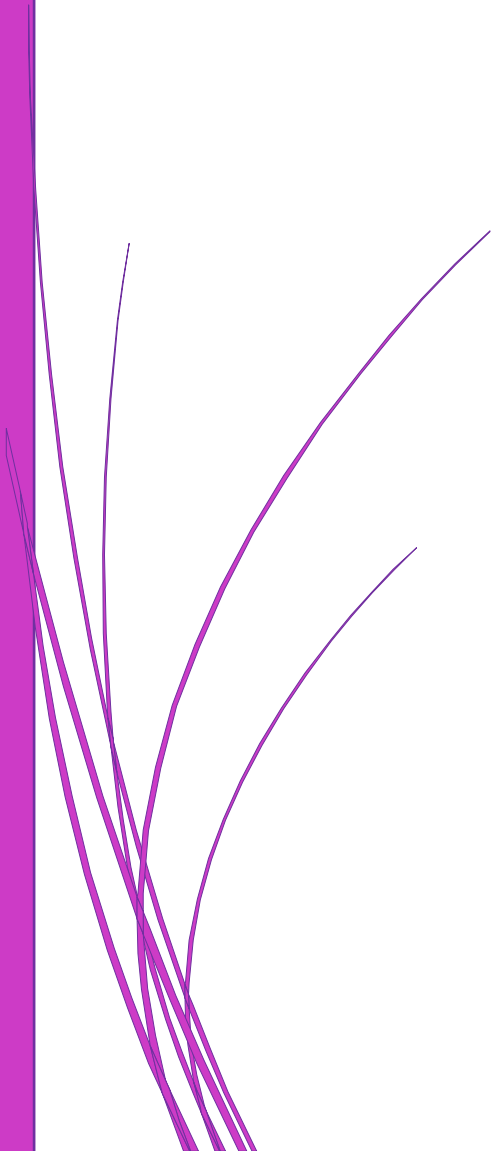




CURSO 2022-2023

# PROYECTO INTEGRADO

ANDROID



Raquel Casares De la Poza  
JUNIO 2023

## ÍNDICE

1. Estudio preliminar del problema.....	Página 4
1.1. Descripción preliminar del sistema	
1.2. Funcionalidades básicas	
1.3. Viabilidad del sistema	
1.4. Estimación (inicial) del coste del sistema	
1.4.1. Recursos humanos	
1.4.2. Recursos hardware	
1.4.3. Recursos software	
2. Análisis y diseño de la aplicación.....	Página 7
2.1. Datos	
2.1.1. Esquema/s entidad-relación	
2.1.2. Traslado a un esquema relacional	
2.1.2.1. Normalización	
2.1.3. Elección justificada de un SGBD relacionado concreto	
2.1.4. Traslado a un esquema físico relacional	

2.1.5. Restricciones no implementables en el modelo de datos que  
deberá contemplar la aplicación

## 2.2. Funciones

2.2.1. Diagramas de flujos de datos

2.2.2. Diccionario de datos

## 3. Implementación del sistema físico..... Página 11

3.1. Elección justificada del hardware en relación calidad/precio

3.2. Elección justificada de los diferentes sistemas operativos

## 4. Planes de pruebas..... Página 12

4.1. Plan de pruebas de caja blanca

4.2. Informe de pruebas de caja blanca

4.3. Plan de pruebas de caja negra

4.4. Informe de pruebas de caja negra

## 5. Manuales..... Página 16

5.1. Manual técnico

5.2. Manual de instalación

5.3. Manual de usuario

5.4. Manual de administración

6. Conclusiones finales..... Página 33

6.1. Grado de consecución de los objetivos fijados

6.2. Posibles mejoras o ampliaciones que implementar en el futuro

7. Bibliografía..... Página 34

## **1. ESTUDIO PRELIMINAR**

### **1.1. Descripción preliminar del sistema**

El sistema es una aplicación móvil diseñada para la gestión de tiendas, y las compras realizadas en dichas tiendas. Su objetivo principal es proporcionar a los usuarios una herramienta eficiente y fácil de usar para administrar el registro y los gastos mensuales de sus compras.

#### **Características principales del sistema:**

**Registro de tiendas:** Permite a los usuarios crear y registrar información básica de sus tiendas.

**Registro de compras:** Permite ingresar los detalles de cada compra, como el nombre de la tienda, el saldo gastado y la fecha de la compra. Esto permite mantener un registro preciso de todas las compras.

**Cálculo del total de gastos:** Realiza el cálculo automático del total de gastos realizados en el mes actual además de poder consultar un período específico de compras, con un mes y un año.

**Modificación y eliminación de tiendas:** Los usuarios tienen la capacidad de realizar modificaciones en la información de las tiendas registradas. También pueden eliminar una tienda de la lista en caso de que ya no sea relevante.

**Modificación y eliminación de compras:** Permite editar o eliminar compras registradas previamente en caso de errores o cambios en los datos.

**Interfaz de usuario intuitiva:** Ofrece una interfaz fácil de usar con formularios y controles claros para facilitar el registro y la visualización de la información.

El sistema se desarrolla utilizando tecnologías móviles, como Android, y se integra con una base de datos centralizada para almacenar y recuperar la información de las tiendas y compras. Además, se utiliza una API REST para la comunicación entre la aplicación móvil y el servidor.

### **1.2. Funcionalidades básicas**

El proyecto va a consistir en crear una aplicación llamada **MisComprasApp** con el programa Android Studio.

Trabajaremos con el almacenamiento remoto montando la API.

Al iniciar la aplicación, se mostrará un listado con todas las compras realizadas en el mes actual con el importe total gastado.

Tiene habilitado un botón de “Volver” para ir de nuevo al listado inicial con las compras del mes actual. Tanto en el listado de compras de la pantalla principal, como en el listado de las tiendas, una pulsación corta sobre un elemento supone la edición, pero una pulsación larga debe mostrar un mensaje de confirmación de borrado. Si se acepta, se

borrará el elemento en cuestión. El CRUD completo, por lo tanto, queda de la siguiente manera:

- Consulta de compras en listado en la pantalla principal.
- Alta de compra, botón + en pantalla principal.
- Edición/Modificación de compras, pulsando en el mismo en el listado de compras del mes en el que se encuentre.
- Borrado de compra, con pulsación larga sobre el mismo en la pantalla principal.
- Consulta de tiendas, pulsando el botón tiendas.
- Alta de tienda, botón + en pantalla del listado de tiendas.
- Edición/Modificación de tienda, pulsando en el mismo.
- Borrado de tienda, con pulsación larga sobre el mismo en el listado de compras del mes en el que se encuentre.

Estas funcionalidades básicas ofrecen a los usuarios la capacidad de gestionar eficazmente sus tiendas y compras, registrar información precisa y realizar un seguimiento de sus gastos mensuales.

### **1.3. Viabilidad del sistema**

Viabilidad técnica: La aplicación ha sido desarrollada utilizando Android Studio y Java, lo cual es una combinación común y sólida para el desarrollo de aplicaciones Android. Estos recursos técnicos están ampliamente disponibles y se pueden encontrar numerosos recursos y documentación para ayudar en el proceso de desarrollo.

Viabilidad económica: Es importante considerar el potencial de ingresos de la aplicación, como publicidad, compras dentro de la aplicación u otros medios. También es necesario evaluar los costos asociados con el desarrollo y mantenimiento de la aplicación, así como los posibles beneficios económicos que se esperan obtener.

Viabilidad operativa: Es necesario asegurarse de que se cuenten con los recursos humanos necesarios para el desarrollo, mantenimiento y soporte de la aplicación.

Viabilidad legal y regulatoria: Al utilizar una API externa y manejar datos de usuarios y transacciones, es fundamental cumplir con las leyes y regulaciones aplicables en términos de privacidad, protección de datos y seguridad de la información. Asegurarse de implementar medidas adecuadas para proteger la información de los usuarios es esencial.

## **1.4. Estimación (inicial) del coste del sistema**

### **1.4.1. Recursos humanos:**

La estimación del coste de los recursos humanos dependerá de los roles y las tarifas por hora de los profesionales involucrados en el desarrollo del sistema. A continuación, se presenta una estimación aproximada de los roles necesarios y las horas de trabajo estimadas:

- Desarrollador de software: 200 horas de trabajo estimadas.
- Diseñador de interfaz de usuario: 80 horas de trabajo estimadas.
- Analista de sistemas: 40 horas de trabajo estimadas.
- Tester/QA: 40 horas de trabajo estimadas.

Es importante tener en cuenta que estas estimaciones pueden variar según la complejidad del sistema y las tarifas por hora de los profesionales contratados.

### **1.4.2. Recursos hardware:**

Los recursos hardware necesarios para el sistema incluye:

- Servidor: Un servidor para alojar la base de datos y permitir el acceso remoto a la aplicación.
- Dispositivos móviles: Se necesitarán dispositivos móviles para probar y ejecutar la aplicación.

El coste de estos recursos puede variar según las características y la capacidad requerida.

### **1.4.3. Recursos software:**

Los recursos software necesarios para el sistema incluye:

- Entorno de desarrollo integrado (IDE): Se requerirá un IDE para el desarrollo de la aplicación móvil y para el desarrollo del backend.
- Bibliotecas y frameworks: Bibliotecas y frameworks específicos para agilizar el desarrollo y proporcionar funcionalidades adicionales.
- Sistema de gestión de base de datos: Base de datos, se requerirá un sistema de gestión de base de datos compatible.

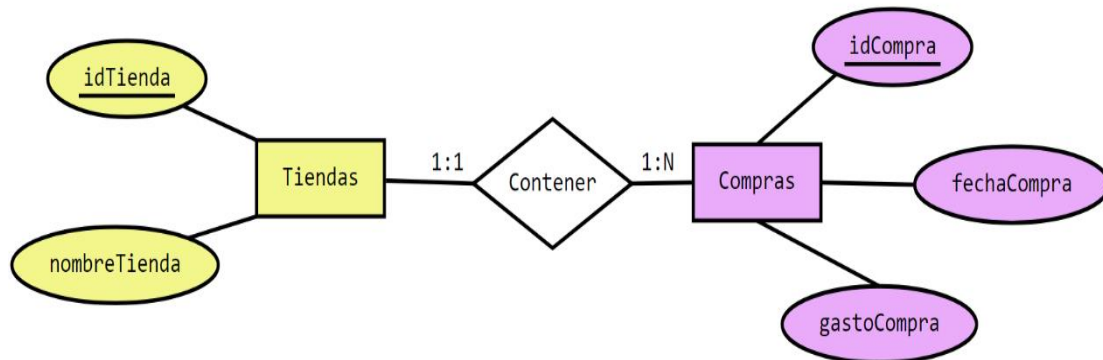
El coste de estos recursos puede variar según las licencias y las opciones elegidas.

Es importante tener en cuenta que estas estimaciones son preliminares y pueden cambiar a medida que se realice un análisis más detallado del proyecto y se tengan en cuenta otros factores, como el tiempo y los costes indirectos.

## **2. ANÁLISIS Y DISEÑO DE LA APLICACIÓN**

### **2.1. Datos**

#### **2.1.1. Esquema Entidad-Relación (ERD)**



#### **2.1.2. Traslado a un Esquema Relacional (ER)**

Tiendas (#idTienda, nombreTienda)

Compras (#idCompra, fechaCompra, gastoCompra, idTiendaFK)

##### **2.1.2.1 Normalización**

La normalización es un proceso utilizado en el diseño de bases de datos para eliminar la redundancia de datos y garantizar la integridad y eficiencia de la información almacenada. A continuación, se presenta una propuesta de normalización para el esquema relacional mencionado anteriormente:

Primera Forma Normal (1NF):

- Cada tabla debe tener una clave primaria única.
- Todos los atributos de una tabla deben contener valores atómicos, es decir, no deben ser multivaluados ni repetidos.
- No debe haber dependencias funcionales parciales, lo que significa que todos los atributos de una tabla deben depender completamente de la clave primaria.

Este diseño cumple con la 1NF, ya que cada tabla tiene una clave primaria única y los atributos son atómicos. Además, en la tabla Compra, todos los atributos dependen completamente de la clave primaria.



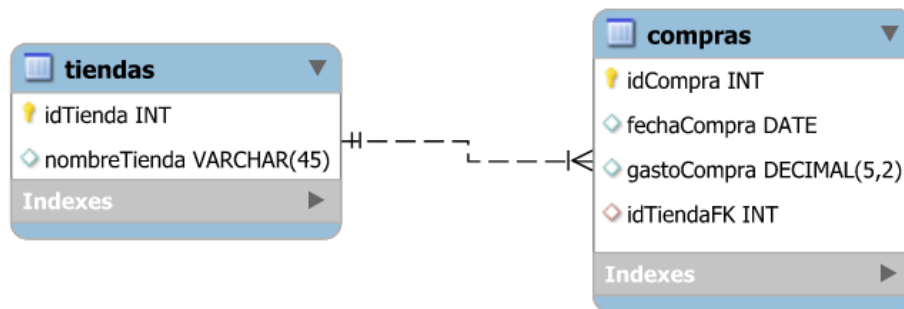
### **2.1.3. Elección justificada de un SGBD relacional concreto**

MySQL es un Sistema de Gestión de Bases de Datos Relacional (SGBD) muy popular y ampliamente utilizado en el desarrollo de aplicaciones web y móviles. Algunas de las razones por las que MySQL es una opción frecuente son:

1. Rendimiento y escalabilidad: MySQL es conocido por su buen rendimiento y capacidad para manejar grandes volúmenes de datos. Ofrece opciones de optimización y configuración para mejorar el rendimiento de las consultas y transacciones.
2. Facilidad de uso: MySQL es relativamente fácil de instalar, configurar y utilizar. Su sintaxis SQL es intuitiva y fácil de aprender, lo que facilita el desarrollo y la gestión de la base de datos.
3. Comunidad y soporte: MySQL cuenta con una amplia comunidad de usuarios y desarrolladores, lo que significa que puedes encontrar una gran cantidad de recursos, documentación, tutoriales y ejemplos en línea. Además, MySQL ofrece soporte técnico a través de sus servicios de suscripción.
4. Amplia compatibilidad: MySQL es compatible con varios sistemas operativos, incluyendo Windows, Linux y macOS. También es compatible con una amplia gama de lenguajes de programación, lo que facilita la integración con diferentes aplicaciones y tecnologías.
5. Seguridad: MySQL ofrece funciones de seguridad sólidas, incluyendo la autenticación de usuarios, encriptación de datos, control de acceso y auditoría. Esto es especialmente importante cuando se trata de proteger la integridad de los datos sensibles.
6. Costo: MySQL es una opción de código abierto, lo que significa que es gratuito para su uso. Sin embargo, también ofrece una versión comercial (MySQL Enterprise Edition) que incluye características adicionales y soporte técnico avanzado para entornos empresariales.

En resumen, MySQL es una elección sólida para un SGBD relacional debido a su rendimiento, facilidad de uso, amplia compatibilidad y soporte comunitario. Pudiendo aprovechar su funcionalidad y características para implementar y gestionar eficientemente la base de datos del sistema de gestión de tiendas y compras.

#### **2.1.4. Traslado a un esquema físico relacional:**

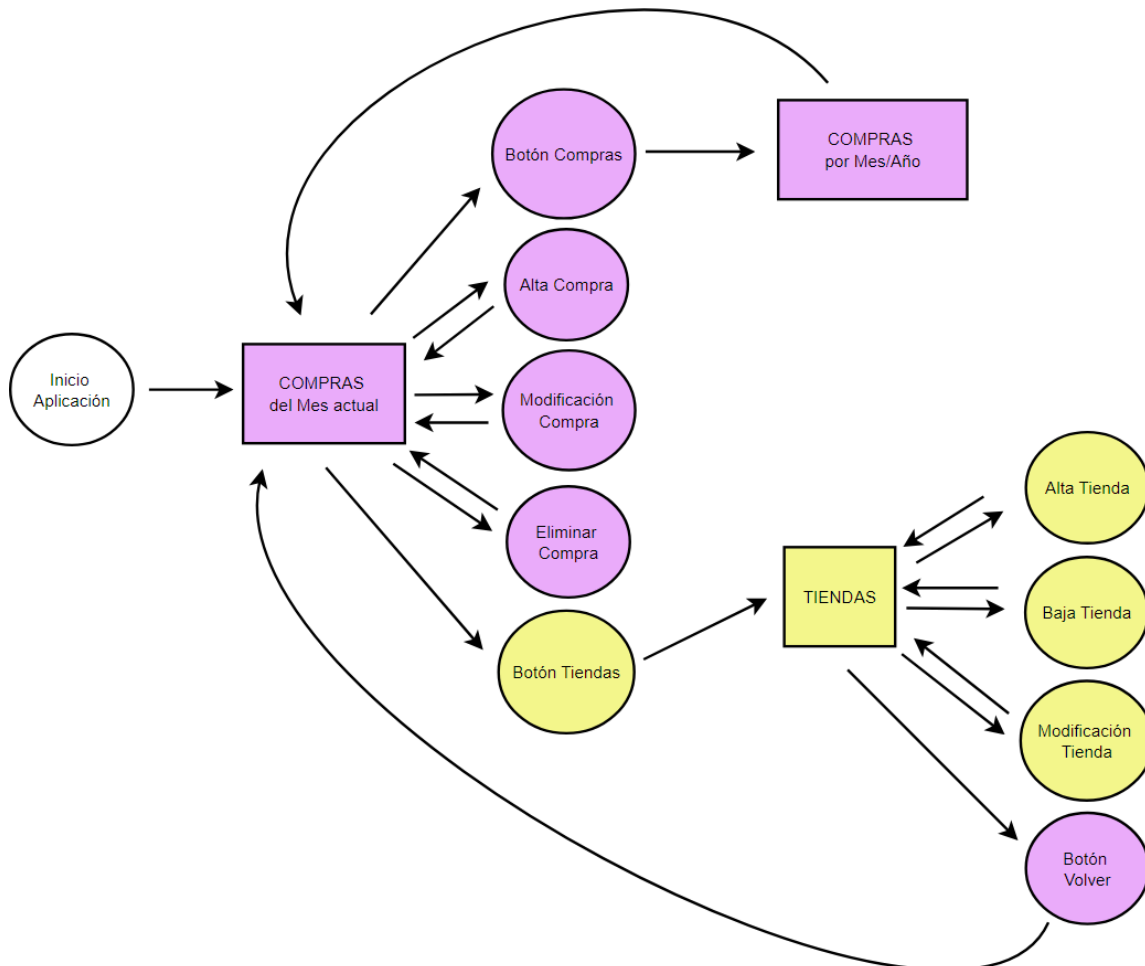


#### **2.1.5. Restricciones no implementables en el modelo de datos que deberá contemplar la aplicación.**

1. Reglas de flujo de trabajo: Contiene reglas de flujo de trabajo o procesos específicos que van más allá de la estructura de los datos, como aprobaciones, notificaciones o acciones específicas que deben ocurrir en determinados estados o situaciones, estas reglas están implementadas en la lógica de la aplicación.
2. Reglas de presentación y validación en la interfaz de usuario: Las reglas de presentación y validación específicas en la interfaz de usuario, como formatos de entrada, validaciones en tiempo real o validaciones dependientes de otros campos, estas reglas están implementadas en la lógica de la interfaz de usuario.

## **2.2. Funciones**

### **2.2.1. Diagrama de flujo de datos**



### **2.2.2. Diccionario de datos**

Sirve para definir datos y procesos. Consiste en una lista organizada de todos los elementos presentes en el sistema, con definiciones precisas y rigurosas que permitan al usuario y al analista tener una base común de entendimiento. También se los conoce como Metabase. El Administrador de la Base de Datos es el responsable de su creación y mantenimiento.

1. Tabla: Tiendas
  - Nombre: VARCHAR (45)
2. Tabla: Compras
  - ID: INTEGER (Primary Key)
  - ID de Tienda: INTEGER (Foreign Key a la tabla Tiendas)
  - Gasto: DECIMAL (5, 2)
  - Fecha: DATE

A continuación, se detallan los elementos de datos de cada tabla:

Tabla: Tiendas

- Nombre: El nombre de la tienda.

Tabla: Compras

- ID: Identificador único de la compra.
- ID de Tienda: Identificador de la tienda a la que pertenece la compra (clave externa).
- Gasto: El monto gastado en la compra.
- Fecha: La fecha en la que se realizó la compra.

### **3. IMPLEMENTACIÓN DEL SISTEMA FÍSICO**

#### **3.1. Elección justificada del hardware en relación calidad/precio**

1. Dispositivo móvil: Para ejecutar la aplicación en Android 5.0 (Lollipop) o superior, se recomienda utilizar un dispositivo móvil compatible con este sistema operativo. Optar por teléfonos inteligentes o tabletas que cumplan con los requisitos mínimos de hardware y tengan un buen rendimiento en términos de velocidad y capacidad de respuesta.
2. Procesador: Para un rendimiento óptimo, se sugiere utilizar un dispositivo con un procesador de al menos 1.5 GHz o superior. Esto permitirá que la aplicación funcione sin problemas y realice cálculos rápidos, especialmente al realizar operaciones como el cálculo de gastos mensuales.
3. Memoria RAM: Se recomienda disponer de al menos 2 GB de RAM en el dispositivo para garantizar un rendimiento fluido de la aplicación. Esto permitirá que la aplicación se ejecute sin problemas y pueda manejar grandes cantidades de datos de forma eficiente.

4. Almacenamiento: El dispositivo debe tener suficiente espacio de almacenamiento para la instalación inicial de la aplicación, lo cual requerirá al menos 50 MB de espacio libre. Hay que tener en cuenta que este espacio puede aumentar a medida que se agreguen más datos a la aplicación, como registros de compras y datos de tiendas. Si es posible, elegir un dispositivo con una cantidad adecuada de almacenamiento interno o con la capacidad de expandir el almacenamiento mediante tarjetas de memoria externas.
5. Conexión a Internet: La aplicación requiere una conexión a Internet para descargar datos de las tiendas, sincronizar compras realizadas y realizar cálculos mensuales de gastos.
6. Permisos: Asegúrate de que el dispositivo tenga los permisos necesarios habilitados para el funcionamiento adecuado de la aplicación. Esto incluye permisos de acceso a Internet y acceso al almacenamiento.

### **3.2. Elección justificada de los diferentes sistemas operativos**

Android: Android es un sistema operativo móvil ampliamente utilizado con una gran base de usuarios. Si tu aplicación está dirigida a dispositivos móviles, Android es una elección lógica debido a su popularidad y alcance. Además, Android ofrece un entorno de desarrollo robusto y herramientas de desarrollo ampliamente utilizadas, como Android Studio, que facilitan la creación de aplicaciones nativas para dispositivos Android.

## **4. Planes de pruebas**

### **4.1. Plan de pruebas de caja blanca**

Pruebas de Caja Blanca o Pruebas Estructurales o Pruebas Unitarias: Se centra en el estudio minucioso de la operatividad de una parte del sistema considerando los detalles procedurales (la lógica del sistema).

Vamos a realizar las pruebas de caja blanca a uno de los métodos que aparecen en la aplicación. Método ValidarFecha ().

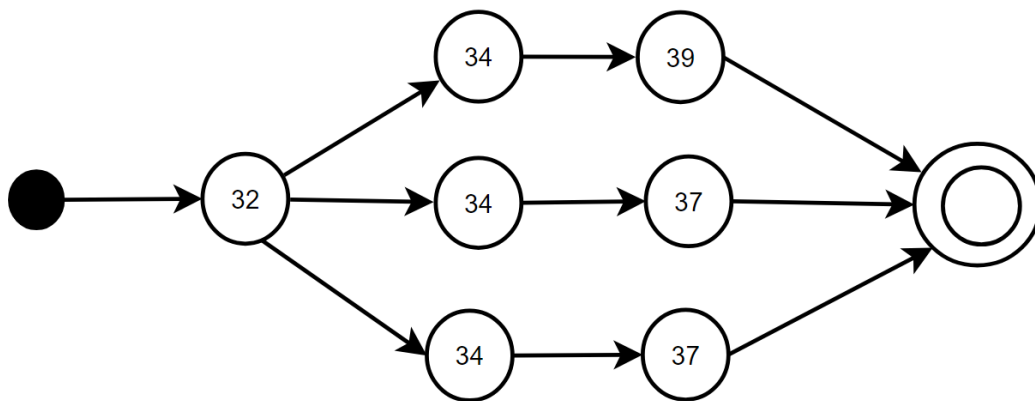
Obtener el grafo de control, el conjunto de caminos básicos o rutas, determinar los casos de prueba, establecer los valores esperados, ejecutar el método con los valores oportunos y comparar resultados.

```

24
25 private static String validarFecha(String fechaEuropea)
26 {
27     SimpleDateFormat formatoEuropeo = new SimpleDateFormat("dd/MM/yyyy");
28     formatoEuropeo.setLenient(false); // Hace que el análisis de la fecha sea estricto
29
30     String f = null;
31     try {
32         Date fecha = formatoEuropeo.parse(fechaEuropea);
33         SimpleDateFormat formatoAmericano = new SimpleDateFormat("yyyy-MM-dd");
34         String fechaAmericana = formatoAmericano.format(fecha);
35         f = fechaAmericana;
36     } catch (Exception e) {
37
38     }
39     return f;
40 }
41

```

### Grafo de recubrimiento



### Complejidad Ciclomática (CC = nº flechas - nº nodos + 2)

$$CC = 10 - 9 + 2 = 3$$

### Caminos y Casos de prueba

1. INICIO 32 34 39 FIN
2. INICIO 32 34 37 FIN
3. INICIO 32 34 37 FIN

<u>Camino</u>	<u>Caso de prueba</u>	<u>Resultado esperado</u>
1	"15/06/2023"	"2023-06-15"
2	"31/02/2023"	"Fecha inválida."
3	"15-06-2023"	"Formato incorrecto."

#### **4.1. Informe de pruebas de caja blanca**

La función **validarFecha()** ha sido sometida a pruebas exhaustivas utilizando la técnica de caja blanca. Se han diseñado casos de prueba para cubrir diferentes rutas y condiciones del código con el objetivo de garantizar un correcto funcionamiento y una gestión adecuada de los errores. A continuación, se presenta un resumen de los resultados de las pruebas realizadas:

1. Prueba de caso exitoso:
  - Entrada: "15/06/2023"
  - Salida esperada: "2023-06-15"
  - Resultado: La prueba fue exitosa. La función devolvió la fecha en formato americano correctamente.
2. Prueba de caso de fecha inválida:
  - Entrada: "31/02/2023"
  - Salida esperada: null
  - Resultado: La prueba fue exitosa. La función manejó correctamente el error de fecha inválida y devolvió null.
3. Prueba de caso de formato incorrecto:
  - Entrada: "15-06-2023"
  - Salida esperada: null
  - Resultado: La prueba fue exitosa. La función manejó correctamente el error de formato incorrecto y devolvió null.
  -

En general, todas las pruebas de caja blanca fueron exitosas y la función **validarFecha ()** demostró un comportamiento correcto y adecuado en diferentes situaciones.

#### **4.1. Plan de pruebas de caja negra**

El plan de pruebas de caja negra se enfoca en probar la funcionalidad del sistema sin conocer los detalles internos de su implementación. El objetivo es validar que el sistema cumpla con los requisitos funcionales y que responda correctamente a diferentes entradas y situaciones. A continuación, se presenta un plan de pruebas de caja negra para el alta de registro de tiendas de la aplicación:

##### **Prueba de registro de tiendas:**

- Verificar que se pueda crear una nueva tienda con un nombre válido.
- Verificar que no se permita crear una tienda sin proporcionar un nombre.
- Verificar que se genere un mensaje de error si se intenta crear una tienda con un nombre que ya existe.

#### **4.1. Informe de pruebas de caja negra**

##### **Pruebas para el Registro de Compras:**

##### **Caso de Prueba 1 - Registro de compra exitoso:**

- Descripción: Se ingresaron datos válidos de una nueva compra, incluyendo el nombre de la tienda, la fecha y el importe gastado.
- Resultado: La compra se registró correctamente en el sistema y se generó un mensaje de confirmación de registro.

##### **Caso de Prueba 2 - Registro de compra sin datos obligatorios:**

- Descripción: Se intentó registrar una compra sin proporcionar uno o más datos obligatorios, como el nombre de la tienda, la fecha o el importe gastado.
- Resultado: El sistema validó la falta de datos obligatorios y generó un mensaje de error indicando que estos campos son requeridos.

##### **Caso de Prueba 3 - Registro de compra con datos inválidos:**

- Descripción: Se intentó registrar una compra con datos inválidos, como una fecha en un formato incorrecto o una tienda que no existe.
- Resultado: El sistema validó los datos inválidos y generó un mensaje de error indicando que se deben proporcionar valores válidos para la fecha y la tienda.



## **5. Manuales**

### **5.1. Manual técnico**

El proyecto va a consistir en crear una aplicación llamada MisComprasApp con el programa Android Studio.

Trabajaremos con el almacenamiento remoto montando la API.

Al iniciar la aplicación, se mostrará un listado con todas las compras realizadas en el mes actual con el importe total gastado.

Habilitar un botón de “Volver” para ir de nuevo al listado inicial con las compras del mes actual. Tanto en el listado de compras de la pantalla principal, como en el listado de las tiendas, una pulsación corta sobre un elemento supone la edición, pero una pulsación larga debe mostrar un mensaje de confirmación de borrado. Si se acepta, se borrará el elemento en cuestión. El CRUD completo, por lo tanto, queda de la siguiente manera:

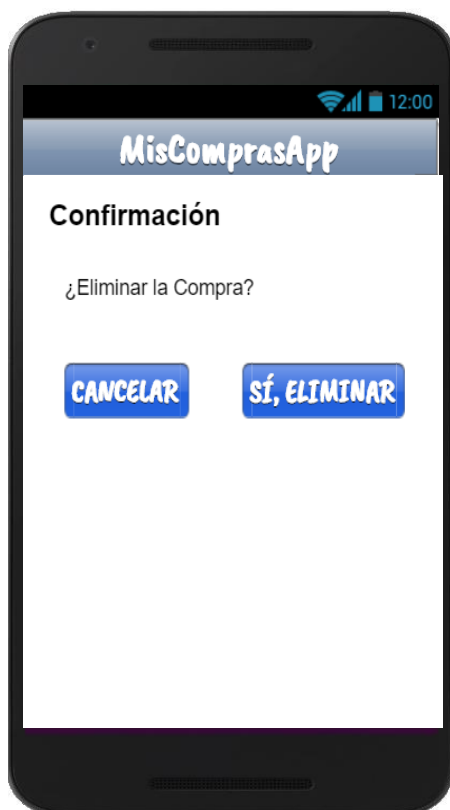
- Consulta de compras en listado en la pantalla principal.
- Alta de compra, botón + en pantalla principal.
- Edición/Modificación de compras, pulsando en el mismo en el listado de compras del mes en el que se encuentre.
- Borrado de compra, con pulsación larga sobre el mismo en la pantalla principal.
- Consulta de tiendas, pulsando el botón tiendas.
- Alta de tienda, botón + en pantalla del listado de tiendas.
- Edición/Modificación de tienda, pulsando en el mismo.
- Borrado de tienda, con pulsación larga sobre el mismo en el listado de compras del mes en el que se encuentre.





CLICK LARGO→ELIMINAR COMPRA

CLICK CORTO→MODIFICAR COMPRA







CLICK LARGO→ELIMINAR TIENDA

CLICK CORTO→MODIFICAR TIENDA



## **5.2. Manual de instalación**

- Instalación de MySQL: Descarga e instala el servidor de base de datos MySQL en el entorno de implementación. Sigue las instrucciones proporcionadas por MySQL para realizar la instalación adecuada.
- Configuración del servidor MySQL: Realiza la configuración inicial del servidor MySQL, incluyendo la asignación de recursos de hardware y configuración de seguridad.
- Creación de la base de datos, creación de tablas y carga de datos iniciales: Se facilita la importación de datos disponibles en MySQL.
- Conexión de la aplicación a la base de datos: Configurar la conexión de tu aplicación móvil con el servidor MySQL.
- Obtener las credenciales de la API: he creado un proyecto en la consola de desarrolladores de Google y obtener las credenciales necesarias para utilizar la API.
- Configurar la API en el proyecto.
- Integrar la API en la aplicación:
- Utilizar los servicios de la API: Una vez configurado y autenticado correctamente la API de Google, se puede comenzar a utilizar sus servicios en la aplicación. Esto incluye el acceso a datos, la realización de operaciones específicas, el envío de solicitudes y la recepción de respuestas según lo especificado en la documentación de la API.

### **Requisitos del sistema:**

1. **Sistema operativo:** Android 5.0 (Lollipop) o superior.
2. **Procesador:** Se recomienda un procesador de al menos 1.5 GHz o superior para un rendimiento óptimo.
3. **Memoria RAM:** Se recomienda al menos 2 GB de RAM para un rendimiento fluido de la aplicación.
4. **Almacenamiento:** La cantidad de almacenamiento requerida dependerá de la cantidad de datos y contenido que se almacene en la aplicación. Se recomienda disponer de al menos 50 MB de espacio libre para la instalación inicial. Además, ten en cuenta que la aplicación puede aumentar su tamaño a medida que se agregan más datos.
5. **Conexión a Internet:** Se requiere una conexión a Internet para descargar datos de las tiendas, sincronizar compras realizadas y realizar cálculos mensuales de gastos. Se recomienda una conexión estable para un rendimiento óptimo.
6. **Permisos:** La aplicación requerirá los siguientes permisos:
  - Acceso a Internet: para descargar datos y sincronizar información con las tiendas y servicios relacionados.

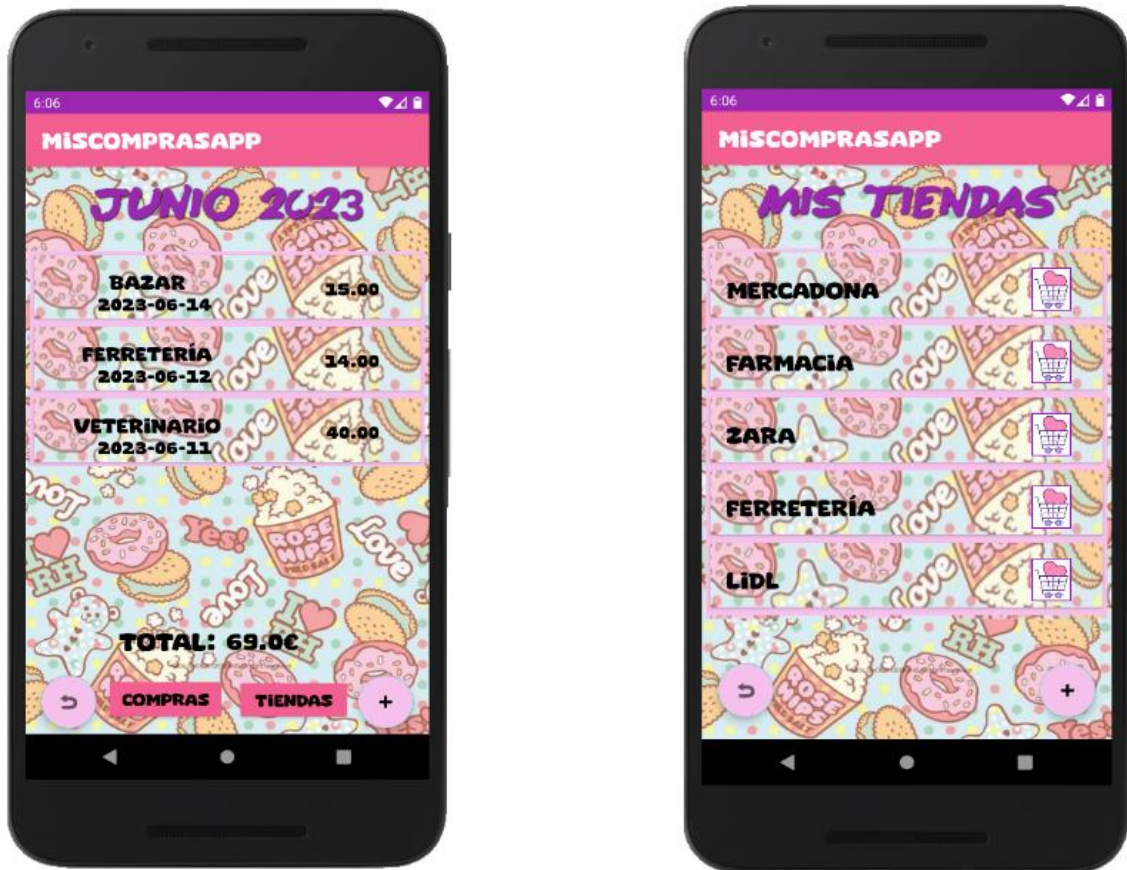
- **Acceso al almacenamiento:** para guardar datos locales y realizar copias de seguridad si es necesario.
7. **Preparación de la base de datos:** Asegúrate de que la base de datos necesaria para la aplicación de Tiendas y Compras esté correctamente configurada y lista para su uso. Esto incluye haber creado la base de datos en tu servidor de bases de datos y haber importado los datos necesarios.
8. **Detalles de la base de datos:** Proporciona a los usuarios los detalles necesarios para establecer una conexión con la base de datos. Estos detalles suelen incluir lo siguiente:
- **Host de la base de datos:** Indica la dirección IP o el nombre de host del servidor de la base de datos donde se encuentra alojada la base de datos.
  - **Nombre de usuario y contraseña:** Proporciona las credenciales necesarias para autenticarse en el servidor de la base de datos.
  - **Nombre de la base de datos:** Indica el nombre de la base de datos utilizada por la aplicación.

### **5.3. Manual de usuario**

Al iniciar la aplicación nos aparecerá el mes actual de las compras realizadas con el nombre de la tienda, la fecha en la que se hizo la compra más el importe gastado de cada compra además reflejará el gasto total mensual en la parte inferior de la pantalla.

Esta funcionalidad proporcionará una visión rápida de las compras realizadas durante el mes actual y te permitirá conocer el gasto acumulado hasta el momento.

Al pulsar sobre el **botón “Tiendas”** nos llevará a la pantalla del listado de tiendas que tenemos introducidas en nuestra aplicación.



Al pulsar el **icono "+"** se abrirá una ventana de diálogo.

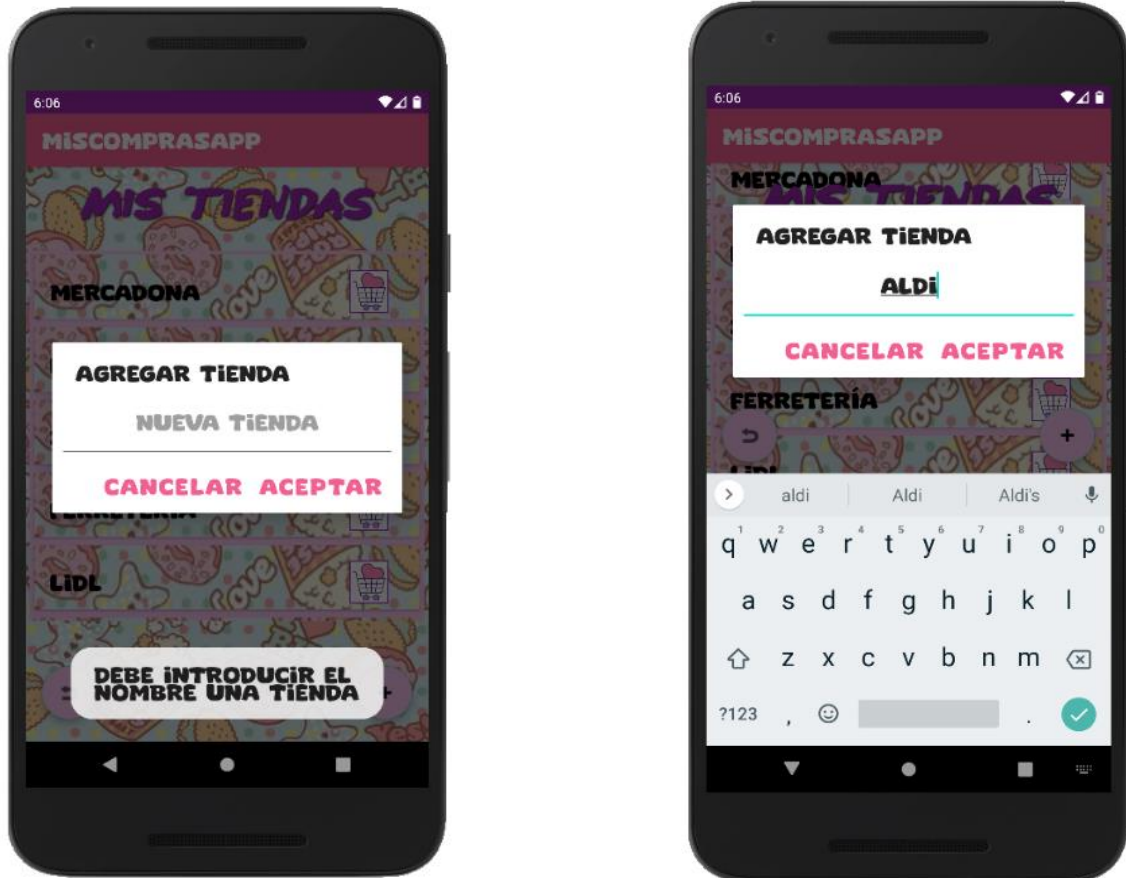
En esa ventana, se nos solicitará ingresar el nombre de la tienda.

Después de ingresar el nombre de la tienda, podremos hacer clic en el botón "Aceptar" para agregarla a nuestro listado de tiendas.

Si intentamos hacer clic en "Aceptar" sin haber ingresado ningún nombre de tienda, se mostrará un mensaje de aviso indicándonos que debemos completar el campo.

Una vez que hayamos ingresado el nombre de la tienda y hecho clic en "Aceptar", la tienda se agregará al listado y estará disponible para nuestra selección en futuras opciones o consultas.





Al hacer **clic en una tienda** existente, se abrirá una ventana de diálogo para poder realizar la modificación.

En esa ventana, el campo del nombre de la tienda se completará automáticamente con el nombre de la tienda seleccionada.

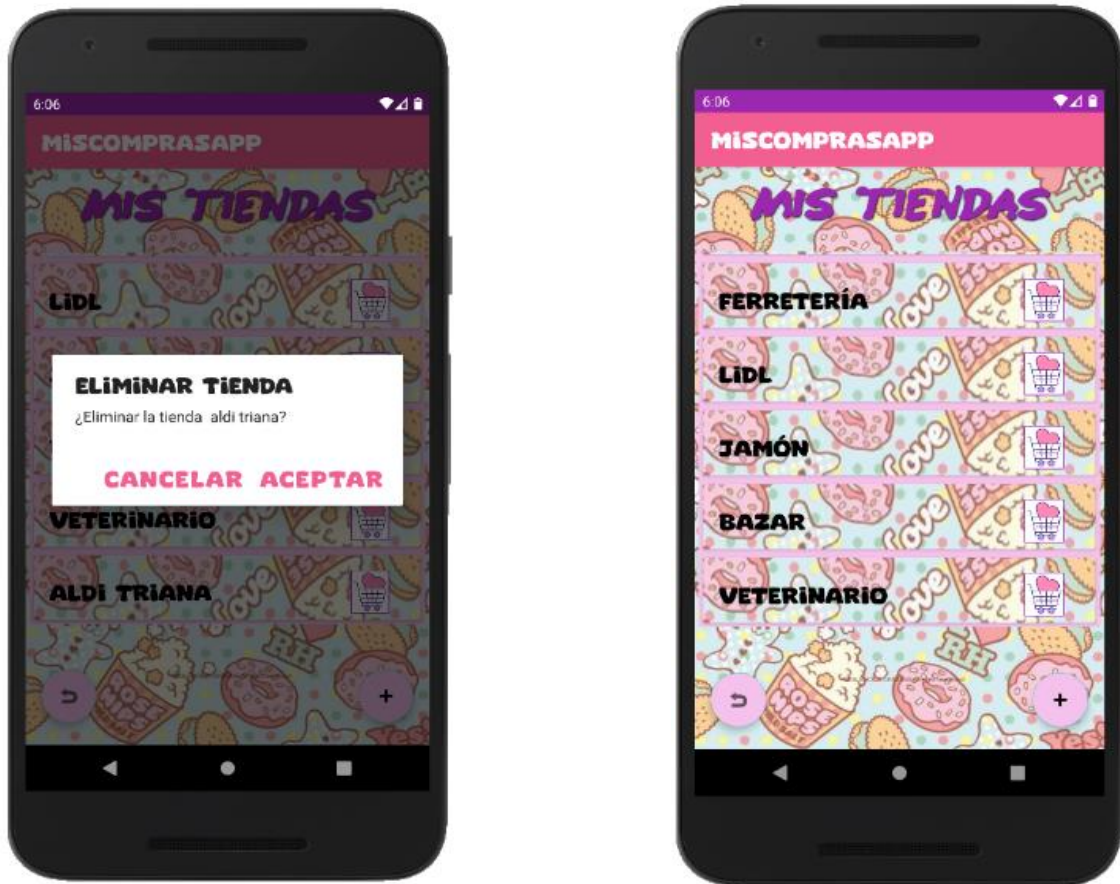
Podrás modificar el nombre de la tienda en el campo correspondiente de la ventana de diálogo.

Si intentamos hacer clic en "Aceptar" sin haber ingresado un nombre de tienda, se mostrará un mensaje de aviso indicándonos que debemos completar el campo.

Una vez que se hayan realizado los cambios y confirmado haciendo clic en "Aceptar", la tienda se actualizará en el listado y reflejará el nuevo nombre ingresado.



Para eliminar una tienda haremos **clic largo sobre la tienda** que queramos eliminar y nos saldrá una ventana de confirmación de borrado. Al hacer clic en "Aceptar" la tienda habrá desaparecido de nuestra lista de tiendas.



Al pulsar el **icono de la “flechita”** en la parte inferior izquierda de la pantalla, volveremos al listado de compras del mes actual.

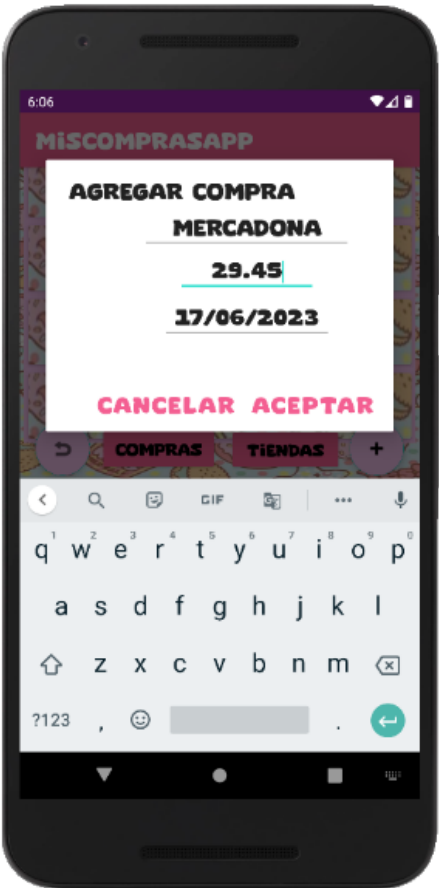
Desde la pantalla de Compras podremos realizar las mismas acciones que en el apartado de Tiendas, utilizando la misma funcionalidad.

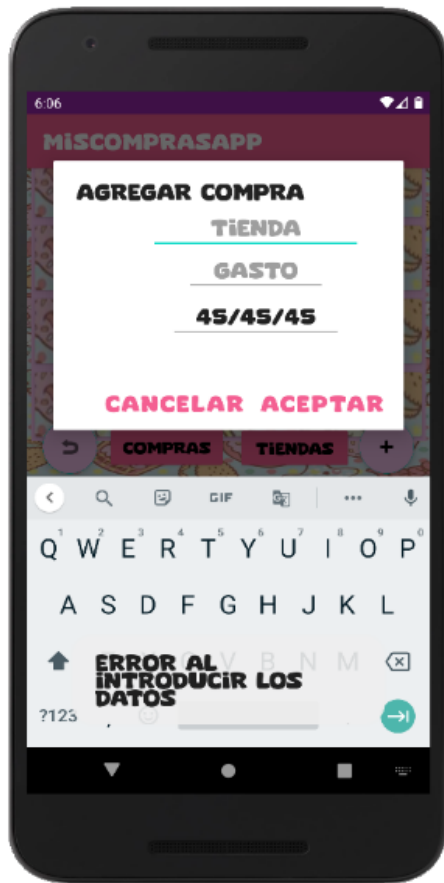
Al pulsar sobre el **icono “+”** podremos dar de alta una nueva Compra.

Aparecerá una ventana de diálogo para poder ingresar el nombre de la tienda, el gasto, y en la fecha nos aparecerá predeterminada la fecha actual, la cual podremos cambiar si lo deseamos. Le damos a “Aceptar” y se nos habrá agregado a nuestro listado de compras del mes y el año introducido.

Se nos mostrará un mensaje de aviso si dejamos los campos vacíos, al introducir valores erróneos tanto en el gasto como en el apartado fecha o si el nombre de la tienda introducida no existe.

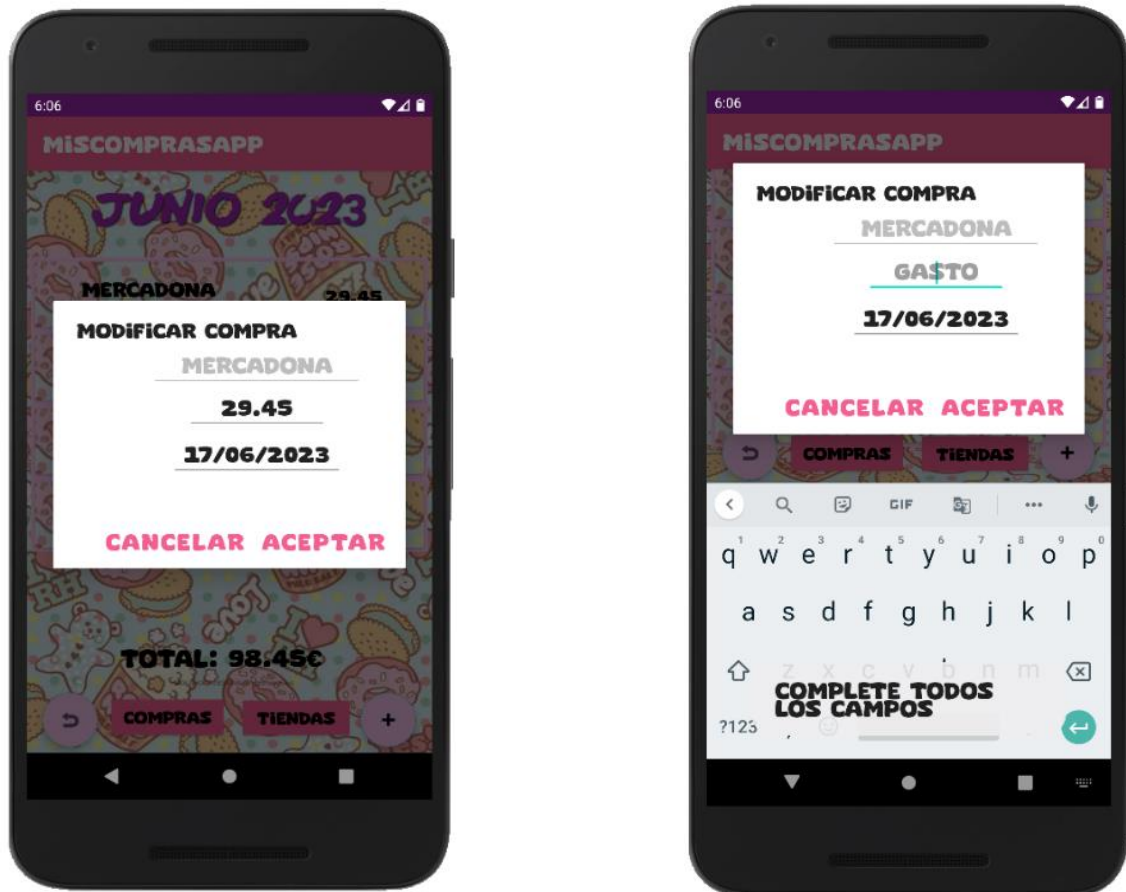






Para modificar una Compra daremos un **clic sobre la compra** que deseemos, apareciendo en una ventana de diálogo donde estarán ya introducidos los datos de la compra seleccionada completados de forma automática, no pudiendo modificar el nombre de la Tienda.

Si dejamos cualquier campo vacío o los valores introducidos son erróneos nos aparecerá el mensaje de aviso en cuestión.



Al realizar la modificación correctamente nos hará el cambio de datos en la compra y se nos actualizará el gasto total mensual si hemos modificado el importe de compra.

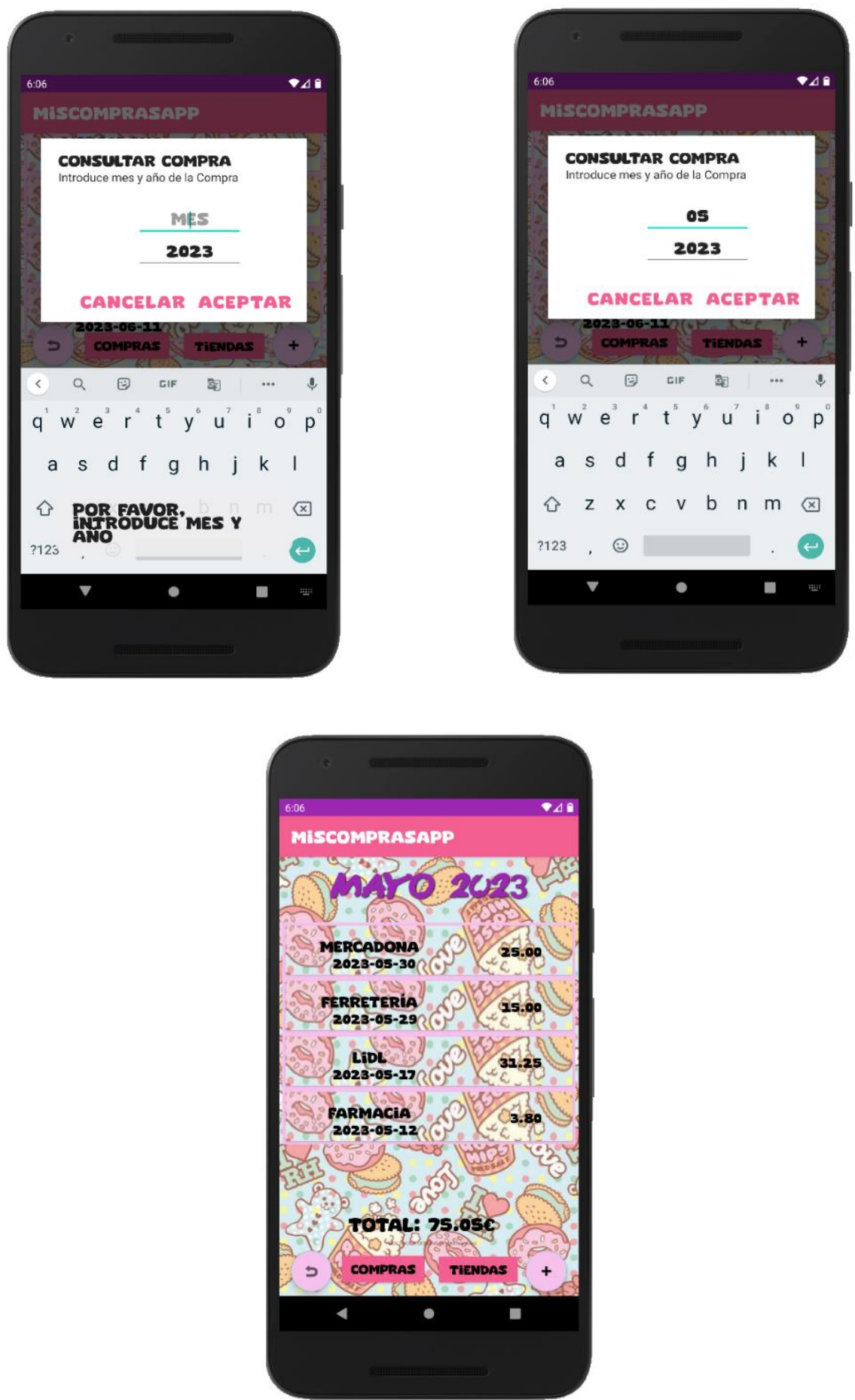
Para eliminar una compra haremos **clic largo sobre la compra** que queramos eliminar y nos saldrá una ventana de confirmación de borrado. Al hacer clic en "Aceptar" la compra habrá desaparecido de nuestra lista de compras, actualizándose el importe mensual.

Al pulsar ese **botón de "Compras"** se nos presentará una ventana donde podremos introducir el mes y el año de la compra que a consultar. Una vez que hayamos ingresado esos valores, se mostrará un listado de todas las compras realizadas en ese mes y año específicos, junto con la información correspondiente a cada compra.

Además de mostrar el listado de compras, también se proporcionará la suma total de todas las compras realizadas en el mes seleccionado. Esto significa que podremos ver el valor total gastado en ese período específico.

Esta funcionalidad nos permitirá acceder rápidamente a información detallada sobre nuestras compras, filtrada por mes y año, lo que puede ser útil para realizar un seguimiento de nuestros gastos mensuales.

Si dejamos campos vacíos o los valores introducidos no son válidos, aparecerá el mensaje de aviso correspondiente.





## **5.4. Manual de administración**

### **Consulta de Tiendas Registradas**

Para consultar las tiendas registradas en la aplicación, siga los siguientes pasos:

- Navegue hasta la sección " Tiendas".
- Verá un listado de todas las tiendas registradas.

### **Registro de Nuevas Tiendas**

Para registrar una nueva tienda en la aplicación, siga los siguientes pasos:

- Navegue hasta la sección " Tiendas".
- Haga clic en el símbolo "+" para agregar una nueva tienda.
- Se abrirá un formulario de registro de tienda.
- Complete los campos requeridos, como el nombre de la tienda.
- Haga clic en el botón "Aceptar" para registrar la nueva tienda en la base de datos.
- La tienda ahora aparecerá en el listado de tiendas registradas.

### **Edición de Tiendas Existentes**

Si necesita realizar cambios en los detalles de una tienda existente, siga los siguientes pasos:

- Navegue hasta la sección " Tiendas".
- Busque la tienda que desea editar en el listado de tiendas.
- Haga clic sobre la tienda que desee editar.
- Se abrirá un formulario de edición de tienda.
- Realice los cambios necesarios en el campo del nombre tienda.
- Haga clic en el botón "Aceptar" para aplicar los cambios realizados en la tienda.

### **Eliminación de Tiendas**

Si desea eliminar una tienda de la aplicación, siga los siguientes pasos:

- Navegue hasta la sección " Tiendas".
- Busque la tienda que desea eliminar en el listado de tiendas.
- Realice una pulsación larga sobre la tienda seleccionada.



- Se mostrará un mensaje de confirmación de eliminación.
- Haga clic en "Aceptar" para eliminar definitivamente la tienda de la base de datos.

### Consulta de Compras del mes actual

Para consultar las compras registradas en la aplicación, siga los siguientes pasos:

- Nada más abrir la aplicación verá la consulta de compras del mes actual.
- Puede ver información detallada de cada compra, como la tienda, el saldo gastado, la fecha y el gasto mensual.

### Registro de Nuevas Compras

Para registrar una nueva compra en la aplicación, siga los siguientes pasos:

- Nada más abrir la aplicación verá la consulta de compras del mes actual.
- Haga clic en el símbolo "+" para agregar una nueva compra.
- Se abrirá un formulario de registro de compra.
- Complete los campos requeridos, como el nombre de la tienda, el saldo gastado y la fecha de la compra.
- Haga clic en el botón "Aceptar" para registrar la nueva compra en la base de datos.

La compra ahora aparecerá en el listado de compras registradas.

### Edición de Compras Existentes

Si necesita realizar cambios en los detalles de una compra existente, siga los siguientes pasos:

- Nada más abrir la aplicación verá la consulta de compras del mes actual.
- Busque la compra que desea editar en el listado de compras del mes actual o accediendo a la compra desde el botón "Compras" añadiendo mes y año de la compra.
- Haga clic sobre la compra seleccionada.
- Se abrirá un formulario de edición de compra.
- Realice los cambios necesarios en los campos deseados, como la tienda, el saldo gastado y/o la fecha.
- Haga clic en el botón "Aceptar" para aplicar los cambios realizados en la compra.

### Eliminación de Compras

Si desea eliminar una compra de la aplicación, siga los siguientes pasos:

- Nada más abrir la aplicación verá la consulta de compras del mes actual.
- Busque la compra que desea eliminar en el listado de compras del mes actual o accediendo a la compra desde el botón “Compras” añadiendo mes y año de la compra.
- Busque la compra que desea eliminar en el listado de compras.
- Realice una pulsación larga sobre la compra seleccionada.
- Se mostrará un mensaje de confirmación de eliminación.
- Haga clic en "Aceptar" para eliminar definitivamente la compra de la base de datos.

### Consulta de Compras Registradas en una fecha concreta

Para consultar las compras registradas en la aplicación, siga los siguientes pasos:

- Navegue hasta la sección con el botón de "Compras".
- Se abrirá un formulario de consulta de compra.
- Puede ver información detallada de cada compra, como la tienda, el saldo gastado, la fecha y el gasto mensual.

## **6. Conclusiones finales**

### **6.1. Grado de consecución de los objetivos fijados**

Estoy satisfecha de los resultados obtenidos, el desarrollo de la aplicación y su implementación han cumplido con los objetivos fijados en el estudio preliminar.

La interfaz gráfica me parece amigable, divertida e intuitiva para el usuario y una aplicación muy útil en el día a día.

## 6.2. Posibles mejoras o ampliaciones a implementar en el futuro.

Esta claro que con la multitud de aplicaciones que existen en el mercado se pueden añadir diversas mejoras y ampliaciones para que la aplicación sea más completa y atractiva para el usuario.

He resaltado algunas funcionalidades extras que se podían incorporar como:

- Introducción de apuntes de compra: Poder añadir si se necesita, algún apunte sobre artículos comprados en esa compra que sea de relevancia como el precio de algún producto en concreto, ofertas destacadas, etc.
- Funcionalidad de búsqueda avanzada: Mejorar la función de búsqueda avanzada que permita a los usuarios buscar compras por diversos criterios, como nombre de tienda o categoría de producto si se introduce dicha opción de apunte de productos.
- Análisis de gastos: Agregar una función de análisis de gastos que proporcione a los usuarios una visión general de sus patrones de gasto, incluyendo gráficos y estadísticas. Esto les permitiría tener un mejor control de sus finanzas personales.
- Gestión de listas de deseos: Permitir a los usuarios crear y gestionar listas de deseos, donde puedan agregar productos o artículos que deseen comprar en el futuro. Esto facilitaría la planificación y organización de las compras.
- Personalización de la aplicación con iconos, temas de fondo de pantalla.

## 7. Bibliografía

- [Estudio de viabilidad de un proyecto ¿Cómo realizarlo? \(infinitiaresearch.com\)](https://infinitiaresearch.com/)
- [EJEMPLO DE ESTUDIO DE VIABILIDAD - ppt descargar \(slideplayer.es\)](https://slideplayer.es/)
- [Comunidad Emagister 66463 sistemas.pdf](#)
- [TFG-G3703.pdf \(uva.es\)](https://uva.es/)
- [Capítulo 6 Normalizacion | Base de Datos \(bookdown.org\)](https://bookdown.org/)
- [Los gestores de bases de datos \(SGBD\) más usados \(inesem.es\)](https://inesem.es/)
- [¿Qué es un sistema de gestión de bases de datos \(SGBD\)? \(hostinger.es\)](https://hostinger.es/)
- [Diagramas de Flujos de Datos \(campustudium.com\)](https://campustudium.com/)
- [Pruebas.pdf \(campustudium.com\)](https://campustudium.com/)
- [Desarrolladores de Android | Android Developers](https://androiddevelopers.com/)
- Temario Grupo Studium 1ºDAM/DAW: Base de Datos
- Temario Grupo Studium 1ºDAM/DAW: Entorno de desarrollo
- Temario Grupo Studium 2ºDAM: PMDM
- Trabajos y prácticas realizados durante el grado que me han servido a modo de consulta y orientación.