

# **Desarrollo de una herramienta inteligente para gestionar las compras familiares**

## **Development of a smart tool to manage family purchases**



### **Trabajo de fin de grado**

Curso 2021-2022

#### **Autores**

Pablo Fernández de la Moya

Sergio Morán Agüero

Jesús Cencerrado Pedrero

#### **Director**

Antonio Sarasa Cabezuelo

#### **Tutor**

Antonio Sarasa Cabezuelo

**Trabajo de Fin de Grado en Ingeniería Informática**

**Facultad de Informática**

**Universidad Complutense de Madrid**



Desarrollo de una herramienta inteligente para  
gestionar las compras familiares  
Development of a smart tool to manage family  
purchases

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA Y DE SOFTWARE

**Autores**

Pablo Fernández de la Moya  
Sergio Morán Agüero  
Jesús Cencerrado Pedrero

**Director**

Antonio Sarasa Cabezuelo

**Tutor**

Antonio Sarasa Cabezuelo

**Convocatoria:** Febrero 2022

**Trabajo de Fin de Grado en Ingeniería Informática y software**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**

**1 de Febrero de 2022**

# Autorización de difusión

Los abajo firmantes, alumno y tutor del Trabajo Fin de Grado (TFG) en Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, el Trabajo Fin de Grado (TFG) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

Título del TFG: **Desarrollo de una herramienta inteligente para gestionar las compras familiares.**

Curso académico: 2021–2022

Nombre de los Alumnos: **Pablo Fernández de la Moya, Sergio Morán Agüero y Jesús Cencerrado Pedrero.**

Director del TFG: **Antonio Sarasa Cabezuelo**, (Sistemas Informáticos y Computación).

Tutor del TFG: **Antonio Sarasa Cabezuelo**, (Sistemas Informáticos y Computación).

*Experiencia es el nombre que damos  
a nuestros errores.  
-Oscar Wilde.*

# Agradecimientos

## **0.1 Pablo Fernández de la Moya**

Quiero agradecer a mi familia todo el apoyo que me han dado durante estos años de carrera, ya que en los momentos más difíciles siempre han estado ahí.

También agradecer a mi novia Paula, que ha sido un apoyo incondicional durante estos años y me ha ayudado a seguir adelante y no rendirme.

Tampoco quiero olvidarme de mis amigos y compañeros, con los que he compartido vivencias, ya que sin ellos a saber donde estaría, seguramente seguiría en MDL de primer curso, sobre todo a Alberto, Sergio y Jesús, estos dos últimos compañeros de viaje en este TFG.

Por último agradecer a los profesores y el personal de la universidad, la cual me alegro de haber elegido en su momento.

## **0.2 Sergio Morán Agüero**

Lo primero de todo quiero agradecer a mis compañeros del TFG por el trabajo realizado estos últimos meses.

También me gustaría hacer una mención especial a mi familia, por estar ahí en los momentos complicados que he sufrido durante la carrera. Han sido una fuente de apoyo muy importante para mí.

Por último, quisiera agradecer a la universidad y a todos sus profesores por la enseñanza que nos han dado estos años que, sin duda, atesoraré el resto de mi vida.

## **0.3 Jesús Cencerrado Pedrero**

Quiero dar las gracias a mi familia, que ha sido el centro de apoyo emocional y mental que tan bien me ha hecho durante toda mi vida y más ahora, y por hacerme ver que puedo con todo y más.

Por mis amigos, tanto los nuevos como los de siempre, por darme consejos, apoyo e indicarme el camino a seguir.

Por toda la gente con la que he coincidido en la carrera y que se ha cruzado en mi camino.

Gracias a todas las personas que me han acompañado en este viaje, esto es por vosotros.

## **0.4 Todos**

Queremos agradecer a nuestro tutor del TFG, Antonio, por habernos guiado en este camino corto pero intenso, por habernos dado consejos siempre que se lo hemos pedido, por el buen trato recibido y por los ánimos que nos ha brindado.

También queríamos dar las gracias a nuestros amigos y familiares, por ser un apoyo indispensable en este largo camino.

Y a todos los profesores con los que hemos compartido tantas clases y aprendizaje, por enseñarnos de la mejor manera posible.

Gracias.

# Resumen

El presente trabajo consiste en una aplicación Android para la gestión de las listas de la compra en los hogares. Esta aplicación se ha diseñado de modo que las funcionalidades implementadas resulten sencillas, cómodas e intuitivas de usar para el usuario.

Para fomentar la interacción de los usuarios se ha desarrollado un sistema que permite compartir listas entre usuarios de la aplicación de modo que puedan acceder a listas de otros usuarios.

Además, los usuarios tendrán un historial de listas para poder ver las últimas compras realizadas y obtener estadísticas detalladas de ellas.

# Palabras clave

- Lista/s
- Producto/s
- Compra
- Usuario
- Historial/Histórico
- Estadísticas

# Abstract

This work consists of an Android application for the management of household shopping lists. This application has been designed in such a way that the implemented functionalities are simple, comfortable and intuitive to use for the user.

To encourage user interaction, a system has been developed that allows sharing lists between users of the application so that they can access other users' lists.

In addition, users will have a list history so that they can see the last purchases made and obtain detailed statistics on them.

# Keywords

- List/s
- Product/s
- Buy
- User
- History/Historical
- Statistics

# Índice

0.1 Pablo Fernández de la Moya	6
0.2 Sergio Morán Agüero	6
0.3 Jesús Cencerrado Pedrero	6
0.4 Todos	7
<b>Lista de figuras</b>	<b>15</b>
<b>Introducción</b>	<b>17</b>
1.1 Introducción	17
1.2 Motivación	17
1.3 Objetivos	18
1.4 Plan de trabajo	18
<b>Introduction</b>	<b>19</b>
1.1 Introduction	19
1.2 Motivation	19
1.3 Goals	20
1.4 Work Plan	20
<b>Estado del arte</b>	<b>21</b>
<b>Especificación de requisitos</b>	<b>22</b>
3.1 Actores	22
3.2 Módulos	22
3.2.1. Módulo usuario	22
3.2.2 Módulo listas	23
3.2.2 Módulo histórico	23
3.3 Diagramas de flujo	24
3.3.1 Crear una lista y añadir productos.	24
3.3.2 Comprar una lista.	26
3.3.3 Consultar una lista en el histórico de listas.	27
3.3.4 Invitar a un usuario a la lista.	28
3.4 Análisis de requisitos	29
3.4.1 Requisitos funcionales	29
3.4.2 Requisitos no funcionales	30
<b>Tecnologías</b>	<b>31</b>
4.1 Android Studio	31
4.2 Firebase	31
4.3 Google Drive	31
Arquitectura y modelo de datos	32
5.1 Arquitectura	32
5.1.1 MVVM	32

5.2 Modelo de datos	33
5.2.1 Modelo Jerárquico	33
5.2.1 Información en la base de datos	33
<b>Implementación</b>	<b>38</b>
6.1 Fase de diseño	38
6.1.1 Elementos Android	38
6.1.1.1 Activities	38
6.1.1.2 ListView	38
6.1.1.3 LinearLayout	38
6.1.1.4 Bottom Navigation Bar	39
6.1.1.5 Menu	39
6.1.1.6 Toolbar	39
6.1.2 Base de datos	40
6.2 Clases principales	40
6.2.1.2 Login/Logout	40
6.2.1.4 Register	41
6.2.2.2 CreateList/DeleteList/ViewList/EditNameList	41
6.2.2.3 CreateProduct/DeleteProduct/ViewProduct	41
6.2.2.5 Buy/Summary/LeaveList	41
6.2.3.2 HistoricLists	42
<b>Evaluación</b>	<b>43</b>
<b>Conclusiones y trabajo futuro</b>	<b>51</b>
8.1 Conclusiones	51
8.2 Trabajo futuro	51
<b>Conclusions and future work</b>	<b>53</b>
8.1 Conclusions	53
8.2 Future Work	53
<b>Aportaciones individuales</b>	<b>55</b>
9.1 Jesús Cencerrado Pedrero	55
9.2 Sergio Morán Agüero	55
9.3 Pablo Fernández de la Moya	56
<b>Apéndices</b>	<b>58</b>
Manual de usuario	58
Bibliografía	89

# Lista de figuras

Figura 3.2 Módulo listas	25
Figura 3.3 Módulo histórico	26
Figura 3.4 Diagrama de crear lista y añadir productos	27
Figura 3.6 Diagrama de consultar historial de listas	29
Figura 3.7 Diagrama de invitar usuario a una lista	34
Figura 5.1 Patrón MVVM estructura	35
Figura 5.2 Diagrama jerárquico base de datos	35
Figura 5.3 Diagrama módulo listas	41
Figura 6.2: Bottom Navigation Bar	41
Figura 6.3: Menú	41
Figura 6.4: ToolBar	45
Figura 7.2: Pregunta 2	46
Figura 7.3: Pregunta 3	47
Figura 7.4: Pregunta 4	47
Figura 7.5: Pregunta 5	48
Figura 7.7: Pregunta 7	49
Figura 7.9: Pregunta 9	50
Figura 7.10: Pregunta 10	50
Figura 7.11: Pregunta 11	51
Figura 7.12: Pregunta 12	51
Figura 7.13: Pregunta 13	60
Figura 10.1: Splash - Primera pantalla	60
Figura 10.2: Inicio de la aplicación	61
Figura 10.3: Registro	62
Figura 10.4: Página principal	63
Figura 10.5: Añadir nombre a la lista	65
Figura 10.6: Productos de una lista vacía	66
Figura 10.7: Productos de una lista	67
Figura 10.8: Formulario de un producto	68
Figura 10.9: Dentro de un producto	69
Figura 10.10: Menú de lista con productos	71
Figura 10.11: Menú de lista sin productos	72
Figura 10.12: Usuarios de una lista	73
Figura 10.13: Invitar usuarios	74
Figura 10.14: Comprar lista	75
Figura 10.15: Popup comprar lista	76
Figura 10.16: Comprar lista con productos marcados	77

Figura 10.17: Generar ticket	78
Figura 10.18: Perfil del usuario	79
Figura 10.19: Editar perfil del usuario	80
Figura 10.20: Borrar cuenta	81
Figura 10.21: Editar perfil del usuario	82
Figura 10.22: Historial de listas	83
Figura 10.23: Dentro de una lista del historial	84
Figura 10.24: Estadísticas	85
Figura 10.25: Notificaciones de invitación a una lista	86
Figura 10.26: Listas a las que te han invitado	87
Figura 10.27: No tienes invitaciones a otras listas	88
Figura 10.28: PopUp para aceptar o rechazar invitaciones a otras listas	89

# Introducción

## 1.1 Introducción

Desde la irrupción de las nuevas tecnologías en el mundo, la gran mayoría de las tareas que se realizaban de forma manual han sido reinventadas y adaptadas a estas nuevas tecnologías para hacerlas más sencillas, eficientes, económicas...

Un ejemplo de esto es ir a hacer la compra. Actualmente, la gestión de las listas de la compra en los hogares se realiza de forma manual. Se escriben los productos en un papel que, más adelante, se llevará al supermercado para cerciorarse de que no se olvida nada.

Sin embargo, este método, además de ser poco práctico, consume una gran cantidad de recursos naturales al utilizar papel.

Por ello, se ha desarrollado una aplicación Android para la gestión de las compras familiares que implementa las funcionalidades necesarias para la gestión de las listas de la compra, de modo que agilice la realización de esta tarea y permita eliminar el malgasto de papel que conlleva.

## 1.2 Motivación

Hoy en día las tecnologías y, sobre todo, los móviles son usados por la gran mayoría de la población. Por ello, a medida que pasan los años, cada vez irrumpen más aplicaciones que cubren una necesidad que antes se realizaba de una forma manual. A pesar de que el sector más mayor de la población siga realizando ciertas tareas como antiguamente, este sector se va encogiendo con el paso del tiempo y está parte de la población va aceptando estas nuevas tecnologías como parte de sus vidas.

Además, otro motivo a destacar es el alto riesgo de escasez de los recursos naturales que los expertos indican para el 2050, por lo que cualquier ayuda para disminuir el consumo de estos recursos siempre es bienvenida. Con esta situación en mente, se contempló como buena iniciativa reducir el consumo de papel en los hogares.

Teniendo en cuenta estos problemas, se ha decidido crear una aplicación para la gestión de las listas de la compra en los hogares, que pueda realizar todas las funcionalidades pertinentes para cubrir las necesidades del usuario y facilitar la realización de esta tarea.

### **1.3 Objetivos**

El objetivo es desarrollar una aplicación capaz de gestionar las listas de la compra de los usuarios e incluir nuevas funcionalidades que un usuario tendría si realizase esta tarea en papel, como el historial de sus listas o las estadísticas de sus compras.

También se pretende crear una aplicación que pueda ser utilizada por cualquier persona. Para ello, al desarrollarla, se debe tener en cuenta a personas de cualquier rango de edad o personas que no estén familiarizadas con las nuevas tecnologías.

De esta manera, se puede realizar de una forma sencilla la misma tarea que antes se realizaba de forma manual.

### **1.4 Plan de trabajo**

Para desarrollar la aplicación, se tuvo que realizar una investigación sobre las tecnologías a utilizar:

- Estructura que se utilizaría para desarrollar la propia aplicación y la base de datos.
- Librerías Android para la conexión a la base de datos.
- Utilización de distintos diseños para las distintas funcionalidades de las listas.
- Elementos del SDK de Android.

Ya que la aplicación estará únicamente disponible en Android, se deberá realizar una investigación previa sobre la programación en Android. Estará centrada principalmente en Android Studio[5], por lo que se deberá conocer su funcionamiento y realizar el diseño e implementación de la aplicación.

Tras definir los objetivos, se empezó con la creación de una Base de Datos para el almacenamiento y extracción de los datos de la aplicación.

Por último, se deberá implementar la interfaz de la aplicación y las funcionalidades incluidas en esta.

# Introduction

## 1.1 Introduction

Since the irruption of new technologies in the world, most of the tasks that used to be done manually have been reinvented and adapted to these new technologies to make them simpler, more efficient, more economic...

An example of this is shopping. Currently, the management of shopping lists in households is done manually. Products are written down on a piece of paper, which is then taken to the supermarket to make sure that nothing is forgotten.

However, this method, in addition to being impractical, consumes a large amount of natural resources by using paper.

For this reason, an Android application has been developed for family shopping management that implements the necessary functionalities for the management of shopping lists, in order to speed up this task and eliminate the waste of paper involved.

## 1.2 Motivation

Nowadays technologies and, above all, cell phones are used by the vast majority of the population. Therefore, as the years go by, more and more applications are appearing that cover a need that used to be performed manually. Although the older sector of the population continues to perform certain tasks as in the past, this sector is shrinking over time and this part of the population is accepting these new technologies as part of their lives.

In addition, another reason to highlight is the high risk of scarcity of natural resources that experts indicate for 2050, so any help to reduce the consumption of these resources is always welcome. With this situation in mind, reducing paper consumption in households was seen as a good initiative.

Taking into account these problems, it was decided to create an application for the management of household shopping lists, which can perform all the relevant functionalities to meet the user's needs and facilitate the completion of this task.

### **1.3 Goals**

The objective is to develop an application capable of managing users' shopping lists and include new functionalities that a user would have if they were to perform this task on paper, such as the history of their lists or the statistics of their purchases.

It is also intended to create an application that can be used by anyone. To this end, when developing it, people of any age range or people who are not familiar with new technologies should be taken into account.

In this way, the same task that used to be done manually can be performed in a simple way.

### **1.4 Work Plan**

In order to develop the application, research had to be carried out on the technologies to be used:

- Structure that would be used to develop the application itself and the database.
- Android libraries for the connection to the database.
- Use of different layouts for the different functionalities of the lists.
- Elements of the Android SDK.

Since the application will be available only on Android, a previous research on Android programming should be done. It will be mainly focused on Android Studio[5], so it will be necessary to know how it works and to design and implement the application.

After defining the objectives, we started with the creation of a database for the storage and extraction of the application data.

Finally, the application interface and the functionalities included in it should be implemented.

## Estado del arte

Como es lógico, existen aplicaciones similares a la que se ha desarrollado en el mercado Android. A continuación, se exponen varios ejemplos de aplicaciones que hacen tareas similares a la nuestra:

- **Bring! Lista de compras [8]**
  - Una aplicación totalmente centrada en crear y organizar listas de la compra, con una interfaz en la que destaca los elementos que puedes añadir a la lista por encima de todo, incluyendo dibujos e iconos bastante grandes. En cada producto tienes su nombre, pero también se puede hacer anotaciones en caso de querer indicar marcas u otros asuntos. También permite crear más de una lista de la compra, añadir a las listas a otros usuarios y mantener entre todos la lista actualizada.
- **Google Keep [9]**
  - Esta aplicación es una clásica de Google, se caracteriza por ser fundamentalmente simple, con pocas funcionalidades y poco elaborada. Es una aplicación orientada a tomar notas y hacer listas colaborativas.
- **Listonic [10]**
  - Esta aplicación permite crear listas de productos, que se van guardando en una base de datos, y sugiere productos según los registros anteriores del usuario. Las listas se pueden compartir, tienen control de gastos y se puede meter productos con la voz. También se puede ir segregando los productos de una lista por tipos para una mejor organización y es compatible con Android Wear.
- **Microsoft OneNote [11]**
  - Esta aplicación permite tomar notas con una app que se sincronizaba con una versión para escritorio, y también escanear listas en papel con Office Lens. Además, integra las “sticky notes” de Windows 10. Las Sticky Notes son post-its que se pueden poner en el escritorio de Windows 10 y tomar todo tipo de notas, incluyendo, por lo tanto, listas de la compra.
- **Out of Milk [12]**
  - Esta aplicación lleva ya más de cinco años siendo una de las más importantes a la hora de gestionar listas de la compra. Mantiene esta puesto en las más importantes con una interfaz moderna y muy sencilla de utilizar, pudiendo crear una lista dividida en varias categorías, y en la que se puede marcar productos como comprados como si fuera Google Keep. También te permite crear varias listas y compartirlas con otras personas entre muchas funciones.

## Especificación de requisitos

En este capítulo se explicarán los distintos requisitos con los que cuenta la aplicación.

### 3.1 Actores

Debido al tipo de funcionalidades de la aplicación, solo se cuenta con un tipo de actor:

- Usuarios registrados: Son aquellos que disponen de una cuenta en el sistema. Tienen acceso a todas las funcionalidades de la aplicación

### 3.2 Módulos

Los módulos de la aplicación se pueden separar en:

- Módulo del usuario
- Módulo de listas
- Módulo histórico

#### 3.2.1. Módulo usuario

En este módulo se tratan las distintas funcionalidades que tienen los usuarios con respecto a la gestión de la información de su perfil.

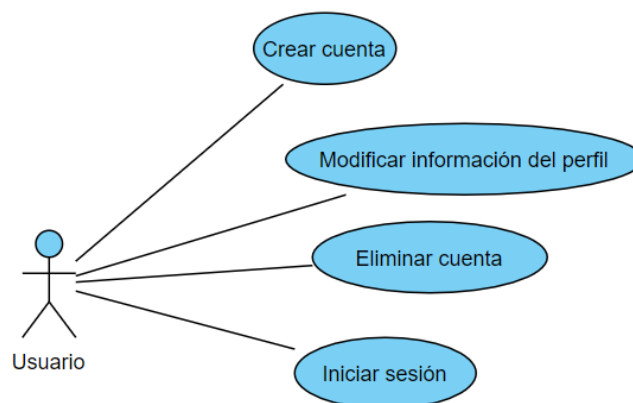


Figura 3.1 Módulo usuario

### 3.2.2 Módulo listas

Este módulo contiene las funcionalidades que los usuarios pueden hacer con las distintas listas de la compra.

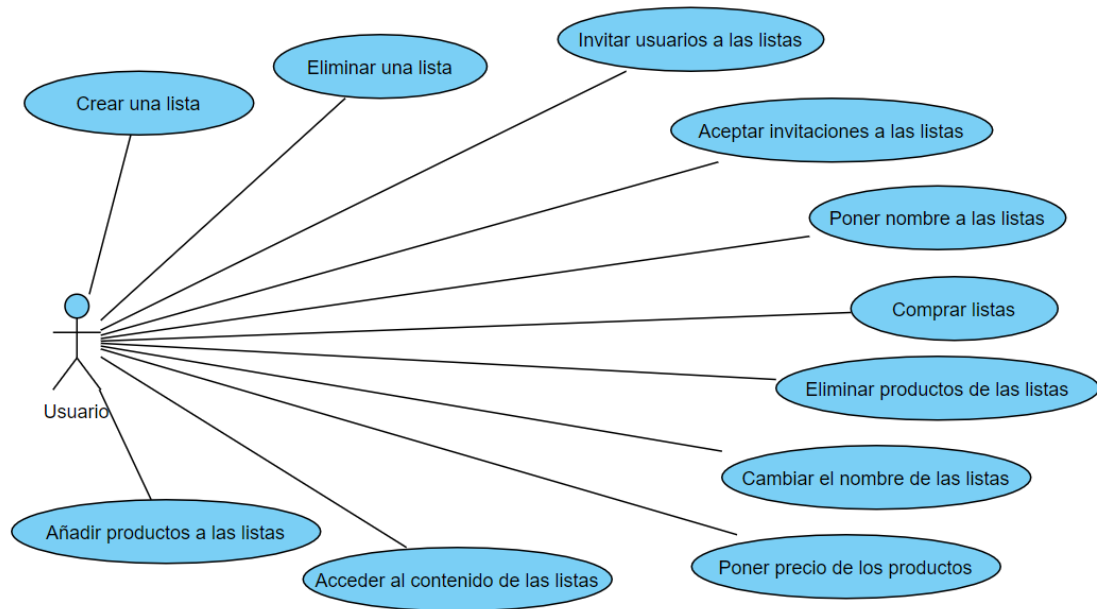


Figura 3.2 Módulo listas

### 3.2.2 Módulo histórico

Este módulo contiene las funcionalidades que los usuarios pueden hacer con las distintas listas de la compra en el histórico.

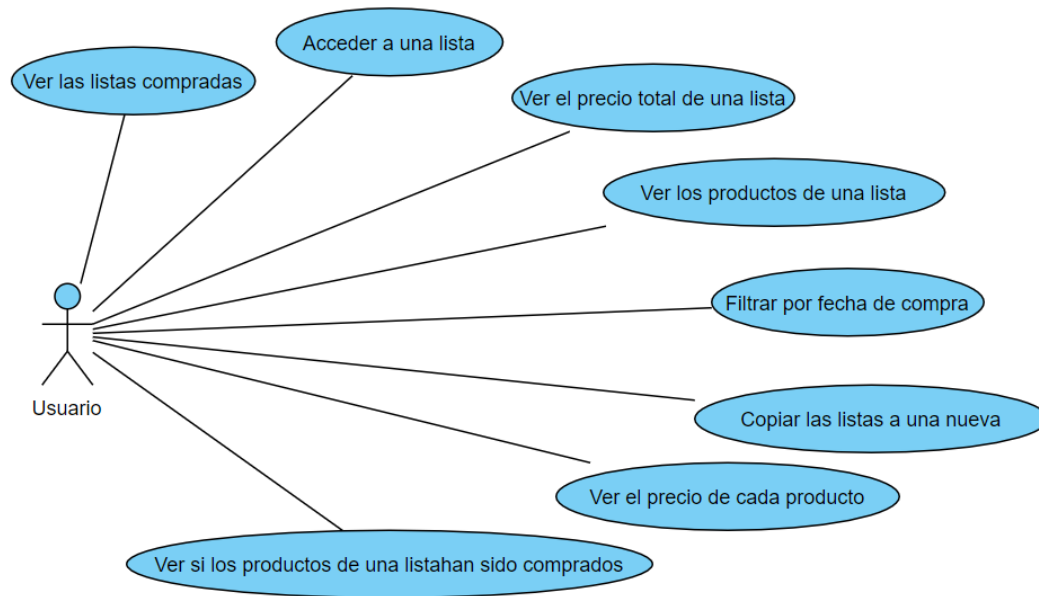


Figura 3.3 Módulo histórico

### 3.3 Diagramas de flujo

El siguiente apartado consta de los distintos flujos que un usuario tiene que hacer para completar y aprovechar algunas funcionalidades de la aplicación. Se da por hecho en los siguientes diagramas que el usuario está registrado en el sistema y tiene la sesión iniciada en la aplicación.

#### 3.3.1 Crear una lista y añadir productos.

Al entrar a la aplicación se ve el botón de crear la lista en la pantalla principal. Tras pulsar el botón, el usuario introduce el nombre de la lista y pulsa el botón de aceptar.

Con esto se crea una lista vacía. Dentro de la lista aparece el botón para añadir productos. Al pulsar en él, el usuario deberá rellenar un formulario para añadir un nuevo producto. Tras rellenar todos los campos, el usuario pulsa el botón de aceptar y con ello crea un nuevo producto dentro de la lista y se le redirige a la pantalla principal de la lista.

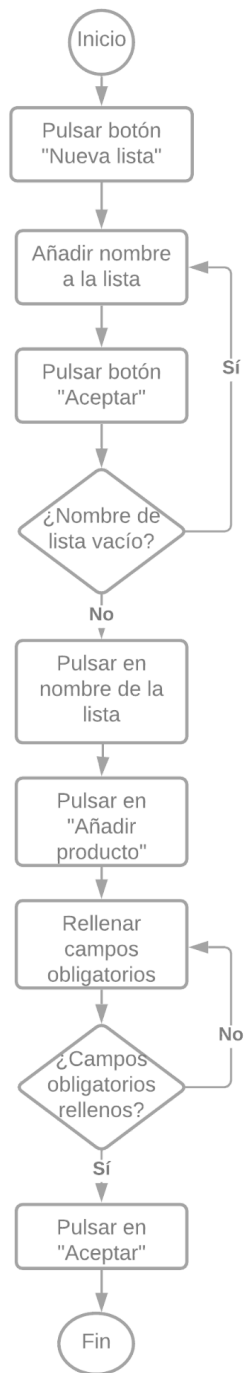


Figura 3.4 Diagrama de crear lista y añadir productos

### 3.3.2 Comprar una lista.

En la pantalla principal, el usuario selecciona la lista que quiere de entre sus listas. Una vez dentro de la lista, el usuario selecciona la opción de comprar lista de entre las opciones del menú de la parte superior. Al pulsarlo, el usuario accede a una pantalla donde se muestran los productos añadidos a la lista. Al ir pulsando en cada producto, el usuario introduce el precio y este se marca como comprado. Cuando el usuario haya comprado todos los productos de la lista, o desee finalizar la compra, pulsa el botón de finalizar compra. Al pulsar, la lista se pasa al histórico de listas y el usuario vuelve a la pantalla principal.



Figura 3.5 Diagrama de comprar una lista

### 3.3.3 Consultar una lista en el histórico de listas.

En la pantalla principal, el usuario utilizará el menú de navegación de la parte inferior para acceder a su perfil, pulsando el botón de la derecha. Al llegar al perfil, verá un resumen de su información de la cuenta y el botón de histórico de listas. Al pulsar llegará a otra pantalla donde se le mostrarán todas las listas que ha comprado con un filtro para poder seleccionar las del periodo de tiempo que desee. Al pulsar en cualquiera de las listas, accederá a otra pantalla donde se le mostrará un resumen de esa compra en formato “ticket” y donde podrá copiar esa lista a sus listas actuales en caso de así requerirse.

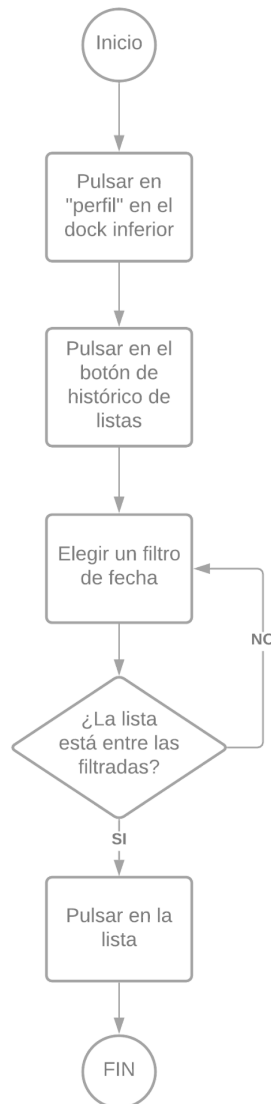


Figura 3.6 Diagrama de consultar historial de listas

### 3.3.4 Invitar a un usuario a la lista.

Desde la pantalla principal, el usuario accede a la lista en la que desea entrar. Una vez dentro seleccionará la opción de usuarios de entre las distintas opciones del menú de la parte superior. Ello le llevará a una nueva pantalla donde se mostrarán los usuarios pertenecientes a esa lista hasta el momento y donde contará con un botón para invitar a nuevos usuarios a esta. Al pulsarlo, se le llevará a otra pantalla donde deberá introducir el email del usuario al que quiere invitar y pulsar aceptar. De este modo el usuario queda invitado, y podrá aceptar o declinar la invitación en el apartado de notificaciones, situado en la parte superior en el icono de la campana.

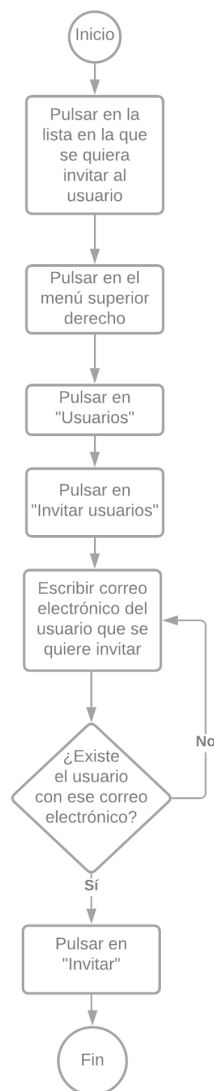


Figura 3.7 Diagrama de invitar usuario a una lista

## **3.4 Análisis de requisitos**

En toda aplicación existen requisitos que debe tener para el buen funcionamiento y la máxima rapidez posible. A continuación se explicarán:

### **3.4.1 Requisitos funcionales**

Son aquellos funcionales de la aplicación:

- Un nuevo usuario puede crear una cuenta a través de un formulario.
- Un usuario existente puede iniciar sesión con su email y contraseña.
- Un usuario registrado puede cerrar sesión.
- Un usuario registrado puede crear una lista.
- Un usuario registrado puede modificar el nombre de la lista.
- Un usuario registrado puede añadir productos a una lista.
- Un usuario registrado puede editar productos de una lista.
- Un usuario registrado puede borrar productos de la lista.
- Un usuario registrado puede borrar la lista.
- Un usuario registrado puede obtener el ticket provisional de la lista.
- Un usuario registrado puede invitar a otros usuarios a la lista.
- Un usuario registrado puede abandonar una lista.
- Un usuario registrado puede consultar las notificaciones que tiene.
- Un usuario registrado puede aceptar o rechazar invitaciones a una lista.
- Un usuario registrado puede ver los usuarios de una lista.
- Un usuario registrado puede eliminar usuario de la lista.
- Un usuario puede marcar productos como comprados y añadirle el precio.
- Un usuario registrado puede comprar una lista de productos.
- Un usuario registrado puede consultar el historial de sus compras.
- Un usuario registrado puede copiar listas del histórico a sus listas.
- Un usuario registrado puede obtener estadísticas de sus compras.
- Un usuario registrado puede editar la información de su perfil.
- Un usuario registrado puede eliminar su cuenta.

### **3.4.2 Requisitos no funcionales**

Son aquellos no funcionales de la aplicación:

- Aplicación desarrollada para sistemas Android.
- Lenguaje de programación Kotlin.
- Entorno de desarrollo Android Studio.
- Base de datos en tiempo real de Google Firebase.
- Firebase Auth [6] para la autenticación de los usuarios.
- Lenguaje web XML [20] para la interfaz.
- Patrón de diseño MVC.

# Tecnologías

En este capítulo daremos a conocer las tecnologías usadas para el desarrollo de la aplicación.

## 4.1 Android Studio

Este programa es uno de los más usados para el desarrollo de aplicaciones en la plataforma Android. Permite desarrollar aplicaciones en lenguaje Java o Kotlin[14] (basado en Java) además de tener otras muchas funciones de personalización de las distintas partes de la aplicación mediante interfaz gráfica.

Este programa también contiene emuladores que permiten probar el código desarrollado en el ordenador como si de un móvil se tratara, opción muy útil si no se dispone de un smartphone Android físicamente hablando.

Como se puede apreciar, es una aplicación muy completa tanto en el desarrollo de código como en opciones de personalización.

## 4.2 Firebase

Firebase[19] es una plataforma desarrollada por Google que facilita el desarrollo de apps, proporcionando un servidor backend para las aplicaciones. Entre sus funcionalidades se encuentra un servicio de autenticación, base de datos en tiempo real, almacenamiento de archivos, solución de errores, funciones backend, testeo, y medida de estadísticas recogidas de los usuarios.

En este caso se ha utilizado:

- Base de datos en tiempo real (Realtime): Firebase proporciona una base de datos noSQL que almacena datos y los sincroniza en tiempo real.
- Autenticación: Firebase proporciona un método de registro e inicio de sesión que no solo incluye autenticación a través de correo, sino que también permite la autenticación a través de proveedores externos como Facebook, Twitter, Github y Google. Aunque estas funcionalidades no han sido utilizadas en este proyecto.

### **4.3 Google Drive [25]**

Es una de las aplicaciones web más conocidas de Google. Trata de una herramienta de almacenamiento en la nube gratuita que se ha utilizado para tener un control de versiones de la aplicación.

# Arquitectura y modelo de datos

En este capítulo se presenta la arquitectura y el modelo de datos de la aplicación.

## 5.1 Arquitectura

En esta sección se presentan los patrones de diseño utilizados para reducir los errores en cuanto al desarrollo del código.

### 5.1.1 MVVM

Para el desarrollo del proyecto, se aplica el patrón MVVM [18], promovido por Android [4]. Para ello, el proyecto se estructura en tres partes, el modelo, la vista, y el modelo de la vista.

- La vista: Son las clases xml del proyecto y en ellas, se muestra la presentación de los datos. Reacciona si se producen cambios en el ViewModel.
- Modelo de la vista: Son las clases .kt del proyecto. Encapsulan la lógica de la aplicación y se comunican con el modelo.
- Modelo: Encargado de obtener los datos de la base de datos. En nuestro proyecto es la base de datos de Firebase.

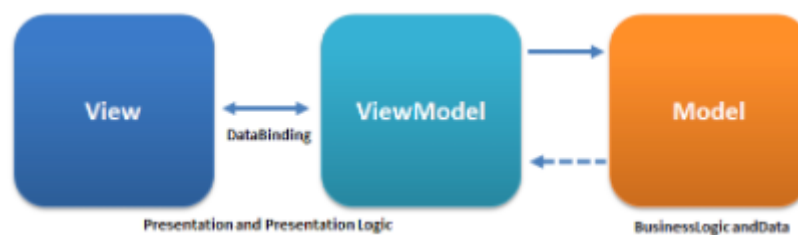


Figura 5.1 Patrón MVVM estructura

La figura representa la estructura de este patrón de diseño.

## 5.2 Modelo de datos

### 5.2.1 Modelo Jerárquico [7]

Este modelo es el empleado por la base de datos de Firebase. Se caracteriza por conceptualizarse como un árbol JSON[13] alojado en la nube, en la que los datos se almacenan como objetos JSON. Debido a ello, no hay tablas ni registros, a diferencia de una base de datos SQL, sino que, cuando se agregan datos al árbol JSON, estos se convierten en un nodo de la estructura JSON existente.

### 5.2.1 Información en la base de datos

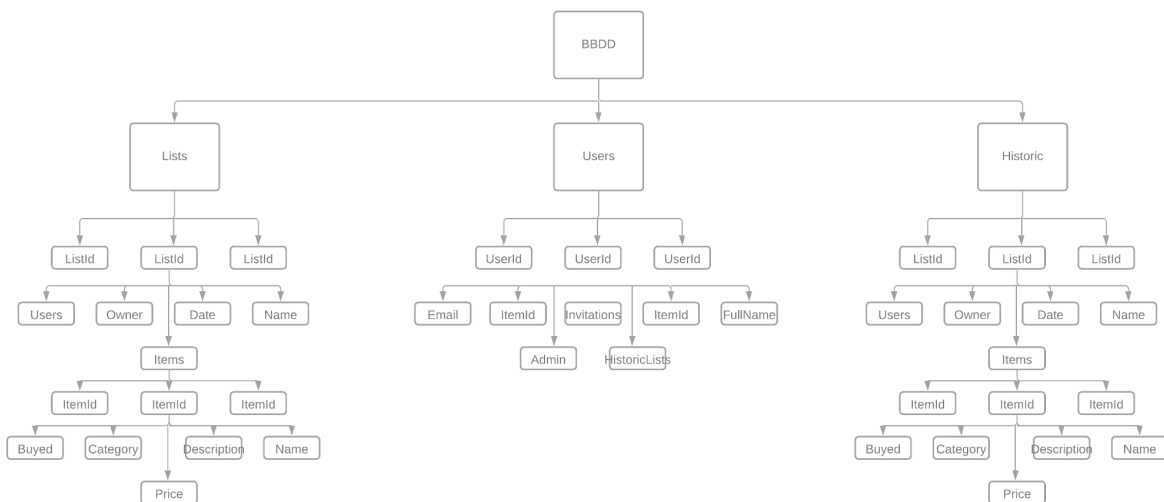


Figura 5.2 Diagrama jerárquico base de datos

En la figura de arriba puede verse cómo está estructurada la información en la base de datos, siguiendo el modelo jerárquico, previamente descrito. Como se puede observar, la base de datos se sustenta en tres pilares fundamentales, que se explican en detalle a continuación.

## Módulo listas.

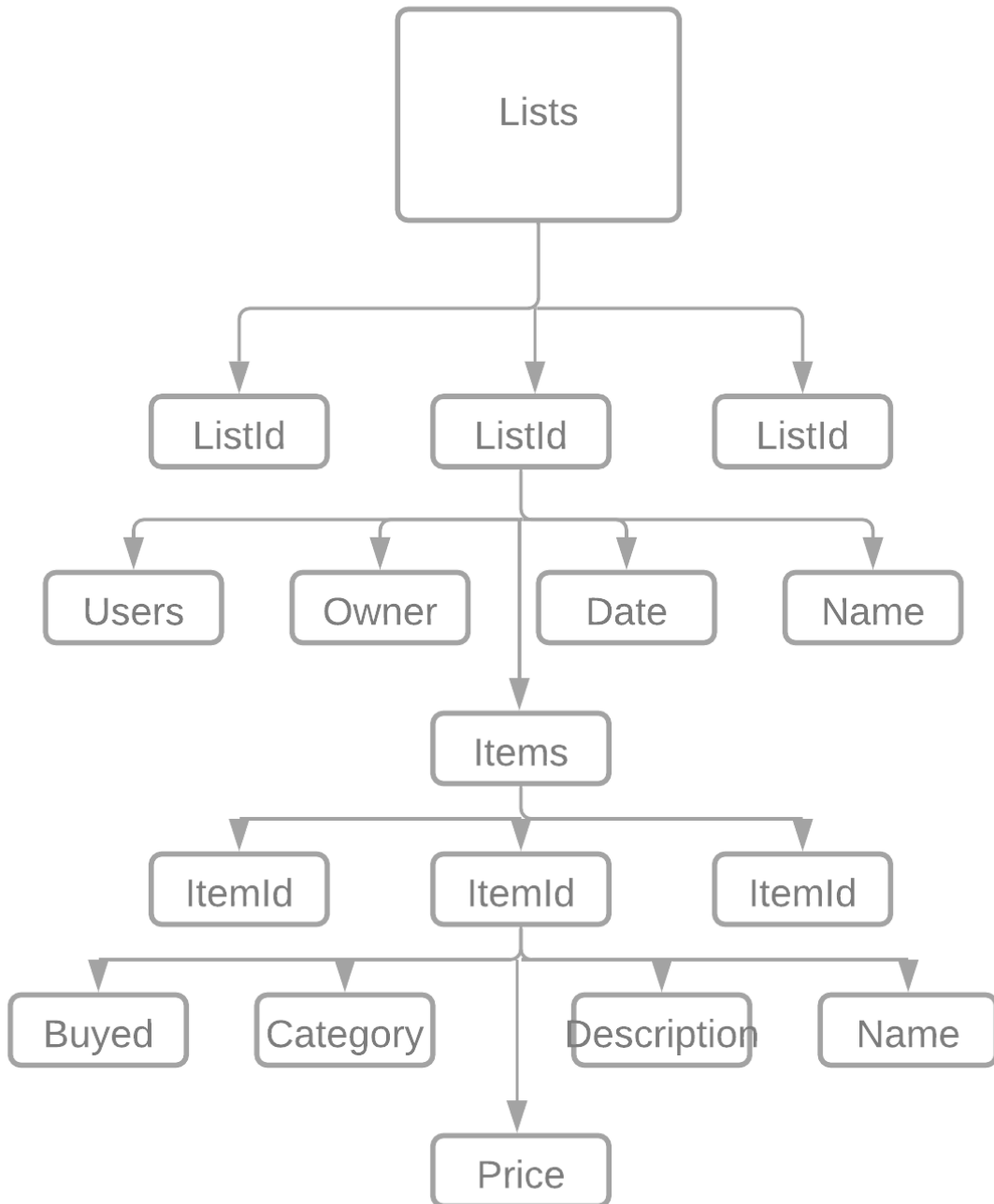


Figura 5.3 Diagrama módulo listas

En el módulo listas se incluyen todas las listas creadas por los usuarios en la base de datos. Estas listas tienen una serie de atributos:

- Id: identificador único de la lista en formato String.
- Date: fecha de creación de la lista en formato DDMMYYYY.
- Items: Consiste en el subárbol de productos que se forma dentro de cada lista. Este también tiene distintos atributos:
  - Id: Identificador único del producto en formato String.
  - Bought: Booleano que indica si el producto se ha comprado o no. False por defecto.
  - Category: La categoría a la que pertenece el producto.
  - Description: Una descripción para el producto en formato String.
  - Name: El nombre del producto en formato String.
  - Prize: Precio del producto en formato Float (este valor será siempre 0 hasta que se compre el producto)
- Name: Nombre de la lista en formato String.
- Owner: El id del usuario propietario de la lista en formato String
- Users: Los ids de los usuarios que pertenecen a la lista. Se trata de una lista de String separados por comas.

### Módulo Usuarios.

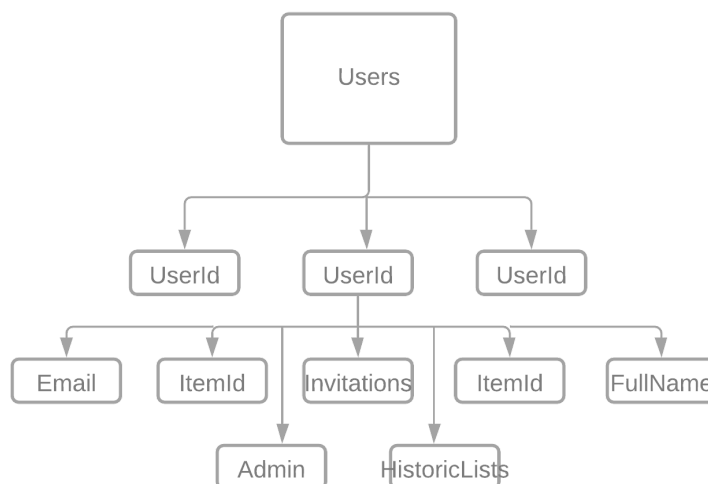


Figura 5.4 Diagrama módulo usuarios

En este módulo se incluyen todos los usuarios que han sido registrados en la base de datos, así como una serie de atributos característicos de cada uno:

- Id: Identificador único del usuario en formato String.
- Admin: Booleano que indica si el usuario es administrador o no.

- Email: Email del usuario.
- FullName: El nombre completo del usuario en formato String.
- HistoricLists: Contiene una cadena de Id de listas en formato String y separados por comas que reflejan las listas que tiene el usuario en el histórico.
- Invitations: Contiene una cadena de Id de listas en formato String y separados por comas que reflejan las invitaciones a listas que tiene el usuario.
- Lists: Contiene una cadena de Id de listas en formato String y separados por comas que reflejan las listas que ha creado el usuario.
- Username: El nombre de usuario del usuario en formato String.

## Módulo Histórico

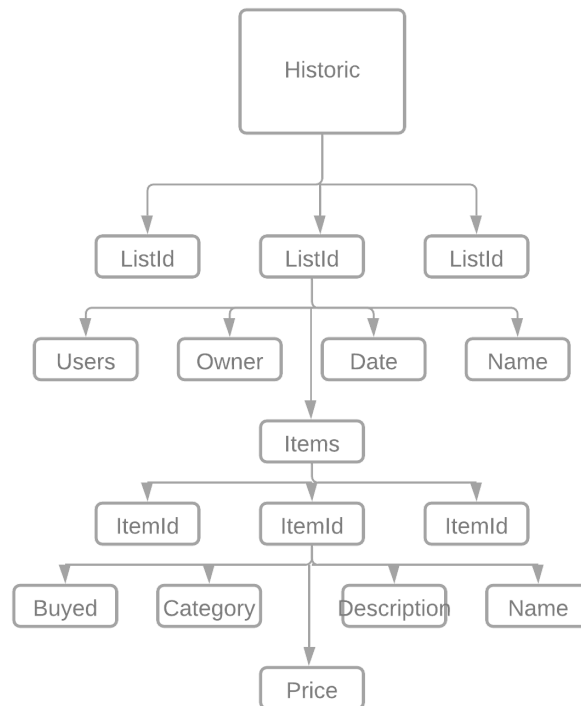


Figura 5.5 Diagrama módulo histórico

A este módulo pertenecen todas las listas que ya han sido compradas por el usuario correspondiente. Los campos son similares a los del módulo listas con algunas diferencias.

- Date: Ahora representa la fecha de compra de la lista.
- Items.

- Precio: Este valor deja de ser 0 y pasa a ser el precio de compra del producto
- Buyed: Pasa a ser true por defecto, ya que se ha comprado el producto.

## Implementación

En este capítulo se detallan los distintos elementos y la forma de implementación del código usados para el desarrollo de la aplicación.

### 6.1 Fase de diseño

#### 6.1.1 Elementos Android

En esta sección se darán a conocer los distintos elementos que se han utilizado para el desarrollo de la aplicación.

##### 6.1.1.1 Activities

Una “activity” se corresponde con un proceso o la ejecución de una tarea o fragmento de código. En este caso, se han utilizado varias “activities” [15] para poder ejecutar las distintas funcionalidades que tiene la aplicación.

##### 6.1.1.2 ListView

Las “ListView” [3] son como se declaran las listas en el código Kotlin. Estas listas se usan para mostrar los productos o las listas de la compra que un usuario crea.

##### 6.1.1.3 LinearLayout

Los “LinearLayout” [16] son un componente de la vista que hace que los elementos se muestren de forma lineal, es decir, los elementos unos debajo de otros.

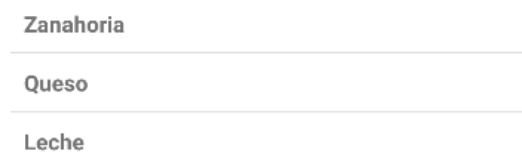


Figura 6.1: LinearLayout

#### 6.1.1.4 Bottom Navigation Bar [2]

Es un menú de navegación dispuesto en la parte inferior de la aplicación para poder navegar a una página distinta en un solo toque.



Figura 6.2: Bottom Navigation Bar

#### 6.1.1.5 Menu [17]

Este elemento consiste en un menú desplegable que contiene enlaces a otras partes o funcionalidades de la aplicación. Se han proporcionado distintos enlaces dependiendo de la página en la que se encuentra el usuario.

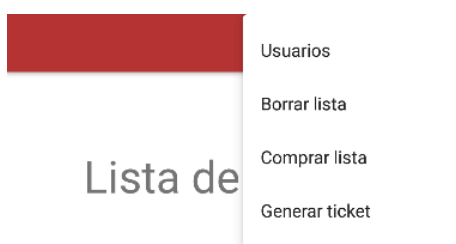


Figura 6.3: Menú

#### 6.1.1.6 Toolbar [1]

Este elemento es la barra superior de la aplicación, donde se encuentra el icono de notificaciones y el menú desplegable, además del nombre de la aplicación.



Figura 6.4: ToolBar

### **6.1.2 Base de datos**

Para la base de datos se ha usado la aplicación de Google Firebase[26]. Es una aplicación web que consiste en una base de datos que puede albergar grandes cantidades de información. Se decidió utilizarla por la comodidad de estar en la nube y la sencillez de uso, así como las grandes cantidades de datos que se podían introducir, además de las funcionalidades de backend que incluye.

Concretamente, se decidió utilizar una base de datos en tiempo real (Realtime), ya que es la que mejor se adapta a las necesidades del proyecto, ya que los datos de la aplicación pueden variar dependiendo de las acciones de los usuarios.

## **6.2 Clases principales**

La aplicación se basa en dos tipos de clases.

- Las clases .kt (Kotlin) son las que contienen la lógica de la aplicación.
- Las clases .xml son las que contienen la interfaz gráfica de la aplicación.

Para cada una de las clases Kotlin existentes que requieran de interfaz gráfica, existe una clase .xml asociada a esta. A continuación se explicarán las principales clases del proyecto haciendo hincapié en las clases .kt.

Al igual que la base de datos, la estructura de clases y las funcionalidades que recogen, se fundamentan en tres módulos principales.

### **6.2.1 Módulo usuarios**

En este módulo se incluyen las clases relativas a la interacción con la cuenta y/o el perfil del usuario.

#### **6.2.1.1 Delete Account/Edit Profile**

Estas clases se encargan de gestionar la cuenta del usuario. Se emplean para eliminar la cuenta de usuario o bien modificar la información de su perfil.

#### **6.2.1.2 Login/Logout**

Estas clases son las que se encargan de gestionar la sesión del usuario en la aplicación, siempre y cuando el mismo esté registrado. Cuando el usuario sale de la aplicación sin cerrar sesión, esta se mantiene activa y cuando el usuario vuelve a entrar carga directamente sus datos.

### **6.2.1.3 Profile**

Esta clase gestiona el acceso al perfil del usuario y su conexión con todas las funcionalidades recogidas en este.

### **6.2.1.4 Register**

Esta clase es la encargada de registrar al usuario en la base de datos con los datos que el usuario nos brinda.

## **6.2.2 Módulo listas**

En este módulo quedan recogidas las clases que permiten al usuario realizar todas las funcionalidades relativas a las listas.

### **6.2.2.1 UserHome**

Esta clase constituye la pantalla del menú principal de la aplicación, a partir de la cual el usuario puede navegar a través de ella.

### **6.2.2.2 CreateList/DeleteList/ViewList/EditNameList**

Estas clases son las que permiten a los usuarios gestionar sus listas personales. Permiten crear nuevas listas, eliminar sus listas o modificar el nombre de estas, y acceder a las listas creadas.

### **6.2.2.3 CreateProduct/DeleteProduct/ViewProduct**

Estas clases contienen la lógica relativa a la gestión de los productos de una lista determinada. Estas nos permiten crear, eliminar y editar la información de los productos, actualizando la base de datos.

### **6.2.2.4 InviteUsers/UserInvitations/UsersInList**

Esta clase contiene la lógica necesaria para permitir que un usuario invite a otros usuarios a su lista.

### **6.2.2.5 Buy/Summary/LeaveList**

Mediante esta clase el usuario puede comprar una lista de entre una de las que ha creado y que esta se lleve al histórico de listas.

## **6.2.3 Módulo histórico**

### **6.2.3.1 Copy List**

Esta clase permite al usuario copiar una lista desde el apartado histórico a la sección de sus listas.

### **6.2.3.2 HistoricLists**

Esta clase es la encargada de mostrar las listas que el usuario ha comprado anteriormente con sus respectivos productos y precios.

### **6.2.3.3 UserStats**

En esta clase se recoge la lógica para la creación y visualización de las estadísticas de las compras realizadas por el usuario.

## **6.2.4 Otras clases**

### **6.2.4.1 MainActivity [22]**

Es la actividad principal de la aplicación, a partir de la cual se lanzan las demás.

### **6.2.4.2 Clases Auxiliares**

Aquí se recogen diversas clases que se han requerido para el desarrollo de distintas funcionalidades.

#### **6.2.4.2.1 CustomAdapter [23]**

Esta clase se usa en la creación de listas que tienen más de un elemento por fila, como por ejemplo la funcionalidad de comprar una lista, donde se muestra el nombre del producto, el precio y una checkbox.

#### **6.2.4.2.2 DataModel [24]**

Es el modelo que usa el custom adapter para crear los objetos. Es una clase cuya única funcionalidad es tener tantos parámetros como distintos elementos se quieran mostrar.

## Evaluación

En este apartado se tratarán de exponer los resultados de una evaluación que se ha realizado a distintas personas, con un rango de edad distinto, relacionados con la usabilidad de la aplicación, con la interfaz, etc.

Esta evaluación se ha realizado a 3 hombres y 3 mujeres de entre 48 y 55 años y a 3 adolescentes entre 17 y 20 años.

Las tareas a realizar por los testers han sido las siguientes:

1. Registro en la aplicación
2. Inicio de sesión
3. Crear una nueva lista de la compra
4. Crear un producto dentro de esa lista
5. Borrar un producto de la lista
6. Añadir a un amigo a la lista
7. Comprar una lista
8. Aceptar una invitación a una lista
9. Ver el histórico de listas y aplicarle un filtro
10. Ver las estadísticas de las compras
11. Editar nombre de perfil
12. Borrar cuenta

A continuación se muestran los resultados obtenidos:

Me ha resultado interesante la aplicación

9 respuestas

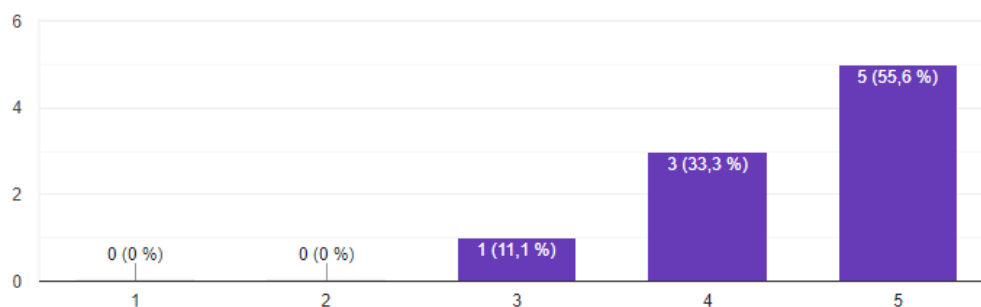


Figura 7.1: Pregunta 1

En la primera pregunta se trató de ver si a las personas que han probado nuestra aplicación les resulta interesante. Como se puede observar en la Figura 7.1, a 8 de las 9 personas les ha parecido bastante interesante, mientras que a otra no le ha parecido tanto, lo cual nos indica que la aplicación, en la mayoría de los casos, es interesante para la gente que la ha probado.

Me ha resultado atractiva la interfaz de la aplicación

9 respuestas

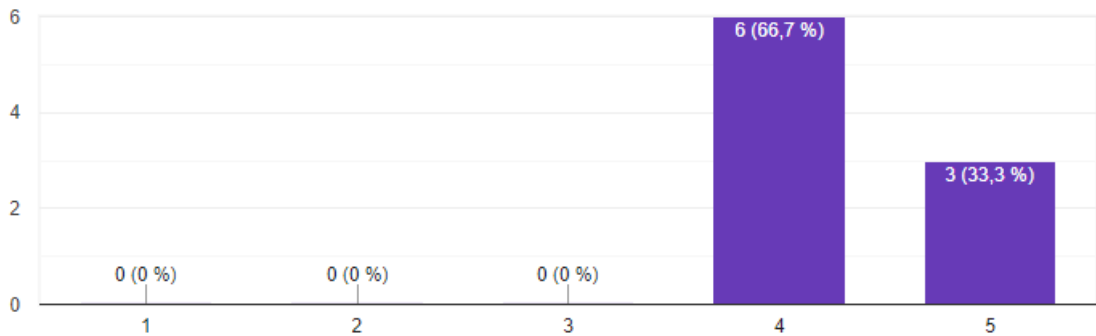


Figura 7.2: Pregunta 2

Observando la siguiente gráfica (ver Figura 7.2), se puede apreciar que la interfaz de la aplicación ha resultado atractiva a 6 de las 9 personas que han probado la aplicación, mientras que a 3 les ha parecido muy atractiva, lo que nos hace ver que se ha realizado un buen trabajo con la interfaz de la aplicación.

Me ha resultado fácil registrarme en la aplicación

9 respuestas

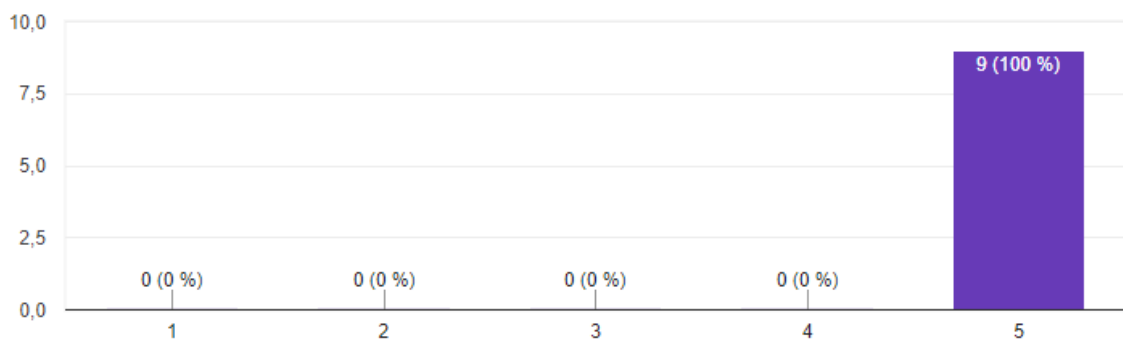


Figura 7.3: Pregunta 3

En esta pregunta (ver Figura 7.3) se trata la facilidad de registrarse en la aplicación. Se puede observar que al 100% de personas que la han testado les ha resultado muy fácil registrarse en ella, ya que nada más abrir la aplicación (si no se está registrado) aparece la opción de registrarse de una manera cómoda y sencilla.

Facilidad para crear una lista

9 respuestas

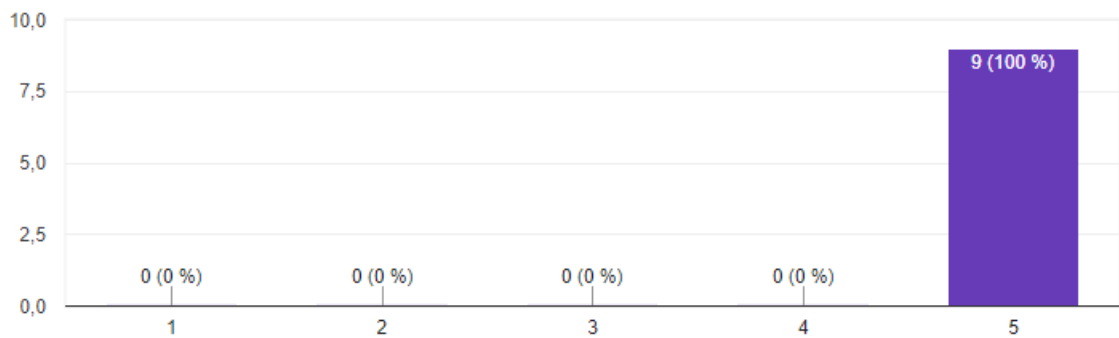


Figura 7.4: Pregunta 4

Esta pregunta (ver Figura 7.4) era de vital importancia que resultara lo más sencilla posible para los usuarios de la aplicación, ya que es la función principal de la misma. Se puede observar que se ha conseguido que resulte muy sencillo crear una lista en la aplicación, ya que el 100% de usuarios que la han probado les ha resultado muy sencillo crearla.

#### Facilidad para crear un producto en una lista

9 respuestas

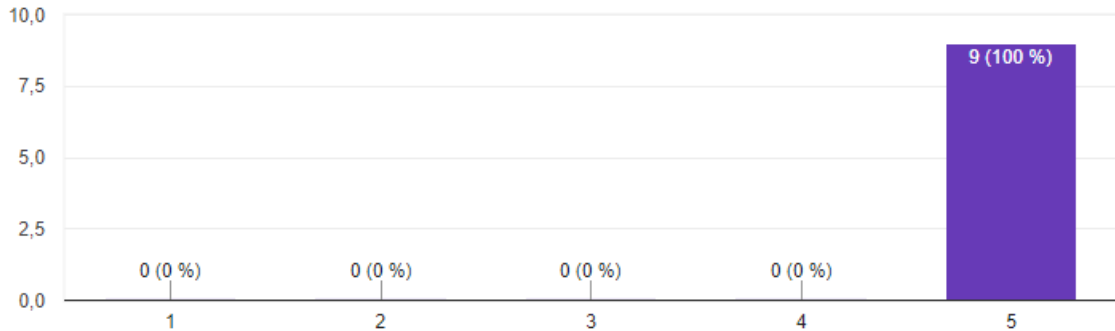


Figura 7.5: Pregunta 5

Al igual que la pregunta anterior (Figura 7.4), se puso mucho énfasis en la sencillez para añadir un producto a una lista de la compra. Como se puede observar (ver Figura 7.5), al 100% de usuarios testadores de la aplicación les ha parecido muy sencillo añadir un producto a una lista.

#### Facilidad para borrar un producto en una lista

9 respuestas

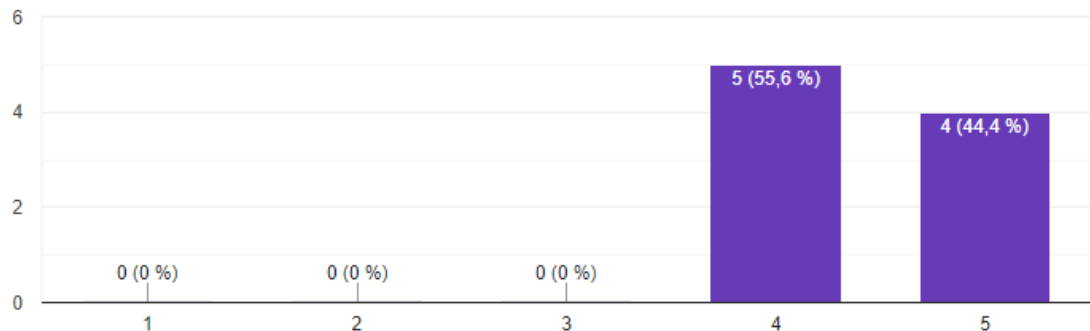


Figura 7.6: Pregunta 6

Como se puede observar en la Figura 7.6, a 5 de los 9 testadores les ha resultado fácil eliminar un producto de una lista, mientras que a 4 de los 9 testadores les ha resultado muy fácil hacerlo.

#### Facilidad para invitar a un amigo a una lista

9 respuestas

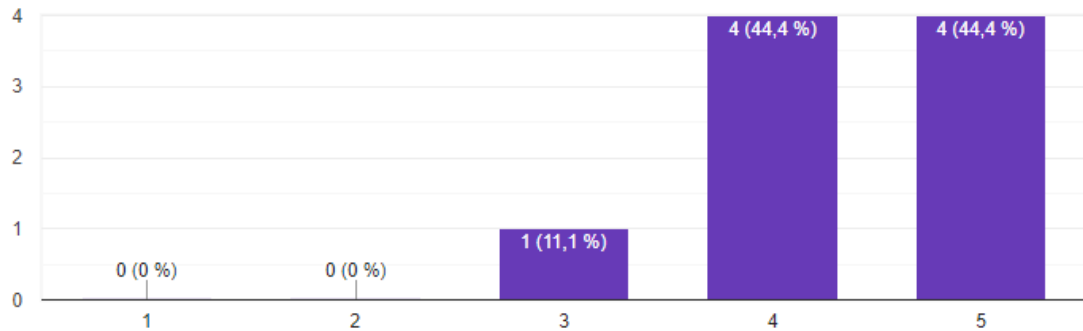


Figura 7.7: Pregunta 7

En esta pregunta (ver Figura 7.7) se puede observar que la mayor parte de las personas que realizaron la prueba, no tuvieron problemas en completarla. Sin embargo, se puede apreciar que hubo una persona que tuvo ciertas dificultades para completar la tarea.

#### Facilidad para comprar una lista

9 respuestas

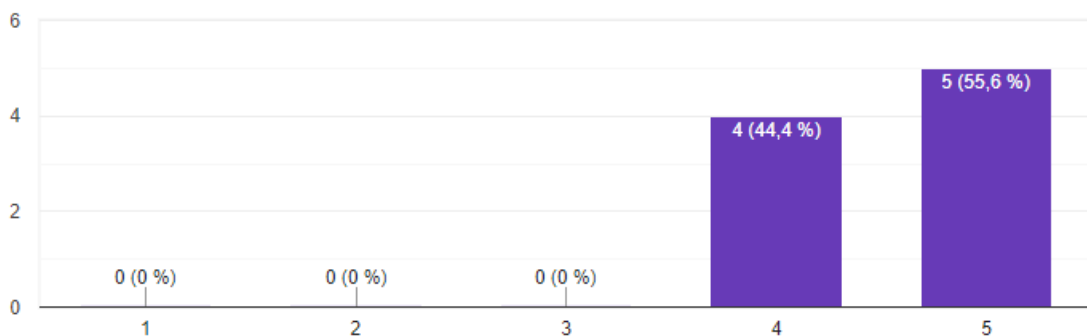


Figura 7.8: Pregunta 8

Esta tarea (ver Figura 7.8), era de vital importancia que fuese sencilla llevarla a cabo, ya que es una tarea importante en la aplicación. Se observa que a 4 de las 9 personas les ha resultado sencillo realizarla, mientras que a 5 de las 9 personas les ha resultado muy sencillo.

#### Facilidad para aceptar una solicitud a una lista

9 respuestas

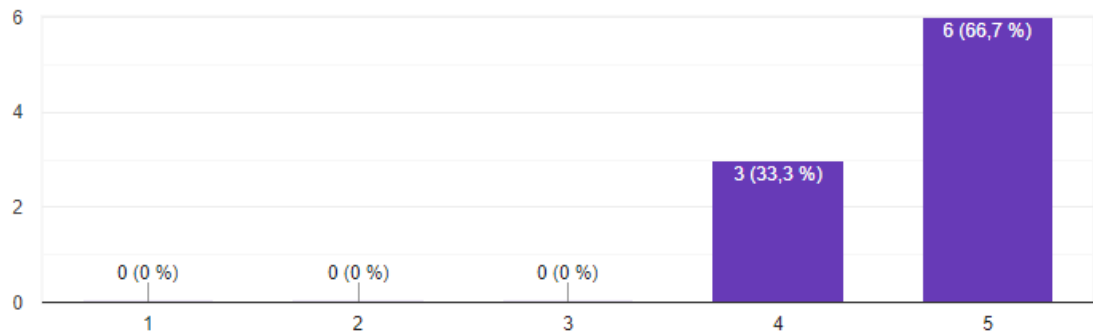


Figura 7.9: Pregunta 9

Como indica la Figura 7.9, a todos los usuarios les ha resultado fácil o muy fácil aceptar una solicitud a una lista de un amigo para crear y compartir productos.

#### Facilidad para ver el histórico de listas

9 respuestas

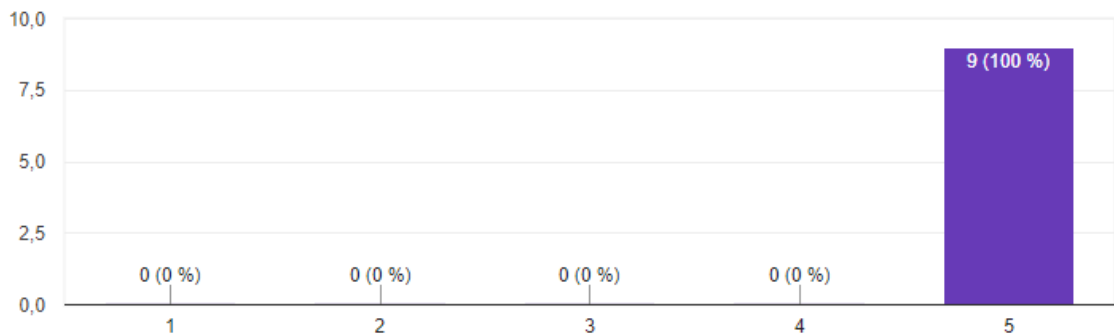


Figura 7.10: Pregunta 10

Como se puede observar en la Figura 7.10, a todos los usuarios que han probado la aplicación les ha resultado muy sencillo acceder al histórico de las listas.

### Facilidad para ver las estadísticas

9 respuestas

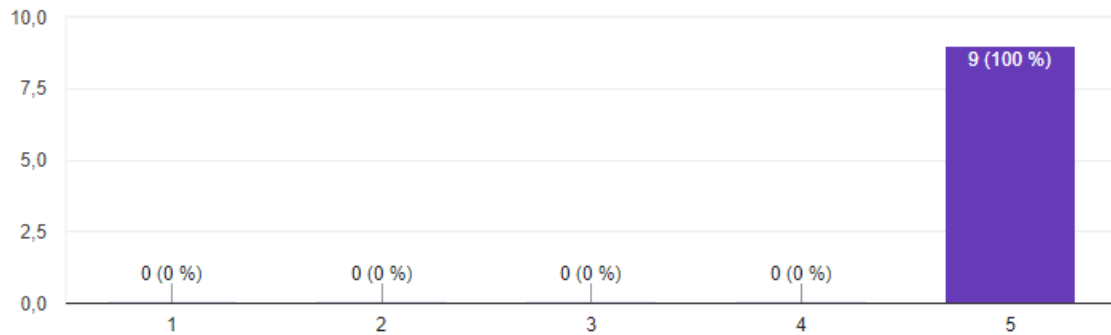


Figura 7.11: Pregunta 11

Al igual, como se puede observar en la Figura 7.11, al 100% de los usuarios testeadores les ha parecido muy sencillo acceder a las estadísticas de sus compras.

### Facilidad para editar el nombre de usuario

9 respuestas

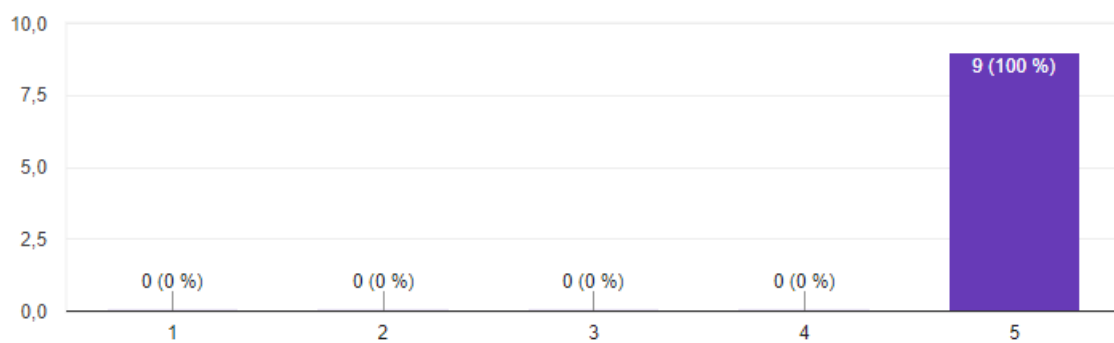


Figura 7.12: Pregunta 12

### Facilidad para borrar la cuenta del usuario

9 respuestas

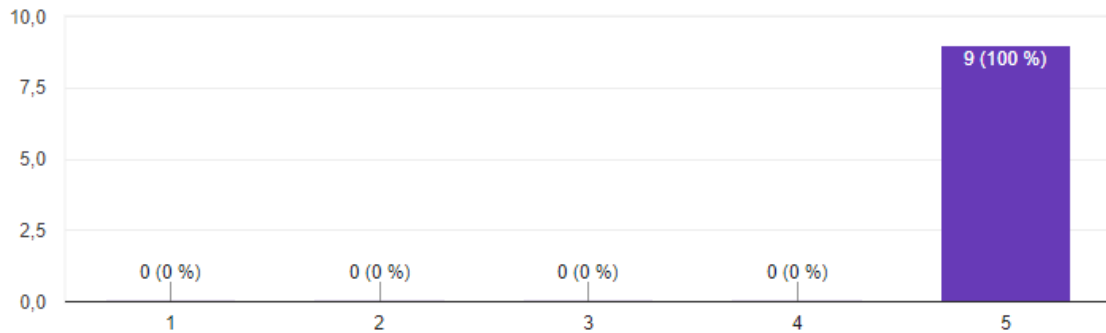


Figura 7.13: Pregunta 13

Se observa en la Figura 7.12 que a todos los usuarios que han probado la aplicación les ha parecido muy sencillo editar el nombre de usuario, al igual que borrar la cuenta de usuario (ver Figura 7.13).

# Conclusiones y trabajo futuro

## 8.1 Conclusiones

Nuestra aplicación presenta como principales funcionalidades la creación, gestión y eliminación de listas de la compra, así como de todo lo incluido dentro de ellas (productos, usuarios...), la inclusión de un historial de listas para la obtención de estadísticas de compras del usuario y una función social mediante la invitación a listas entre usuarios.

Se ha implementado una aplicación orientada hacia cualquier tipo de persona ya que ofrece una solución a un problema que puede afectar a cualquier persona. Las principales características de esta aplicación serían que es intuitiva y sencilla de utilizar para cualquier tipo de usuario, priorizando estas propiedades por encima de la complejidad de funcionalidades, de modo que pueda ser utilizada por cualquier tipo de persona. También se ha pretendido que la aplicación sea interactiva entre usuarios, dando la opción de compartir listas entre personas.

Sin embargo, es cierto que nuestra aplicación presenta algunos inconvenientes, como la imposibilidad de cambiar el idioma de esta o la falta de estadísticas para ofrecer al usuario una mayor cantidad de datos sobre sus compras.

## 8.2 Trabajo futuro

- Conectar aplicación con tecnologías de voz
  - Una idea para realizar en un futuro sería permitir a los usuarios conectar la aplicación con tecnologías de reconocimiento de voz, como Alexa. De modo que, a través de comandos de voz, puedan realizar algunas de las funcionalidades.
- Uso de Google Vision API. [21]
  - Se trata de una API de reconocimiento de imágenes. Se podría incorporar esto a este proyecto para facilitar al usuario añadir nuevos productos.
- Multilinguaje
  - La incorporación de nuevos idiomas para la aplicación puede servir para extender el uso de esta por diferentes países el tener una opción de cambiar el idioma al que pertenezca el usuario.

- Recomendaciones personalizadas para cada usuario.
  - Mediante el uso de la inteligencia artificial, se podría estudiar los patrones de compra de cada usuario para que la aplicación le haga recomendaciones para sus compras futuras.
  
- Mejorar las estadísticas
  - Actualmente, las estadísticas que se muestran al usuario son limitadas. Se podrían incorporar nuevas estadísticas para mejorar la información que el usuario obtiene de sus compras.
  
- Interacción con otras aplicaciones
  - Otra posible mejora para un futuro sería conectar la aplicación con otras aplicaciones de reparto de comida. De este modo, el usuario crearía su lista, y un repartidor se la enviaría a su casa.

# Conclusions and future work

## 8.1 Conclusions

This application presents as main functionalities the creation, management and deletion of shopping lists, as well as everything included in them (products, users...), the inclusion of a list history to obtain statistics of the user's purchases and a social function through the invitation to lists among users.

An application oriented towards any type of person has been implemented since it offers a solution to a problem that can affect any person. The main features of this application would be that it is intuitive and simple to use for any type of user, prioritizing these properties over the complexity of functionalities, so that it can be used by any type of person. We have also intended the application to be interactive between users, giving the option to share lists between people.

However, it is true that this application has some drawbacks, such as the impossibility to change the language of the application or the lack of statistics to provide the user with more data about their purchases.

## 8.2 Future Work

- Connecting applications with voice technologies.
  - One idea to realize in the future would be to allow users to connect the app with voice recognition technologies, such as Alexa. So that, through voice commands, they can perform some of the functionalities.
- Use of Google Vision API.
  - This is an image recognition API. It could be incorporated into this project to make it easier for the user to add new products.
- Multilanguage
  - The incorporation of new languages for the application can serve to extend the use of this by different countries to have an option to change the language to which the user belongs.
- Personalized recommendations for each user.
  - Through the use of artificial intelligence, it would be possible to study the buying patterns of each user so that the application could make recommendations for future purchases.

- Improve statistics
  - Currently, the statistics shown to the user are limited. New statistics could be incorporated to improve the information the user obtains from his purchases.
  
- Interaction with other applications
  - Another possible improvement for the future would be to connect the application with other food delivery applications. In this way, the user would create his list, and a delivery person would send it to his home.

## Aportaciones individuales

### 9.1 Jesús Cencerrado Pedrero

Mi trabajo al comienzo de la aplicación fue explorar cómo se podía hacer una aplicación basada en Android (herramientas a utilizar, lenguaje de programación de la misma...). Todos los miembros del grupo llegamos a la conclusión que la aplicación se desarrollaba en Kotlin mediante el programa Android Studio. Hubo una pequeña discusión sobre si desarrollar la aplicación en Java o en Kotlin y, como nunca nos habíamos enfrentado a Kotlin (y viendo que estaba basado en Java) nos lanzamos a desarrollarlo en este lenguaje para así tener una idea de cómo desarrollar una aplicación real.

También estuvimos discutiendo acerca de qué base de datos usar, si una en phpmyadmin (localhost) y acto seguido subirla a un servidor o una base de datos en la nube. Al final nos decantamos por la base de datos en la nube ya que nos parecía más cómodo y ya estaba en la nube, por lo que nos ahorramos el paso de tener que subirla a un servidor.

Después de esto, estuve indagando en cómo se estructura la aplicación. En un primer momento empezamos a desarrollar la aplicación mediante fragmentos (fragments), pero más tarde pensamos en hacerlo de otra manera ya que nos parecía más cómodo no usar los fragmentos porque dificulta (a nuestro entender) la comunicación entre las distintas partes de la aplicación y decidimos utilizar actividades (Activities) directamente.

A partir de aquí, yo fui el encargado de ver cómo mostrar las listas visualmente hablando (mediante el archivo xml) y del desarrollo de la vista de la interfaz además del desarrollo de distinto código en distintas partes de la aplicación y de algunas funciones como la SplashScreen (que es la pantalla que aparece nada más abrir la aplicación mostrando su logotipo).

Antes de dar los últimos retoques a la aplicación, fui el encargado de desarrollar la memoria con todos los apartados que podíamos completar antes de presentar la aplicación para así adelantar trabajo y dejar la estructura hecha para después completar las partes que no hayamos podido completar hasta que la aplicación no estuviese terminada.

### 9.2 Sergio Morán Agüero

Mi primera aportación al proyecto consistió en una investigación individual sobre las distintas formas de implementar una aplicación en Android, la cual fue realizada también por mis compañeros. Tras esta primera investigación llegamos a la

conclusión de que la forma más eficiente de realizar el proyecto era mediante Android Studio y empleando el lenguaje de programación Kotlin.

Una vez decidimos el entorno y el lenguaje de programación pasamos con la siguiente cuestión, la base de datos. Para ello, decidimos emplear Firebase debido a que nos presentaba mayores ventajas que la otra opción.

Tras decidir las piezas fundamentales del proyecto comenzamos con el diseño, creación e implementación de la aplicación en sí.

Primero comenzamos con el diseño de la base de datos, donde participé muy activamente fijando la estructura principal de esta.

Más adelante, comenzamos con la implementación de las funcionalidades de la aplicación.

En este apartado, mi aportación ha sido básicamente el desarrollo y depuración del código de dichas funcionalidades, así como la corrección de los errores que nos iban apareciendo mientras avanzábamos. Destacando, sobre todo, mi aportación en la parte de la lógica del código y menos en la parte de la interfaz.

Además, me encargué de buena parte del desarrollo de la memoria, así como de la corrección de distintos puntos de esta y la supervisión del formato final para su entrega.

### **9.3 Pablo Fernández de la Moya**

Mi primera aportación al proyecto, al igual que mis compañeros, fue investigar sobre las formas de desarrollar una aplicación en Android. Principalmente la herramienta, ya que busqué una que fuera compatible con MacOS y Windows debido a que yo trabajo con el entorno de Apple. Junto con la aprobación del profesor, llegamos a la conclusión de utilizar Android Studio y Kotlin.

El siguiente paso fué elegir cómo íbamos a gestionar los datos de la aplicación. Finalmente nos decidimos por Firebase, ya que yo personalmente ya había trabajado con ello y porque nos facilitaba mucho las cosas a la hora de conectarlo.

Una vez llegados a este punto, empezamos a desarrollar. Inicialmente estuve mano a mano con Sergio desarrollando la estructura de la BBDD. A partir de ahí, yo me encargué sobre todo de la parte lógica de la aplicación, la cual ha requerido mucho trabajo de investigación para conseguir determinadas funciones, como por ejemplo: controlar la autenticación del usuario, recuperar y guardar los datos en la aplicación, implementar diferentes tipos de objetos, utilizar paquetes externos (como el que usamos para representar los gráficos en las estadísticas), etc...

También he aportado de forma considerable en la parte de la interfaz gráfica, la cual también me ha requerido trabajo de investigación.

La parte más destacable de mi trabajo ha sido el apartado de estadísticas del usuario.

Por último también he realizado aportaciones en la memoria, sobre todo aquellas que iban enfocadas y relacionadas con la lógica y el funcionamiento de la aplicación.

# Apéndices

## Manual de usuario

En este apartado se hará una descripción detallada de la interacción del usuario con la aplicación, incluyendo imágenes de esta para facilitar su comprensión.

Al iniciar la aplicación se muestra un "Splash" a modo de pantalla de bienvenida, mostrando el icono de la aplicación como se puede observar en la Figura 10.1.



Figura 10.1: Splash - Primera pantalla

Después de mostrar esta pantalla, se redirige al usuario a la pantalla de Login.  
Figura 10.2



Figura 10.2: Inicio de la aplicación

En caso de disponer de una cuenta, el usuario puede iniciar sesión en la aplicación mediante su email y contraseña. En caso contrario, el usuario puede pulsar el botón de "Regístrate" para crear una nueva cuenta. Si lo hace, se le redirige a la pantalla de registro. Figura 10.3.

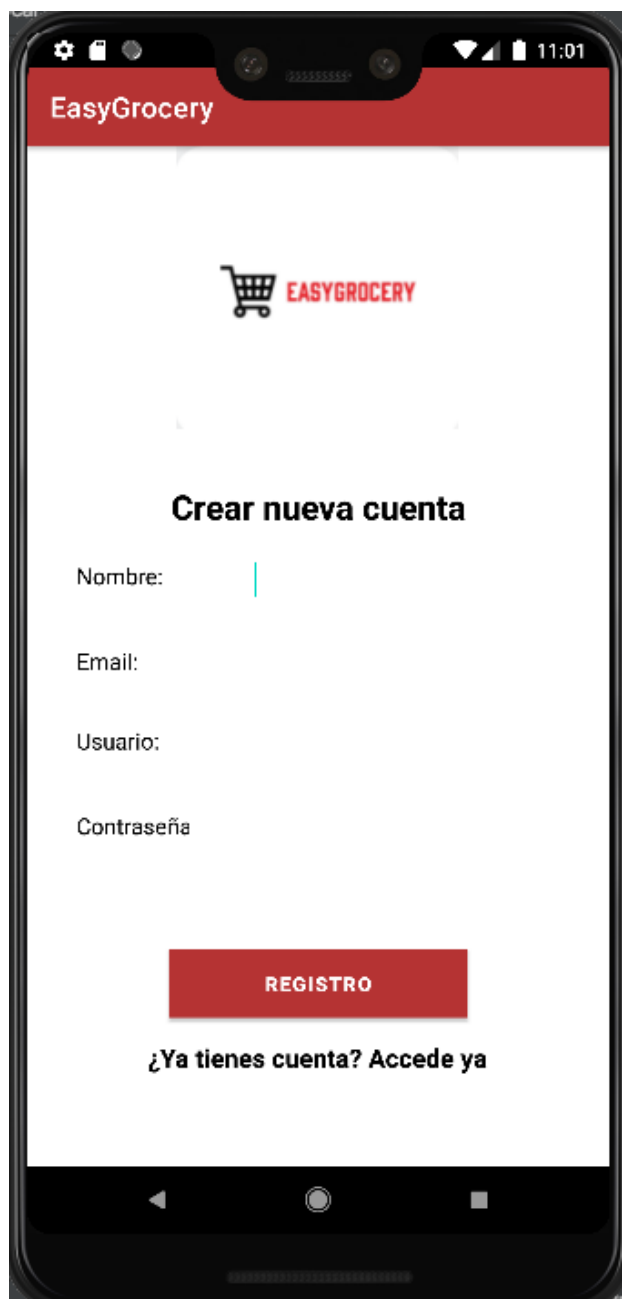


Figura 10.3: Registro

En esta pantalla, el usuario debe rellenar los campos especificados y después pulsar el botón de aceptar. Si todos los campos han sido rellenados correctamente, se crea la cuenta del usuario y se accede automáticamente a esta, redirigiendo al usuario

a la pantalla principal de la aplicación. Figura 10.4.

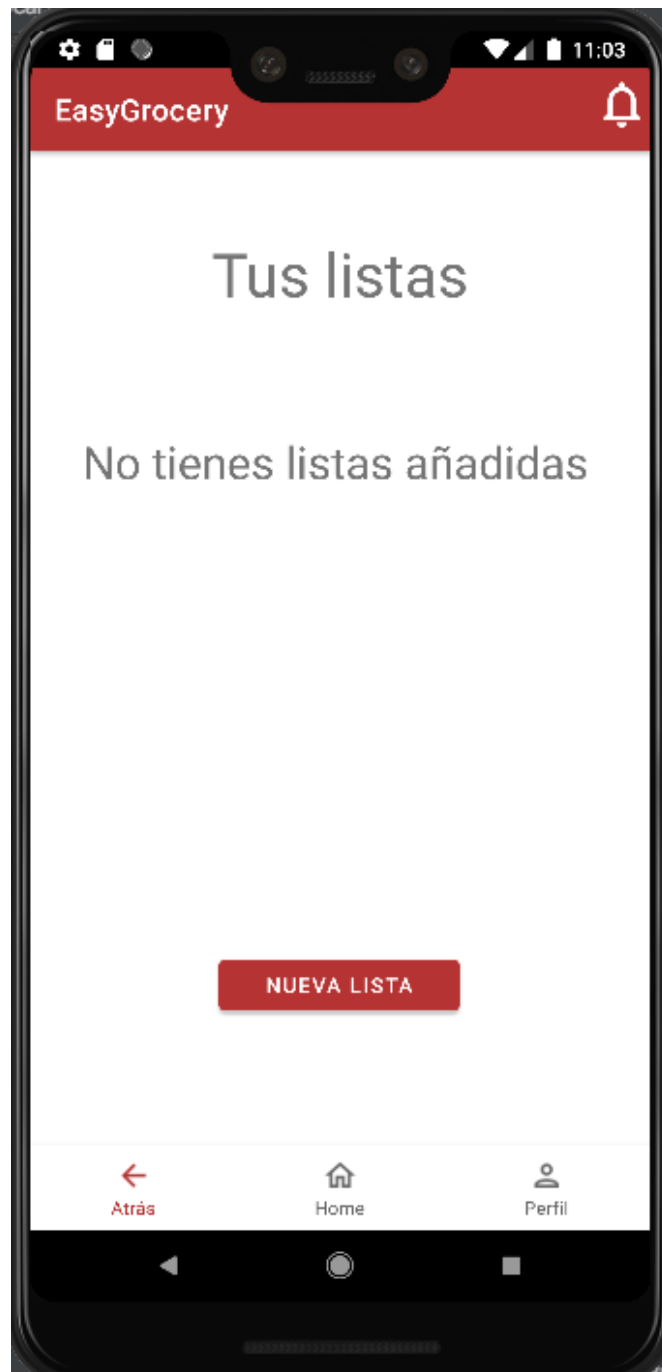


Figura 10.4: Página principal

En esta pantalla se le presentan varias opciones al usuario. En la parte de abajo de esta, se encuentra el menú de navegación, mediante el cual el usuario puede navegar a través de los distintos apartados de la aplicación. Cuenta con:

- Volver hacia atrás (no aplicable en esta pantalla)
- Volver a home

- Ir al perfil del usuario.

Además, situado en el centro de la pantalla, el usuario tiene el botón de crear lista, mediante el cual comienza el proceso para crear una lista.

Si se selecciona el botón de crear lista, se redirige al usuario a otra pantalla como en la Figura 10.5, en la que el usuario debe introducir un nombre para la nueva lista que desea crear.

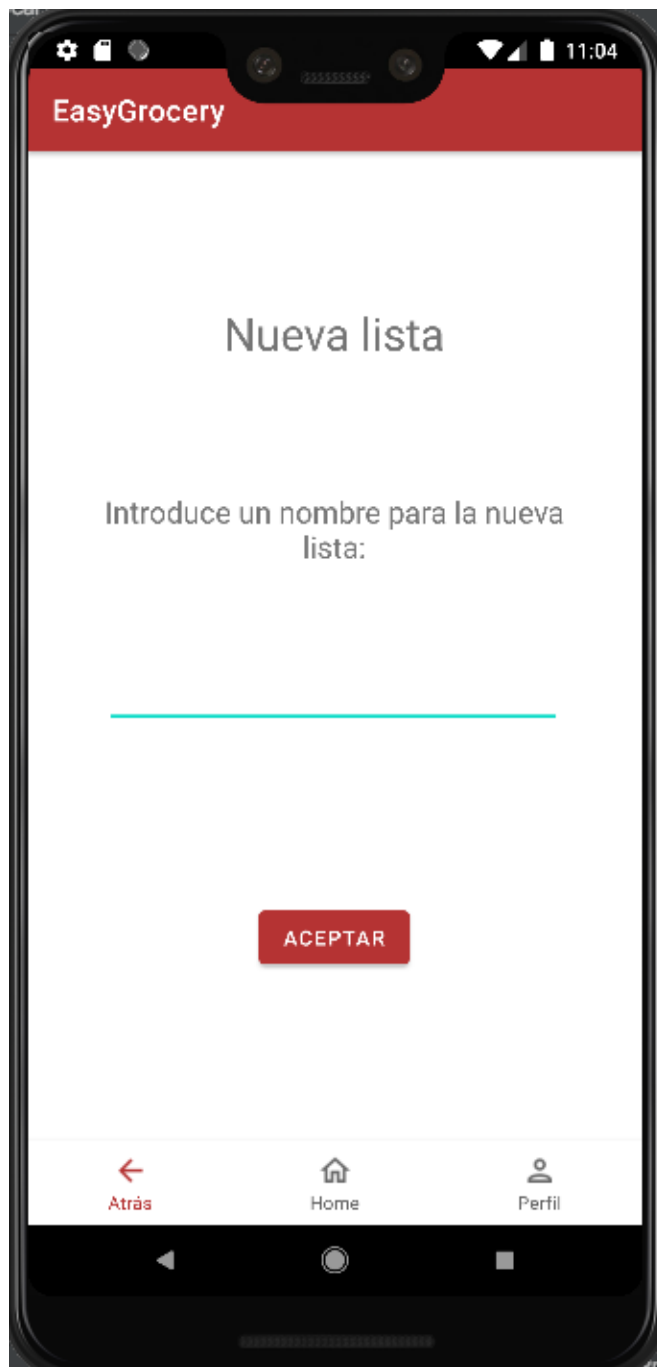


Figura 10.5: Añadir nombre a la lista

Dicho nombre no puede ser vacío ni mayor de 16 caracteres. Una vez introducido un nombre válido, el usuario pasa a la pantalla principal de la lista, como en la Figura 10.6.



Figura 10.6: Productos de una lista vacía

En esta pantalla se le muestran al usuario los productos introducidos a dicha lista (Figura 10.7) y, en caso de no tener ninguno, un mensaje de lista vacía.

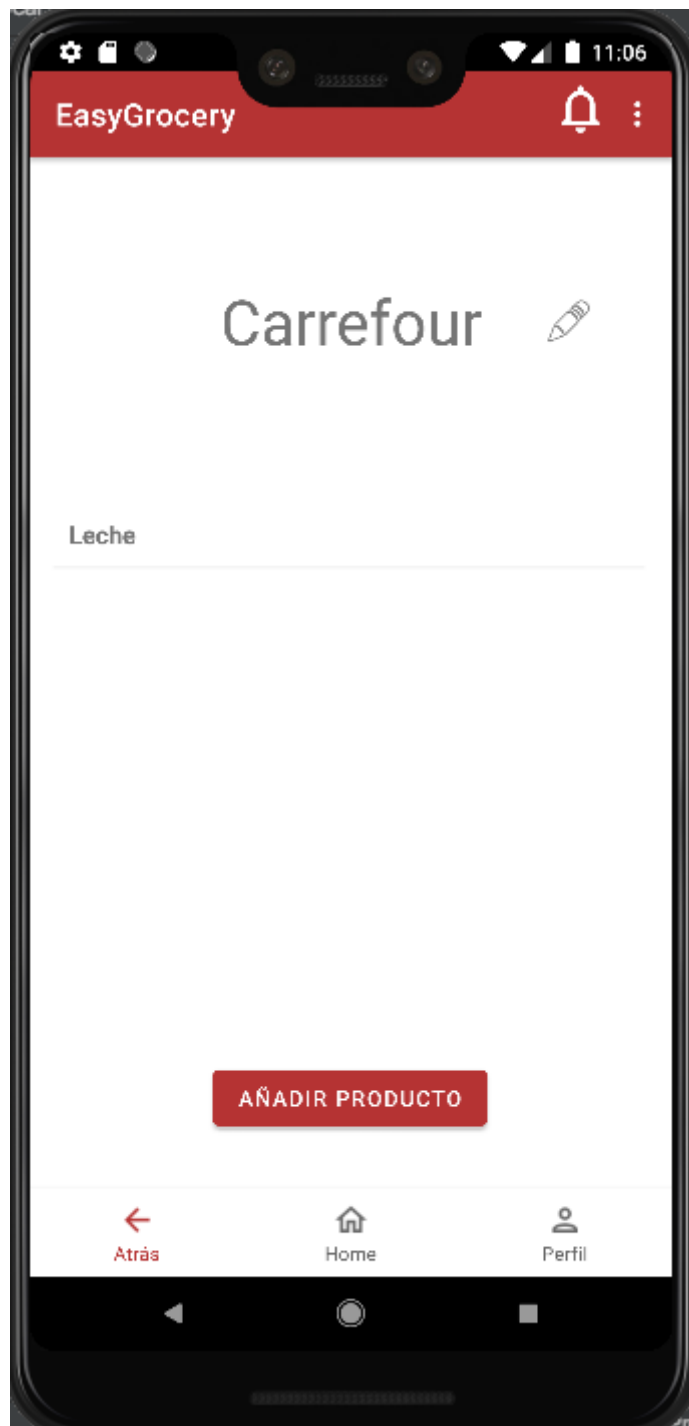


Figura 10.7: Productos de una lista

Además, el usuario dispone de un botón de añadir productos a la lista. Al pulsarlo se le lleva a una pantalla en la que debe rellenar un pequeño formulario para añadir el producto, como en la Figura 10.8.

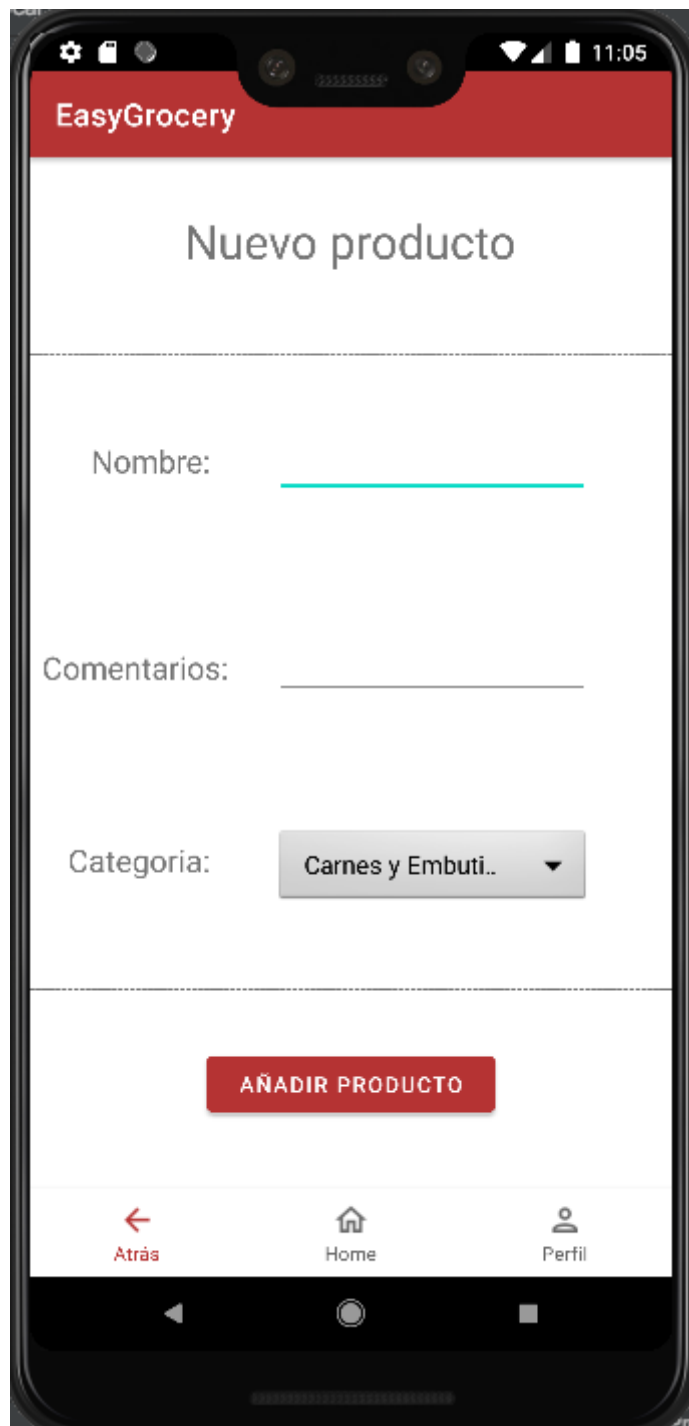


Figura 10.8: Formulario de un producto

En dicho formulario, el usuario no puede:

- Introducir un nombre vacío o mayor de 16 caracteres
- Introducir un comentario mayor de 150 caracteres.

Tras rellenarlo y pulsar el botón de aceptar, el producto se añade a la lista y se redirige al usuario a la pantalla principal, donde puede ver el nuevo producto añadido (Figura 10.7)

Al pulsar en el producto, el usuario accede a él, y se le lleva a una nueva pantalla como la de la Figura 10.9

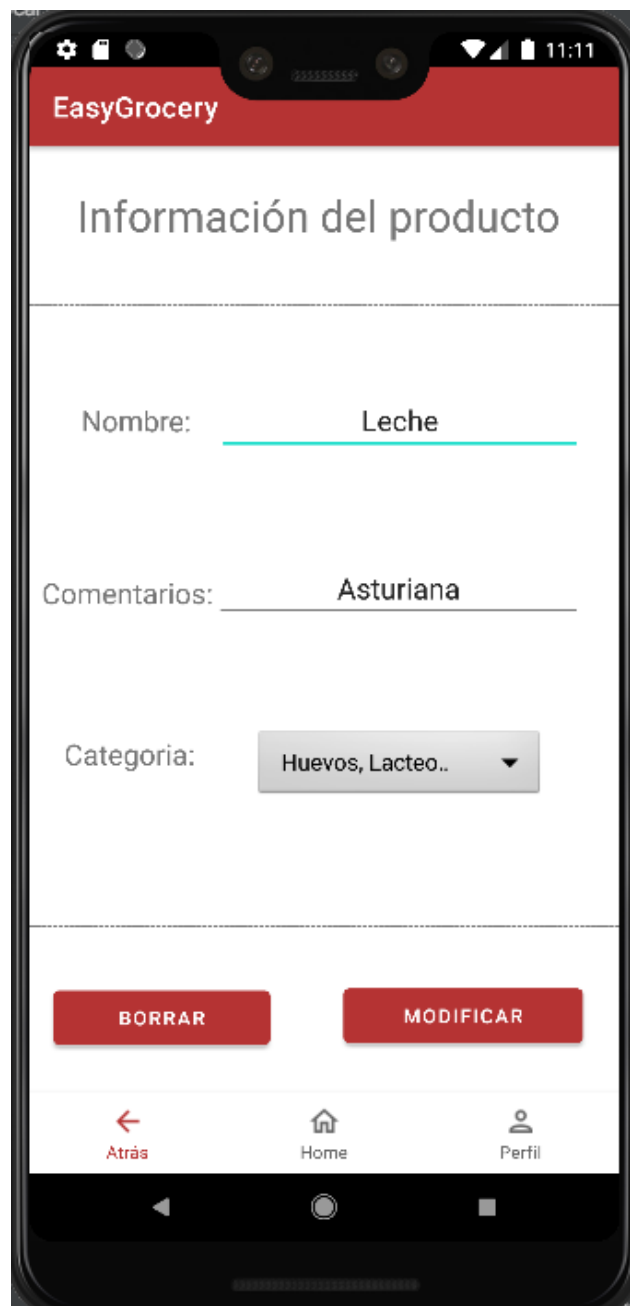


Figura 10.9: Dentro de un producto

Aquí el usuario dispone de varias opciones:

- **Eliminar:** Un botón que sirve para eliminar el producto de la lista. Si se pulsa, el producto quedará eliminado y el usuario volverá a la pantalla anterior.
- **Modificar:** El usuario puede modificar la información del producto, y, al pulsar en modificar, esta información será actualizada y el usuario volverá a la pantalla anterior. (Figura 10.7)

En esta misma pantalla, en la parte superior, dispone de un menú de opciones para interactuar con la lista. En función del estado de la lista, el usuario dispone de distintas opciones en dicho menú.

Si la lista tiene productos añadidos, el menú se verá como en la Figura 10.10.

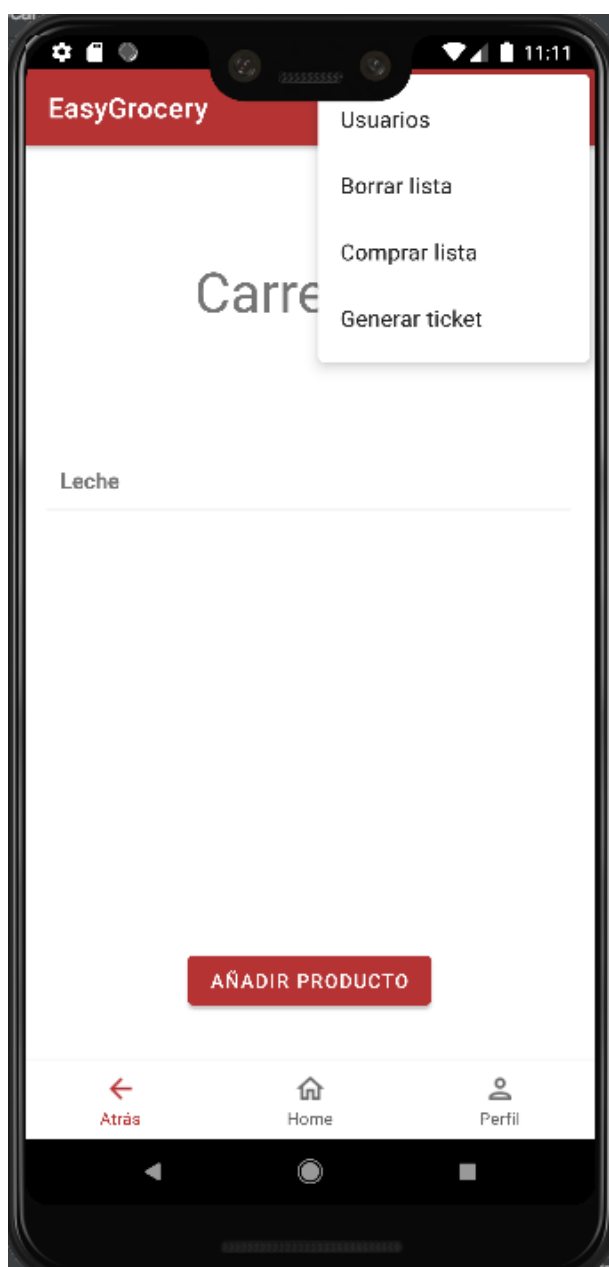


Figura 10.10: Menú de lista con productos

En caso contrario, se verá como en la Figura 10.11.



Figura 10.11: Menú de lista sin productos

Como se puede observar, el usuario dispone de cuatro distintas opciones en el menú de la Figura 10.10:

-Usuarios: Si el usuario accede a este apartado, se le redirige a una pantalla como la Figura 10.12.



Figura 10.12: Usuarios de una lista

En esta pantalla puede ver los usuarios que han sido añadidos a la lista mediante invitación y, además, tiene la posibilidad de eliminarlos con hacer click en

ellos. También dispone de un botón para invitar a más usuarios. Al pulsarlo, avanza a otra pantalla donde debe introducir el email del usuario que desea invitar y, tras darle a aceptar, se enviará una invitación al usuario. Figura 10.13



Figura 10.13: Invitar usuarios

-Borrar lista: Si el usuario pulsa este botón se borrará la lista y todos sus productos de la base de datos y será redirigido a la pantalla de home.

-Abandonar lista: Esta opción solo está habilitada cuando el usuario no es el dueño de la lista y sirve para abandonar la lista cuando el usuario lo desee.

-Comprar lista: Al hacer click en esta opción se redirige al usuario a una pantalla donde puede ver todos los productos que tiene en la lista, como en la Figura 10.14.



Figura 10.14: Comprar lista

Según va haciendo click en cada uno, aparecerá una ventana emergente como la de la Figura 10.15 para que introduzca el precio de compra del producto y pulse en aceptar.



Figura 10.15: Popup comprar lista

Una vez pulse aceptar, se actualiza la pantalla, quedando el producto como marcado y el precio establecido, como se puede ver en la Figura 10.16. En la parte de abajo, el usuario dispone de un botón de finalizar compra que solo estará habilitado cuando haya al menos un producto marcado. Si el usuario lo pulsa, la lista será comprada y se pasará al historial de listas. El usuario será redirigido entonces a la pantalla principal.

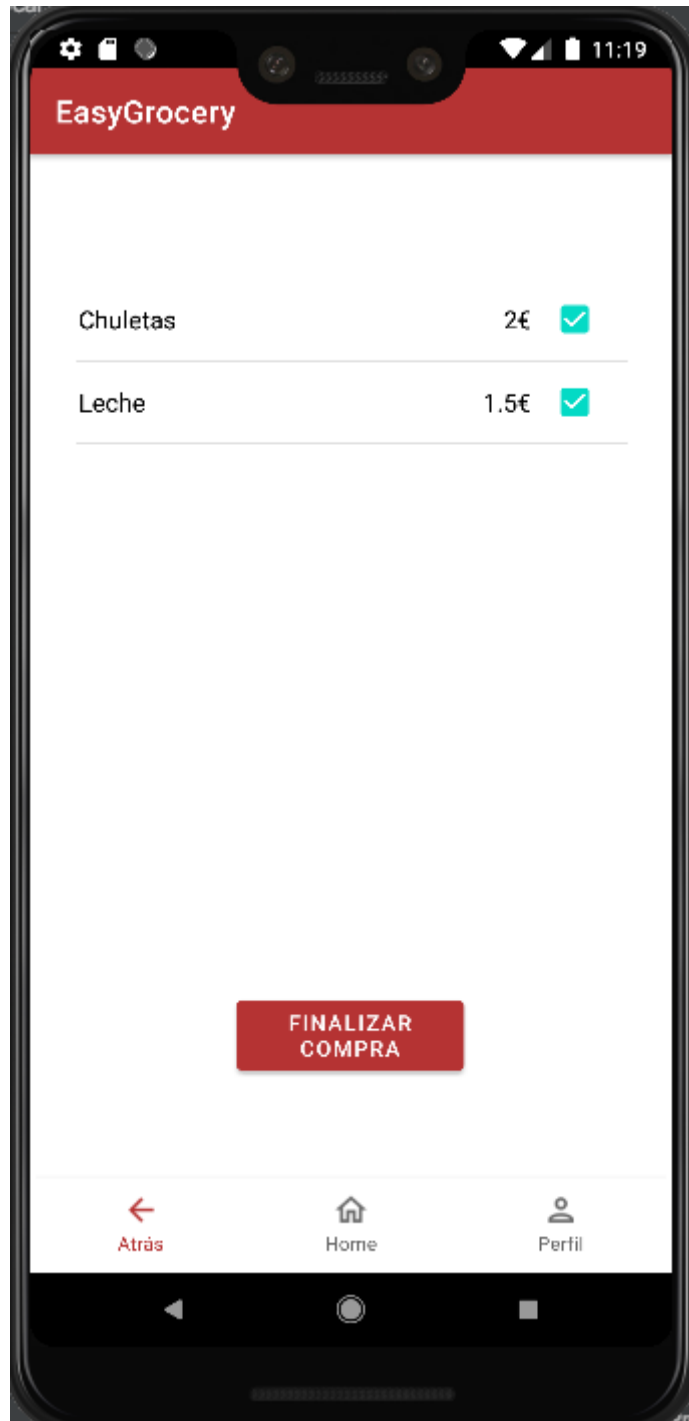


Figura 10.16: Comprar lista con productos marcados

-Generar ticket: Al pulsarlo, se redirige al usuario a una pantalla como en la Figura 10.17 , donde el usuario puede ver los productos añadidos a la lista con su precio de compra al lado y el total más abajo, con un formato de "ticket".



Figura 10.17: Generar ticket

Si el usuario por el contrario pulsa en el botón de perfil accede a una nueva pantalla como la Figura 10.18. En esta se puede ver el nombre del usuario y una foto de perfil por defecto. Además, cuenta con varias opciones para el usuario.



Figura 10.18: Perfil del usuario

En primer lugar, en la parte superior se puede ver un menú desplegable que cuenta con dos opciones (Figura 10.19)

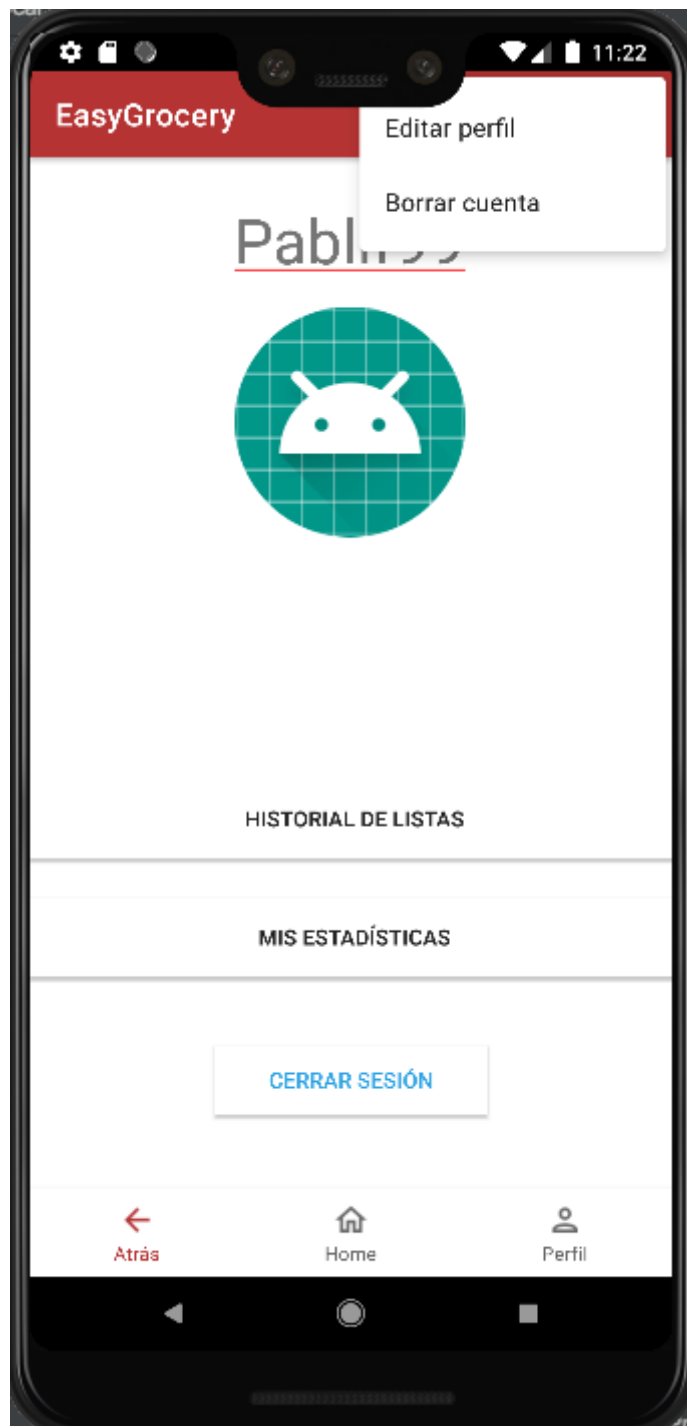


Figura 10.19: Editar perfil del usuario

-Borrar cuenta: Sirve para eliminar la cuenta del usuario. Al pulsar salta una notificación emergente como en la Figura 10.20, para que el usuario confirme la acción.



Figura 10.20: Borrar cuenta

-Editar perfil: Esta opción lleva al usuario a otra pantalla donde puede ver la información de su perfil como en la Figura 10.21. En esta pantalla, puede editar tanto su nombre como el nombre de usuario, pero no el gmail. Cuando el usuario ha acabado de editar sus datos, pulsa en modificar y actualizar la base de datos con los nuevos datos introducidos.



Figura 10.21: Editar perfil del usuario

En el centro de la pantalla (Figura 10.19), el usuario tiene dos botones bien diferenciados:

-Histórico de listas: Al pulsar en este botón el usuario llega a una pantalla donde puede ver todas las listas que ha comprado, aplicando un filtro de fecha en función de la precisión que este quiera, como se puede ver en la Figura 10.22.

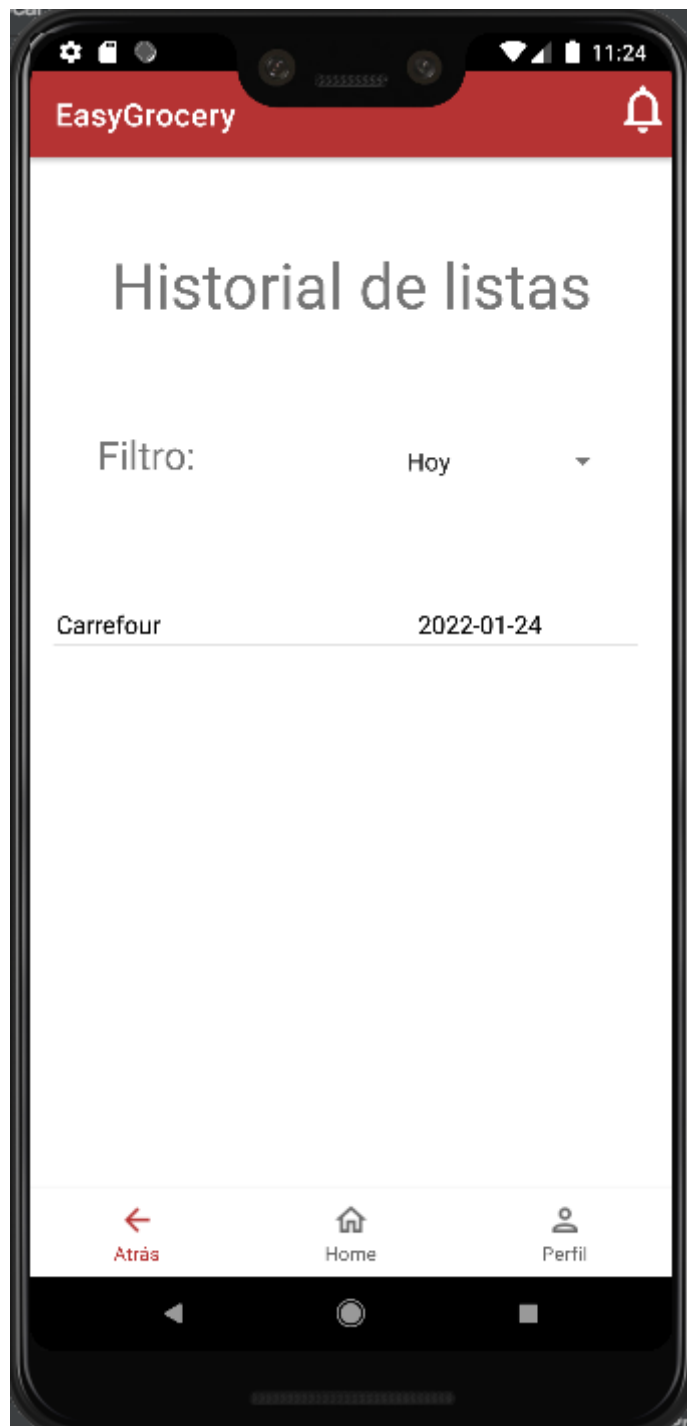


Figura 10.22: Historial de listas

Al pulsar en la lista que quiera, el usuario accede a otra pantalla donde puede verla con formato "ticket". En esta misma pantalla, arriba a la derecha (Figura 10.23) dispone de un botón para copiar listas del histórico a sus listas.



Figura 10.23: Dentro de una lista del historial

Si lo pulsa, le saldrá un mensaje informando de que se ha copiado con éxito la lista . Esta la podrá ver más adelante volviendo a la pantalla de home.

-Estadísticas: Al pulsar en este botón, el usuario accede a otra pantalla donde puede ver dos gráficos, uno del gasto mensual que ha hecho, y otro de la cantidad de productos comprados de cada categoría (Figura 10.24).

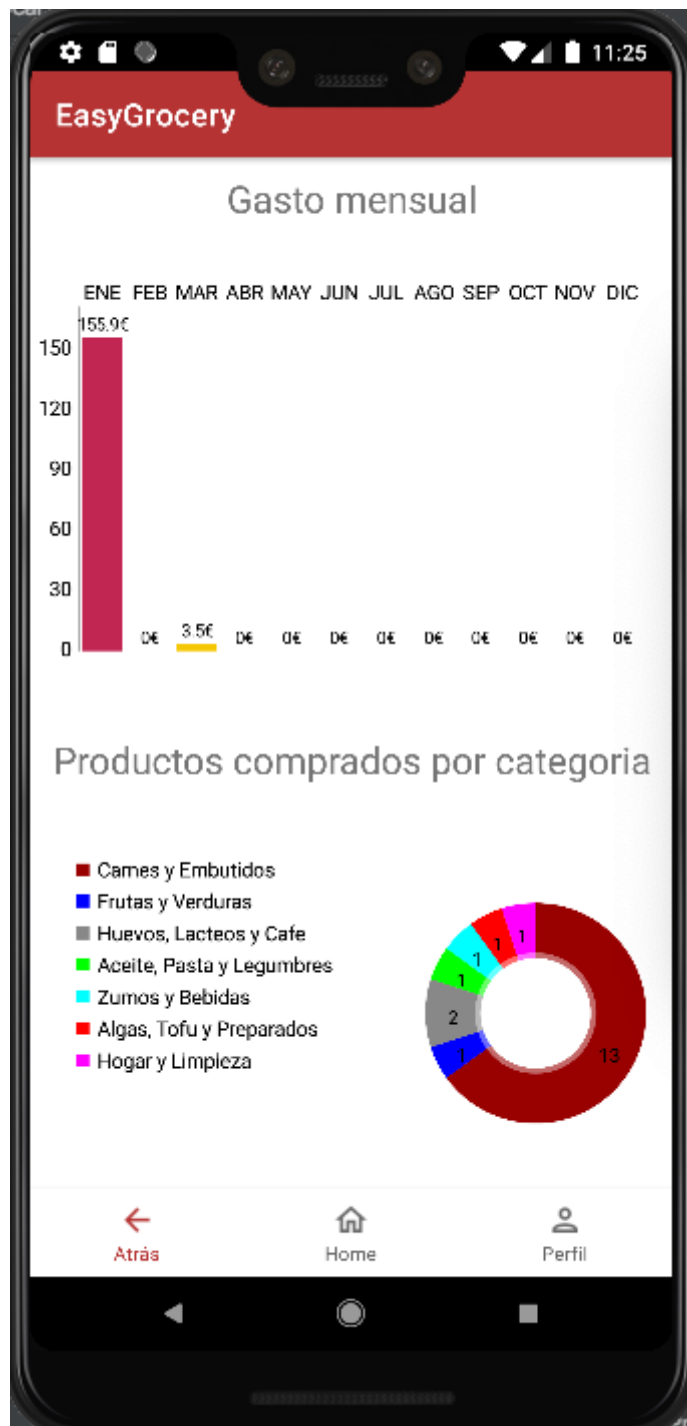


Figura 10.24: Estadísticas

El usuario además, dispone de un botón de notificaciones situado en la ToolBar en la parte superior derecha, representado por el icono de una campana (Figura 10.25).

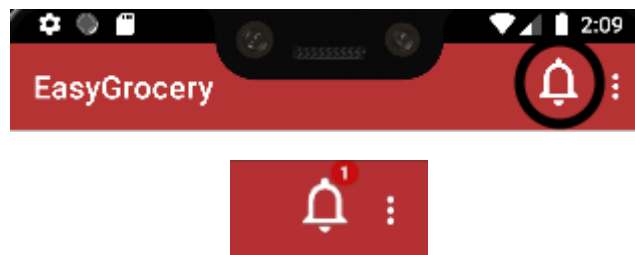


Figura 10.25: Notificaciones de invitación a una lista

Este botón lo tiene en la gran mayoría de las pantallas para que pueda acceder a las notificaciones con facilidad. Al pulsarlo le lleva a una pantalla donde puede ver las invitaciones que tiene, Figura 10.25, o un mensaje indicando que no tiene invitaciones si no tiene ninguna, Figura 10.26.



Figura 10.26: Listas a las que te han invitado



Figura 10.27: No tienes invitaciones a otras listas

Si tiene invitaciones y hace click en alguna de ellas, aparecerá una ventana emergente, igual que la de la Figura 10.26. El usuario puede elegir si aceptar o rechazar la invitación.



Figura 10.28: PopUp para aceptar o rechazar invitaciones a otras listas

-Si acepta: se añade la lista a sus listas y podrá acceder a ella desde la pantalla de home.

-Si rechaza: La invitación será eliminada y el usuario no formará parte de dicha lista



## Bibliografía

- [1] Android. Add the app bar. Disponible en <https://developer.android.com/training/appbar>
- [2] Android. BottomNavigationBar. Disponible en <https://developer.android.com/reference/com/google/android/material/bottomnavigation/BottomNavigationView>
- [3] Android. ListView. Disponible en <https://developer.android.com/reference/android/widget/ListView>
- [4] Android. Guía de arquitectura de apps | Desarrolladores de Android. 2019. Disponible en <https://developer.android.com/jetpack/guide?hl=es-419>
- [5] Google. Download Android Studio and SDK tools. 2013. Disponible en <https://developer.android.com/studio>
- [6] FIREBASE. Autenticación de usuarios. Disponible en <https://firebase.google.com/docs/auth/android/password-auth#kotlin+ktx>
- [7] BBDD. Bases de datos jerárquicas. Disponible en <https://ayudaleyprotecciondatos.es/bases-de-datos/jerarquicas/>
- [8] Ruben Chicharro. Bring, lista de compras. Disponible en <https://www.tuexpertoapps.com/2019/02/01/bring-10-claves-de-esta-aplicacion-para-hacer-la-lista-de-la-compra/>
- [9] Google. Google Keep. Disponible en <https://keep.google.com/u/0/>
- [10] Listonic. Listonic. Disponible en <https://listonic.com/es/>
- [11] Microsoft. One note. Disponible en <https://www.microsoft.com/es-es/microsoft-365/onenote/digital-note-taking-app>
- [12] Out of Milk. Disponible en <https://www.outofmilk.com/>

- [13] Firebase. Estructura tu base de datos. Disponible en <https://firebase.google.com/docs/database/web/structure-data?hl=es>
- [14] Android. Aprende Kotlin. Disponible en [https://developer.android.com/kotlin/learn?gclid=Cj0KCQiAubmPBhCyARIsAJWNpiP5WnazGLeyhj0tnmqjFtwRlGbrs1zZGsVR5xUSsoNb69KJb-iL1VUaAk48EALw\\_wcB](https://developer.android.com/kotlin/learn?gclid=Cj0KCQiAubmPBhCyARIsAJWNpiP5WnazGLeyhj0tnmqjFtwRlGbrs1zZGsVR5xUSsoNb69KJb-iL1VUaAk48EALw_wcB)
- [15] Android. Activities. Disponible en <https://developer.android.com/guide/components/activities/intro-activities>
- [16] EngineersDream. LinearLayout. Disponible en [https://www.youtube.com/watch?v=Z5mXiFW0Iws&ab\\_channel=EngineersDream](https://www.youtube.com/watch?v=Z5mXiFW0Iws&ab_channel=EngineersDream)
- [17] Android. Desarrollo de menús. Disponible en <https://developer.android.com/guide/topics/ui/menus?hl=es-419>
- [18] Leomaris Reyes. MVVM. Disponible en <https://medium.com/@reyes.leomaris/aplicando-el-patr%C3%B3n-de-dise%C3%B1o-mvvm-d4156e51bbe5>
- [19] Google. Firebase. Disponible en <https://firebase.google.com/>
- [20] W3School. XML. Disponible en [https://www.w3schools.com/XML/xml\\_what\\_is.asp](https://www.w3schools.com/XML/xml_what_is.asp)
- [21] Google. Google Vision API. Disponible en <https://cloud.google.com/vision/>
- [22] Android. MainActivity. Disponible en <https://developer.android.com/codelabs/kotlin-android-training-app-anatomy#2>
- [23] Android. CustomAdapter. Disponible en <https://www.tutorialspoint.com/how-to-write-a-custom-adapter-for-my-list-view-on-android-using-kotlin>
- [24] Android. DataModel. Disponible en <https://www.javatpoint.com/kotlin-data-class>

[25] Google. Google Drive. Disponible en  
<https://www.google.es/drive/>

[26] Firebase. Enlace a la BBDD del proyecto. Disponible en:  
<https://console.firebase.google.com/u/0/project/tfgjsp/database/tfgjsp-default-rtdb/data>

