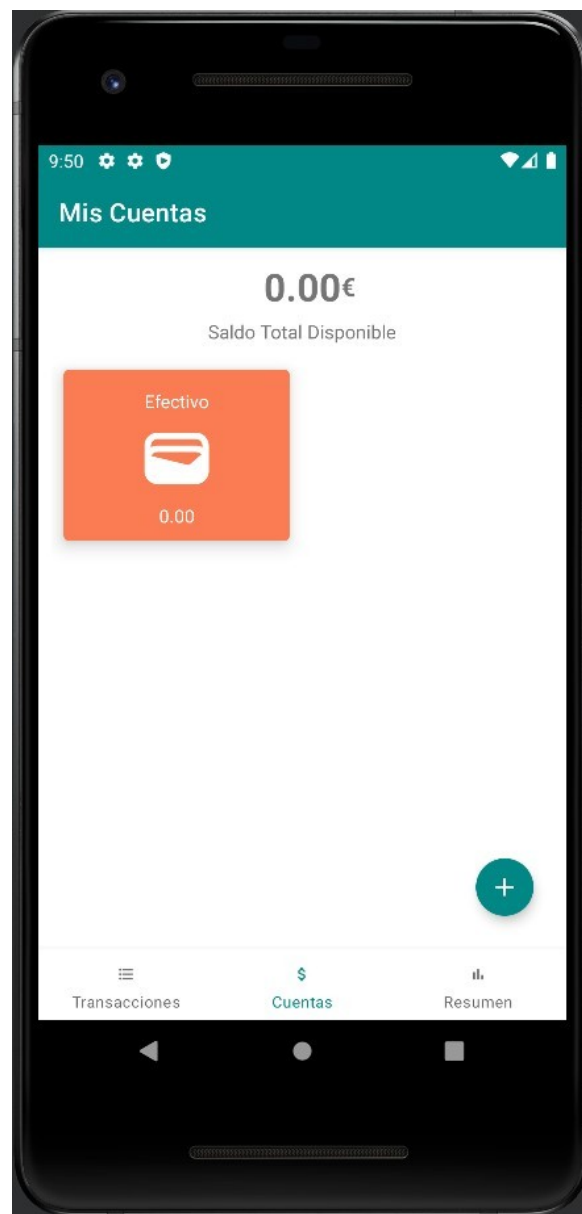


Imagen 16

Añadir cuenta

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 cuenta

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

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

Eliminar cuenta

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 cuenta desaparece.

6.3) Pantalla Resumen

En esta pantalla es donde podremos ver un gráfico en el que se verán reflejados nuestros transacciones por categoría además del total de Gastos e Ingresos existentes.

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



Imagen 17

Pantalla Contraseña

Desde la pantalla de resumen, en la esquina superior derecha podemos encontrar un icono de candado, el cual al pulsarlo nos llevará a la pantalla de contraseña.

Aquí podremos activar o desactivar la opción de introducir contraseña al inicio de la aplicación.

Pantalla Introduce Contraseña

Si se ha activado la opción de contraseña, será la primera pantalla que nos aparezca al iniciar la aplicación en la cual introduciremos la contraseña para poder acceder.



Imagen 18

7) Enlaces consultados y Bibliografía

- <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)
- <https://stackoverflow.com/> (Foro donde resolver dudas y problemas de programación)
- <https://www.youtube.com/> (Ver algunos ejemplos de programación)
- <https://developer.android.com/guide/topics/media/camera?hl=es-419> (Usar cámara)
- Apuntes del aules

8) Mi repositorio GitHub

https://github.com/ieslavereda-projects/22_23_DAM_JACOBO_RUIZ.git

Dentro podremos encontrar las carpetas con los ficheros necesarios para arrancar el proyecto, las carpetas donde se encuentra el código y la lógica de la aplicación es src/main/, dentro de ella podemos encontrar:

- Carpeta java/com.example.spendwisely: dividida en los 4 siguientes paquetes:

1. data → Se encuentra las clases necesarias para el ORM de Room Database (entidades, DAOs, repositorios, conversores, etc)
2. fragmentos → Paquete con los fragmentos de las distintas pantallas: Transacciones, Cuentas y Resumen.
3. otros → Aquí podemos encontrar algunas clases necesarias para la lógica del proyecto como por ejemplo Categoria desde donde se obtiene las categorías de las Transacciones.

4. activities → Como bien indica su título, paquete donde encontramos las activities necesarias para mi aplicación.

También encontramos:

- Una carpeta Documentos donde encontraremos la Memoria del proyecto y un .apk para poder instalar la aplicación en un dispositivo Android.