



# MÓDULO PROYECTO

---

CFGS Desarrollo de Aplicaciones  
Multiplataforma  
Informática y Comunicaciones

---

**<DeliVibes>**

**Tutor individual:** <Cristina Silván Pardo>

**Tutor colectivo:** <Cristina Silván Pardo>

**Año:** <2024>

**Fecha de presentación:** (27/5/2024)

**Nombre y Apellidos:** Iván Martín Gómez  
**Email:** ivan.margom@educa.jcyl.es

**Foto actual  
del alumno**

## Contenido

1. IDENTIFICACIÓN DEL PROYECTO .....	4
2. ORGANIZACIÓN DE LA MEMORIA.....	4
3. DESCRIPCIÓN GENERAL DEL PROYECTO .....	5
3.1. OBJETIVOS .....	5
3.2. CUESTIONES METODOLÓGICAS .....	5
3.3. ENTORNO DE TRABAJO .....	5
4. DESCRIPCIÓN GENERAL DEL PRODUCTO .....	6
4.1. VISIÓN GENERAL DEL SISTEMA .....	6
4.2. DESCRIPCIÓN BREVE DE MÉTODOS, TÉCNICAS O ARQUITECTURAS.....	7
4.3. DESPLIEGUE DE LA APLICACIÓN .....	7
4.3.1 PLATAFORMA TECNOLÓGICA .....	7
4.3.2 INSTALACIÓN DE LA APLICACIÓN Y PUESTA EN MARCHA .....	7
5. PLANIFICACIÓN Y PRESUPUESTO .....	8
6. DOCUMENTACIÓN TÉCNICA .....	9
6.1 ESPECIFICACIÓN DE LOS REQUISITOS.....	9
6.2 ANÁLISIS DEL SISTEMA.....	10
6.3 DISEÑO DEL SISTEMA.....	11
6.3.1 DISEÑO DE LA BASE DE DATOS .....	11
6.3.2 DISEÑO DE LA INTERFAZ DE USUARIO .....	14
6.3.3 DISEÑO DE LA APLICACIÓN .....	20
6.4 IMPLEMENTACIÓN.....	20
6.4.1 ENTORNO DE DESARROLLO .....	20
6.4.2 ESTRUCTURA DEL CÓDIGO.....	20
6.4.3 CUESTIONES DE DISEÑO E IMPLEMENTACIÓN RESEÑABLES.....	23
6.5 PRUEBAS .....	24
7. MANUALES DE USUARIO .....	27

7.1 MANUAL DE USUARIO .....	27
7.1.1 PANTALLA DE CARGA .....	27
7.1.2 PANTALLA DE INICIO DE SESIÓN .....	28
7.1.3 PANTALLA DE REGISTRO .....	29
7.1.4 PANTALLA PRINCIPAL/INICIO .....	30
7.1.5 PANTALLA DE NUESTRO MENÚ .....	31
7.1.6 PANTALLA DE RESERVAS .....	32
7.1.7 PANTALLA DE NUESTROS PRODUCTOS .....	35
7.1.8 PANTALLA DE COMPRA .....	36
7.1.9 PANTALLA DE PROMOCIONES .....	39
7.1.10 PANTALLA DE TÉRMINOS Y CONDICIONES .....	40
7.1.11 PANTALLA DE CUENTA .....	41
7.2 MANUAL DE INSTALACIÓN .....	42
8. CONCLUSIONES Y POSIBLES AMPLIACIONES .....	42
8.1 CONCLUSIONES .....	42
8.2 POSIBLES AMPLIACIONES .....	42
9. BIBLIOGRAFÍA .....	43
10. ANEXOS .....	43

# 1. IDENTIFICACIÓN DEL PROYECTO

Portada de la memoria

## 2. ORGANIZACIÓN DE LA MEMORIA

La memoria consistirá en los siguientes apartados:

1. Identificación del proyecto: Es la propia portada de la memoria.
2. Organización de la memoria: Explicación de los apartados que consta la memoria.
3. Descripción general del proyecto: Se divide a su vez en tres apartados:
  - 3.1 Objetivos: Descripción de lo que se ha pretendido alcanzar en el proyecto.
  - 3.2 Cuestiones metodológicas: Metodología de desarrollo que se ha seguido.
  - 3.3 Entorno de trabajo: Tecnologías de desarrollo y herramientas utilizadas.
4. Descripción general del producto: Se divide a su vez en tres apartados:
  - 4.1 Visión general del sistema: Límites del sistema, funcionalidades básicas...
  - 4.2 Descripción breve de métodos, técnicas o arquitecturas.
  - 4.3 Despliegue de la aplicación: indicar plataforma tecnológica, instalación y puesta en marcha.
5. Planificación y presupuesto de la aplicación realizada
6. Documentación técnica, análisis, diseño, implementación y pruebas: Se divide en los siguientes apartados:
  - 6.1 Especificación de requisitos: Requisitos mínimos que tiene la aplicación.
  - 6.2 Análisis del sistema: Analizar el sistema de la aplicación.
  - 6.3 Diseño del sistema: Se divide en los siguientes apartados:
    - 6.3.1 Diseño de la base de datos: Explicación del modelo de la base de datos utilizada.
    - 6.3.2 Diseño de la interfaz de usuario: Diagrama de pantallas y su flujo.
    - 6.3.3 Diseño de la aplicación: Enumeración de todos los procesos de la aplicación.
  - 6.4 Implementación: Se divide en los siguientes apartados:
    - 6.4.1 Entorno de desarrollo: Herramientas utilizadas para el desarrollo de la aplicación.
    - 6.4.2 Estructura del código: Librerías, código con POO o a eventos...
    - 6.4.3 Cuestiones de diseño e implementación reseñables: Código que merezca la pena resaltar.
  - 6.5 Pruebas: Realizar y comentar las pruebas de la aplicación.
7. Manuales de usuario: Se divide a su vez en dos apartados:
  - 7.1 Manual de usuario: Manual que describe la interfaz de la aplicación y su flujo entre las pantallas.
  - 7.2 Manual de instalación: Manual en el que se resaltan los requisitos y pasos para la instalación de la aplicación.
8. Conclusiones y posibles ampliaciones.
9. Bibliografía.
10. Anexos.

Cabe destacar, que existen apartados con subapartados a su vez que no se han incluido por la extensión del apartado y por la repetición en el índice. Por lo que se han explicado los más importantes.

### 3. DESCRIPCIÓN GENERAL DEL PROYECTO

#### 3.1. *OBJETIVOS*

Los objetivos que se pretendían alcanzar al realizar esta aplicación era la composición de una aplicación móvil funcional para la gestión de una cadena de restaurantes y sus tiendas físicas. Además, suponía un gran esfuerzo realizar la aplicación con una tecnología nueva desconocida hasta la fecha, como puede ser el lenguaje de programación y la base de datos que se han utilizado en la realización de la app, por lo que sería otro objetivo.

Por lo cual, al finalizar la aplicación, los objetivos se han cumplido.

#### 3.2. *CUESTIONES METODOLÓGICAS*

La realización del proyecto consta de unas ideas, las cuales tenían su propia posición dentro del orden de ser programadas, por lo que la mejor metodología para implementar todas las ideas ordenadas era la metodología en cascada, la cual es muy útil ya que sigue una estructura rígida y estructurada, perfecta para ir desarrollando paso por paso las ideas propuestas. Además, permite volver a pasos anteriores si se ha encontrado algún error y arreglarlo.

Las pruebas tampoco han sido un problema, ya que cada vez que se programaba alguna funcionalidad nueva, permitía probarla para corroborar que funcione, permitiendo así que sea más dinámico el flujo implementación-pruebas.

#### 3.3. *ENTORNO DE TRABAJO*

La aplicación se define como un modelo cliente-servidor, es decir, una persona accede a diferentes pantallas en la app que se conectan con servidores externos, por lo que es necesario un entorno de desarrollo adecuado y capacitado para que al realizar nuestra aplicación nos permite conectarnos a servidores externos.

Es por eso, que el IDE de desarrollo elegido es Android Studio, el cual facilita mediante sus archivos de configuración la inserción de dependencias, plugins y funcionalidades necesarias para la implementación de conexiones con los servidores en la aplicación. Además, permite mostrar todo lo que se programe en un emulador virtual del propio IDE.

Android Studio presenta muchas ventajas, en las cuales también recoge a los lenguajes de programación con los que podemos trabajar en sus servicios, por lo que uno de ellos es con el que se ha desarrollado la aplicación, Kotlin.

Kotlin es un lenguaje de programación de código abierto. JetBrains lanzó Kotlin bajo la licencia Apache 2.0, lo que permite su libre uso, modificación y distribución, tanto para proyectos comerciales como no comerciales.

La aplicación consta de funcionalidades diversas, las cuales más adelante se hace referencia a ellas, muchas de las cuales son mediante una base de datos y una API.

En cuanto a la base de datos, la elegida es Firebase, proporcionada por Google, proporciona una amplia gama de servicios en la nube que ayudan a los desarrolladores a construir, mejorar y escalar aplicaciones de manera más rápida y eficiente.

Muestra una serie de ventajas como la autenticación, el almacenamiento en la nube, datos en tiempo real (lo que hace que no tengamos que volver a lanzar nuestro programa para ver los cambios que hemos realizado en la base de datos) y analytics (proporciona información detallada sobre el rendimiento y el comportamiento de los usuarios de la aplicación).

Como se ha dicho previamente, también es utilizada una API, para realizar operaciones con ella, como la lectura de datos. En este proyecto se ha usado la API Open Food Facts, la cual guarda los productos de los supermercados, así como su información nutricional, código de barras, imágenes... todo con un identificador único.

## 4. DESCRIPCIÓN GENERAL DEL PRODUCTO

### 4.1. *VISIÓN GENERAL DEL SISTEMA*

DeliVibes es una aplicación por y para sus restaurantes y tiendas, por lo que se define como una aplicación autónoma, aunque requiera de servicios externos para su correcto funcionamiento.

Es por eso, que sus funcionalidades son exclusivamente nativas, de las cuales se detallan a continuación:

- Gestión de usuarios. La aplicación permite la gestión de usuarios gracias a la base de datos que tiene integrada, en la cual se comprueba si el usuario que intenta acceder está registrado o no. Esta funcionalidad se lleva a cabo mediante un inicio de sesión o un registro.
- Reserva: Se permite la reserva de una mesa, teniendo que rellenar los campos necesarios para efectuar la misma.
- Validación de la reserva: Con los datos de la reserva, se hace una llamada a la base de datos para comprobar que la reserva se puede realizar o hay ya una reserva hecha con los datos proporcionados.
- Visualización de información de los productos: Permite al usuario consultar la información nutricional de cada uno de los productos existentes en tienda, mediante la conexión a una API.
- Gestión de la cuenta: El usuario puede consultar sus propios datos privados, además de poder eliminar su cuenta si lo requiere.
- Vista del menú actual: Existe la posibilidad de que el usuario pueda ver el menú que dispone el restaurante.
- Compra: La compra de un producto es posible, mediante la selección del producto deseado y la inserción de campos obligatorios que se deben rellenar para la efectución de la misma.
- Términos y condiciones: Los términos y condiciones del restaurante DeliVibes se podrán leer desde la aplicación.
- Promociones: El restaurante brinda varias promociones para celebrarse, las cuales el usuario puede consultar en la app.

Existen más funcionalidades en la aplicación, las cuales se explicarán en el manual de usuario.

En la aplicación sólo existe un tipo de usuario, el cliente, es decir, toda persona que quiera usar los servicios de DeliVibes, tendrá las mismas funcionalidades disponibles que el resto, no habiendo así un tipo de usuario con funcionalidades especiales o diferentes al resto. No obstante, la aplicación sólo está disponible en el sistema operativo Android, por lo que las personas que dispongan de un sistema operativo diferente no podrán usar DeliVibes.

## **4.2. DESCRIPCIÓN BREVE DE MÉTODOS, TÉCNICAS O ARQUITECTURAS**

Las especificaciones dichas en el título de este apartado ya han sido nombradas y explicadas en puntos anteriores, los cuales son los siguientes:

- Descripción breve de métodos y técnicas: Apartado 4.1
- Arquitecturas: Apartado 3.2

## **4.3. DESPLIEGUE DE LA APLICACIÓN**

### **4.3.1 PLATAFORMA TECNOLÓGICA**

La aplicación se ha desarrollado en Android Studio, como previamente se ha mencionado, la cual se ha utilizado el lenguaje Kotlin para el desarrollo de la misma.

No obstante, existen diferentes versiones de los lenguajes en el IDE, por lo que la elección de una versión u otra es de suma importancia a la hora de saber qué usuarios pueden tener dicha aplicación instalada en sus dispositivos móviles.

La versión de API usada en este proyecto es la 34, es decir, que la aplicación funciona sin problemas para el 96,3 % de los dispositivos móviles en los que esta se instale, todo ello gracias a que utiliza la versión de Android "7.0".

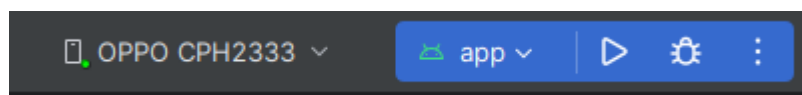
Por último, la versión que se ha utilizado de Kotlin es la 1.9.

Dado que la aplicación comprende al 96,3% de los dispositivos móviles, no hay unos requisitos de despliegue específicos, sólo que tengan la versión de Android necesaria, en este caso, la 34.

### **4.3.2 INSTALACIÓN DE LA APLICACIÓN Y PUESTA EN MARCHA**

Hasta la fecha, la aplicación no se encuentra en plataformas de instalación, como puede ser Microsoft Store, Google Play o Apple Store, por lo que la única forma de instalarlo es seguir estos pasos:

1. Descarga de Android Studio: IDE necesario para poder abrir la aplicación que vamos a instalar en nuestro dispositivo móvil.
2. Conectar nuestro dispositivo móvil con un cable USB a nuestro ordenador para que nos reconozca el *device*.



3. Activar las opciones de desarrollador de nuestro dispositivo móvil:
  - 3.1. Entrar en configuración->Información del dispositivo->Versión, pulsar tantas veces como sea necesario en el número de versión hasta que nos salga la opción de meter nuestra contraseña.
  - 3.2. Una vez estén activadas las opciones de desarrollador, entrar en configuración->Ajustes adicionales->Opciones para desarrolladores, y activar las siguientes configuraciones:
    - Depuración por USB
    - Verificar aplicaciones por USB
4. A continuación, en nuestro dispositivo móvil, seleccionar la opción de Transferencia de archivos/Android Auto (si no se ha hecho nada más conectar el cable al ordenador), y con el Android Studio abierto, seleccionar la opción “Aceptar” que nos aparece en nuestro dispositivo móvil.
5. Por último, pulsar la opción de play en el Android Studio:



## 5. PLANIFICACIÓN Y PRESUPUESTO

- Planificación: La planificación en el proyecto es de suma importancia, dado que se tiene que definir con antelación las fases en las que va a consistir el proyecto. Dicha planificación se ha organizado de la siguiente manera:
  - 0-15 horas:

En las primeras 15 horas de trabajo, los avances realizados han sido estructurar el proyecto y saber qué disposición de pantallas realizar acorde a la idea inicial, ajustándose al tiempo de desarrollo establecido.

Durante este tiempo, la aplicación fue cogiendo forma, ya que se desarrollaron las pantallas de inicio, login y registro.
  - 15-30 horas:

En las siguientes 15 horas se realizaron avances en la pestaña home (la pestaña que aparece una vez iniciado sesión/registrado en la aplicación) y retoques de diseño en las páginas de login y registro.
  - 30-50 horas:

Durante las 20 horas siguientes, se realizaron todas las pestañas de la aplicación, las cuales muchas de ellas incluían lógica, además, se desarrolló la estructura común de un menú desplegable, que después de varias horas intentando su correcto funcionamiento, al fin se pudo implementar correctamente.
  - 50-60 horas:

La aplicación se completa, se acaba de desarrollar la funcionalidad de las pantallas que no tenían lógica y se remata el diseño de la aplicación, incluyendo todas las peculiaridades de cada una de las ventanas.
  - 60-80 horas:

Retoques de última hora, cambio en el diseño de algunas pantallas, y creación de nueva funcionalidad para complementar la lógica ya establecida.

La aplicación se comprueba para que no haya ningún error y se da por acabada.



- Presupuesto:

Dado que la realización del proyecto ha sido en horario no laboral, no se ha obtenido ninguna remuneración por el mismo, por lo que el precio a pagar por el desarrollo de la aplicación si hubiese sido en horario laboral sería el siguiente:

- Un programador de media en España cobra 15€ la hora.
- La realización del proyecto ha constado de 80 horas.
- El precio de la aplicación serían 1200€.

No obstante, el desarrollo del proyecto ha requerido aplicaciones y herramientas las cuales se necesitan usar para su correcto funcionamiento. Dichas no han supuesto un desembolso económico, por lo que todo el software ha salido gratis.

En cambio, la aplicación ha requerido un hardware que no se disponía al inicio, por lo que se ha hecho un desembolso total de 750€ en diferentes componentes de este.

Por último, pero no menos importante, se ha hecho un desembolso del hosting, ya que, aunque la aplicación se dependía de un servidor para lanzarse (pero sí para conectarse a la base de datos y la API), requería una gran memoria RAM y ROM para su rápido desarrollo, por lo cual se desembolsó dinero en el hardware.

## 6. DOCUMENTACIÓN TÉCNICA

### 6.1 ESPECIFICACIÓN DE LOS REQUISITOS

Para la realización de la aplicación, se ha seguido una serie de requisitos impuestos por quién me demanda el proyecto, aquí cito los cuales:

- Presentación formal:
  - Se ajusta a los requerimientos formales establecidos: Portada, índice estructurado, extensión (mínimo 40 páginas), buena redacción, sin faltas de ortografía.
  - Se incluyen conclusiones finales sobre el proyecto e investigación realizada, posibles ampliaciones y bibliografía/webgrafía.
  - Presenta el proyecto en la forma y plazo establecidos.
- Propuesta del proyecto:
  - Originalidad del tema elegido.
  - Grado de dificultad en orden a la investigación de contenidos.
  - Grado de actualidad del tema elegido (el tema está relacionado con tecnologías actuales y vanguardistas).
- Solución:
  - En el caso de una aplicación mobile/web: consume datos de un servicio externo (conexión BBDD o a través de API), implementación CRUD, indica modelo de la base de datos, autenticación de usuarios En caso de otro tipo de aplicación (IA, videojuego...) documenta convenientemente todo el proceso de gestión de datos.
  - La aplicación es totalmente funcional, se controlan errores, comentarios del código.
  - Incluye manual de usuario/Contenido multimedia.

- Durante la realización del proyecto, se realizan avances en la ejecución del proyecto (revisión del repositorio del proyecto cada 15 días para comprobar avances). En el README se indican las tareas realizadas (enlaces a tutoriales, esquemas de diseño...).
- El proyecto está subido a GitHub como público. El repositorio contiene el código fuente de la aplicación y un archivo README con el siguiente contenido: Título del proyecto, descripción, enlace a la memoria del proyecto en formato PDF, enlace al manual de usuario en formato PDF.
- Defensa oral:
  - Se ajusta al tiempo marcado, explica el funcionamiento de la aplicación y no se apoya en vídeos explicativos.
  - Administra el tiempo en relación a los contenidos del proyecto de forma homogénea.
  - Establece un esquema o guía para la exposición, y lo sigue.
  - Hace referencia al objetivo del trabajo, los puntos esenciales y las conclusiones finales.
  - Responde a las cuestiones planteadas por los miembros del Tribunal con autoridad y de forma razonada. Aspecto físico adecuado a la presentación del proyecto.
  - Grado de conocimiento y dominio de los contenidos expuestos y lenguaje técnico utilizado.

## 6.2 ANÁLISIS DEL SISTEMA

Apartado relacionado con el 4.1, en el que se describen algunas de las funcionalidades principales de la aplicación, por lo que dichas funcionalidades tienen sus propios desgloses que a continuación se van a especificar y explicar:

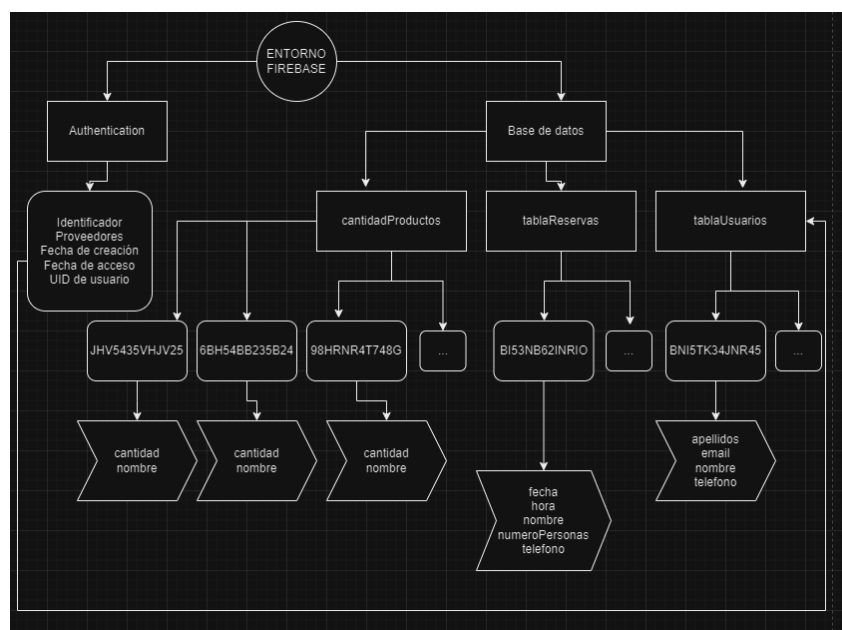
- Gestión de usuarios: Consta de unas pantallas de inicio de sesión y de registro, ambas con sus conexiones a la base de datos para comprobar dicho usuario:
  - Inicio de sesión: Mediante la inserción de los campos de correo y contraseña, la aplicación comprueba en la base de datos si esos campos existen o si son correctos, permitiendo así el inicio de sesión del usuario en caso de ser correcto.
  - Registro: El usuario rellena los campos de nombre, apellidos, teléfono, correo y contraseña, teniendo requisitos en la contraseña y en el correo que en apartados posteriores se explicarán. Si el registro es aceptado, se permite el acceso del usuario al resto de pantallas.
- Reserva: El usuario dispone de la elección de reservar, mediante la inserción previa de una hora, día, número de personas, nombre de la persona a reservar y un teléfono de contacto. Validación de la reserva: Si los datos de los campos son correctos a la hora de hacer la reserva, esta se permitirá hacer, en caso de datos erróneos o datos que pertenecen a otra reserva (día y hora) no se dejará reservar.
- Validación de la reserva: Al tener los datos insertados para realizar la reserva, se procederá a reservar, lo cual sucede al pulsar el botón "Reservar". A continuación, se accede a la base de datos, en la cual se comprueba si los datos que se han insertado en la reserva ya existen en otro registro, es decir, que otra persona ya haya reservado ese mismo día a la misma hora.

- Visualización de información de los productos: El usuario dispone de un botón en cada uno de los productos disponibles en la tienda, el cual si se pulsa muestra la información nutricional del producto asociado.
- Gestión de la cuenta: Los datos de la persona que ha accedido a través de un inicio de sesión o de un registro, tendrá la posibilidad de ver sus datos personales, además de dos botones, el botón de salir (el cual, cierra la sesión actual) y el botón de eliminar cuenta (permite al usuario eliminar su cuenta, por lo que se cierra la sesión actual y la cuenta se borra de la base de datos).
- Compra: La compra se efectuará mediante los siguientes pasos:
  - Primero, el usuario debe escribir en la barra de búsqueda el producto que desea.
  - Después, debe consultar la cantidad del producto.
  - Por último, debe rellenar los campos de cantidad, información adicional y dirección
  - Después, aparecerá un dialog, en el que debe insertar el número de tarjeta y el código postal.
- Términos y condiciones: DeliVibes tiene, como cualquier otro restaurante, unos términos y condiciones que debe seguir, por lo que desde la propia aplicación se pueden consultar.
- Promociones: El restaurante tiene tres promociones que ofrece al público, celebración de cumpleaños, comidas y cenas para bodas y festividades de Navidad y fin de año. La información de dichas promociones se puede leer desde la aplicación.

Cabe recalcar que, en este apartado se han descrito sólo las funcionalidades mencionadas en el apartado 4.1.







## 6.3 DISEÑO DEL SISTEMA

### 6.3.1 DISEÑO DE LA BASE DE DATOS



En la foto se muestra el modelo E/R de la estructura seguida de la base de datos, en la que se puede ver los diferentes flujos:

- Entorno de firebase: es todo aquello que conlleva usar Firebase en la aplicación. De todas las funcionalidades disponibles que nos brinda este servicio, se utilizan dos, Firebase Authentication, y la propia base de datos.
- Firebase Authentication: Se trata de un sistema que, como su propio nombre indica, revisa la autenticación de los usuarios. Para su manejo es necesario especificar el modo de registro de los usuarios, es decir, si es por email y contraseña, por autenticación mediante Google, Facebook...

Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
ivan3@gmail.com		13 abr 2024	5 may 2024	GDc7HYXRWUd9NRTgVZBDA...
ivan2@gmail.com		13 abr 2024	13 abr 2024	86AZbQgKFvOfMdcxeywa0Rx...
ivan2@gamil.com		13 abr 2024	13 abr 2024	PgTtOlK9G9dcAGnqQtawyyaVD...
ivan1@gmail.com		13 abr 2024	13 abr 2024	k43s6QJDN3gplG9eDEfvmT7...
prueba1@gmail.com		13 abr 2024	13 abr 2024	uKQtdFxivJbnrCgyqJIYfki7ff2
p@gmail.com		11 abr 2024	11 abr 2024	tHPXdgdvj40IAPpV6r8xuKtUT...

Al registrarse como nuevo usuario, se guardan campos con el correo, fecha de creación, fecha de acceso, uuid del usuario (no visible en la aplicación) y proveedores (la forma de autenticarse en la aplicación, en este caso, mediante correo).

Es un concepto rompedor e inusual, debido a que nos permite guardar los registros de los usuarios sin necesidad de crear una base de datos con tablas.

- Firestore Database (base de datos de Firebase): Base de datos no relacional, utilizada en la aplicación para el control de guardado de cada uno de sus diferentes datos en función de las funcionalidades.


Al no ser una base de datos relacional, su estructura no se desglosa en tablas->columnas->registros, si no que lo hace de la siguiente forma:

(default)	cantidadProductos	10aa4x9lwU2mlNcb9FiD
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
cantidadProductos >	10aa4x9lwU2mlNcb9FiD >	+ Agregar campo
tablaReservas	40i2rMiFaT3S416NIilc	cantidad: 20
tablaUsuarios	Cw5q7i6SXdVUZ0ojEIeN	nombre: "Mermelada de frambuesa"
	HW3FRJCxN9V0n4bjPyZk	
	Jgc4eHdbk0Y3xxYGZQe5	
	SVj3z8133g287MuqphyH	
	Wj6bkhGxr0Bn7YEou5gp	
	XGPG1mNo7Pn9rjoNaUYo	
	Xz76q3zfX10pr8ILE7XW	
	gTHRZqHIPr3jjzYFZhM7	
	xClD4VV9rc9LjQcPZZ8P	
	xiR4c4XVW6uQbB2ccGqX	

Como se observa en la imagen, la base de datos se divide en tres colecciones (las tablas de las bases de datos convencionales), por cada colección, existen varios documentos, los cuales actúan como un objeto en POO, el cual contiene el nombre del campo con su valor asociado. De esta manera, al ser una tabla que contiene objetos, la forma correcta para acceder a un dato en concreto es mediante un bucle, de esta manera hacemos una iteración que recorre todos y cada uno de los documentos hasta que se encuentre con el dato buscado; por lo que es una forma distinta de hacer consultas con respecto a las bases de datos relacionales.

Las imágenes tienen correlación entre ellas, la información que aparece en el diagrama E/R es la misma que la información que aparece en las imágenes de la base de datos, excepto un pequeño detalle:

Si uno se fija en el modelo E/R, verá que hay una línea que une tablaUsuarios con los campos del authenticator, y esto tiene un por qué. Los datos del authenticator sólo se utilizan para el momento en que una persona quiere acceder a la aplicación mediante un registro, por lo que si les queremos mostrar o actualizar, necesitaremos que estén en una base de datos, es por eso que se ha creado una colección llamada tablaUsuarios:

<div>  &gt; tablaUsuarios &gt; 86AZbQgKFvOf...         <div>Más funciones en Google Cloud</div> </div>		
(default)	tablaUsuarios	86AZbQgKFvOfMdcxeywa0RxeW82
<div>+ Iniciar colección</div> <div>cantidadProductos</div> <div>tablaReservas</div> <div>tablaUsuarios &gt;</div>	<div>+ Agregar documento</div> <div>86AZbQgKFv0fMdcxeywa0... &gt;</div> <div>GDc7HYXRWUd9NRTgVZBDA...</div> <div>Ivan1234</div> <div>LuVzARJ6VPcSW8kyw3KI</div> <div>MjSs1PA7p8PI1afQkcW1A...</div> <div>NuK7bRuuEUSw77MMHVWzt...</div> <div>ivanmartingomez2511@g...</div> <div>ivi</div> <div>ivi@gmail.com</div> <div>p@gmail.com</div> <div>prueba@gmail.com</div> <div>uKQtdFxivJbncGyqJIYf...</div>	<div>+ Iniciar colección</div> <div>+ Agregar campo</div> <div>apellidos: "martin gomez"</div> <div>email: "ivan2@gmail.com"</div> <div>nombre: "ivan"</div> <div>telefono: "98949495"</div>

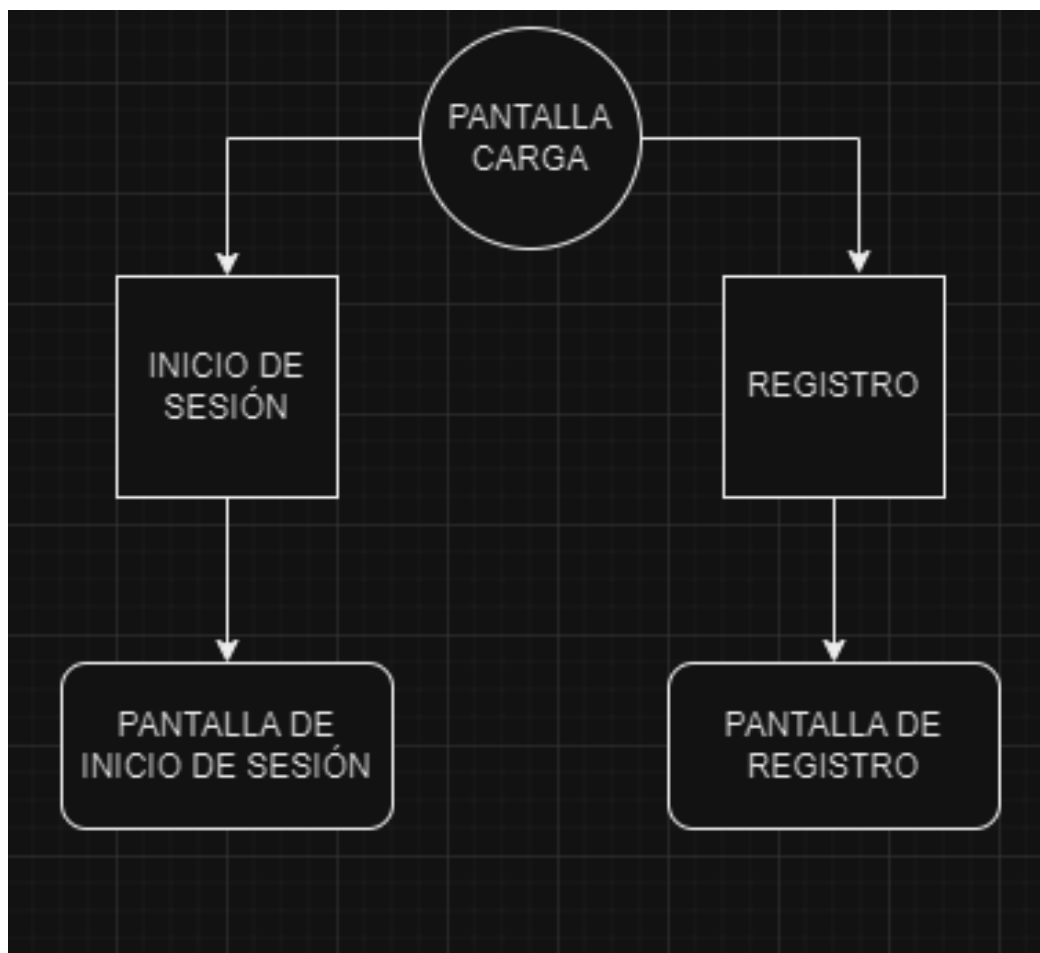
En dicha tabla, cada vez que se hace un registro nuevo, además de guardarse en un campo en el authenticator, también se crea un nuevo documento en la colección, de esta manera tenemos los datos del usuario repetido cada uno para su funcionalidad, es decir, los datos del usuario que están en el authenticator sólo se usan para el registro/inicio de sesión, mientras que los datos que están en la base de datos se utilizan para poder acceder a ellos y realizar operaciones con los mismos.

Los documentos se guardan con un id (Letras mayúsculas y minúsculas) para un mejor acceso.

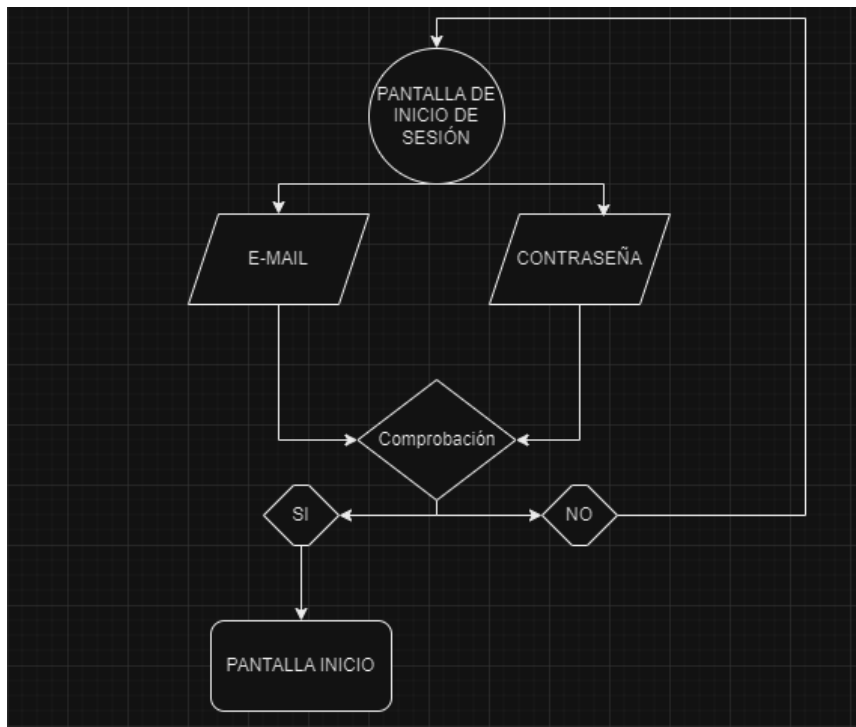
### 6.3.2 DISEÑO DE LA INTERFAZ DE USUARIO

A continuación, se muestran los diseños de la interfaz de cada una de las pantallas.

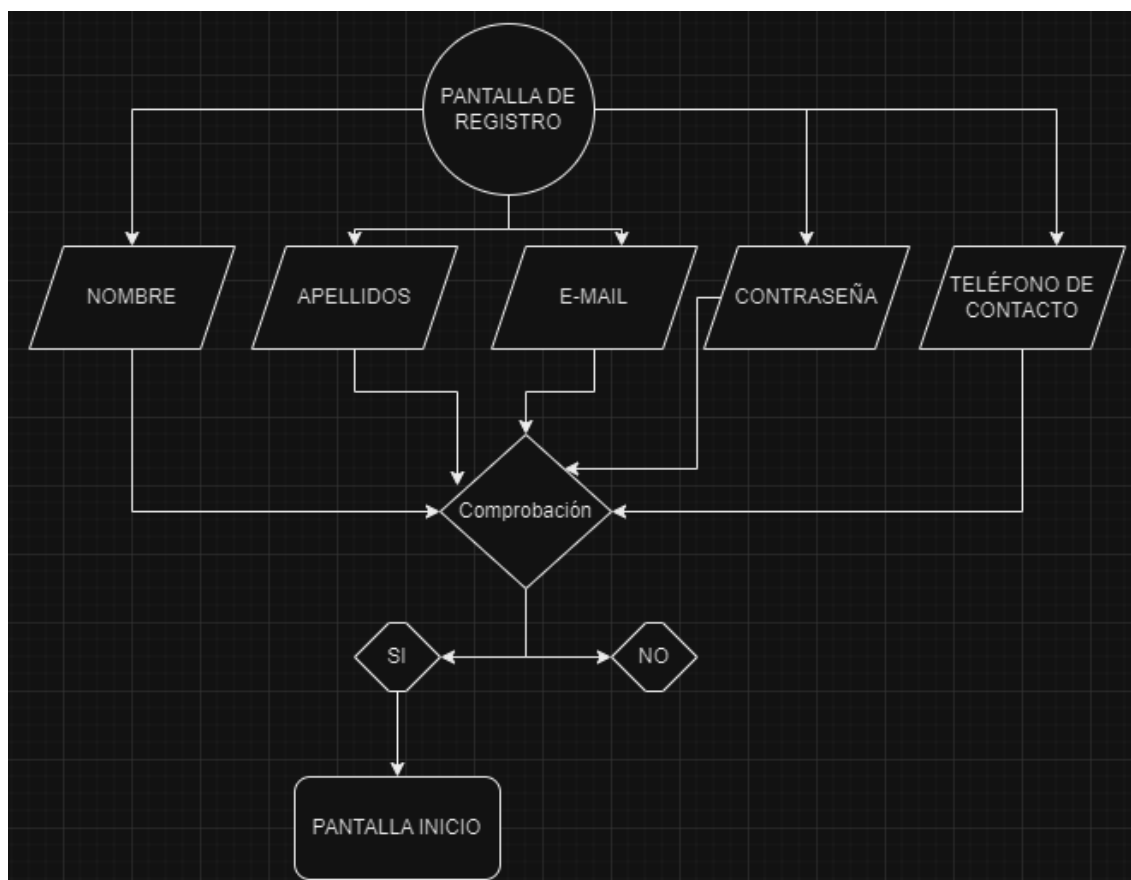
#### PANTALLA CARGA



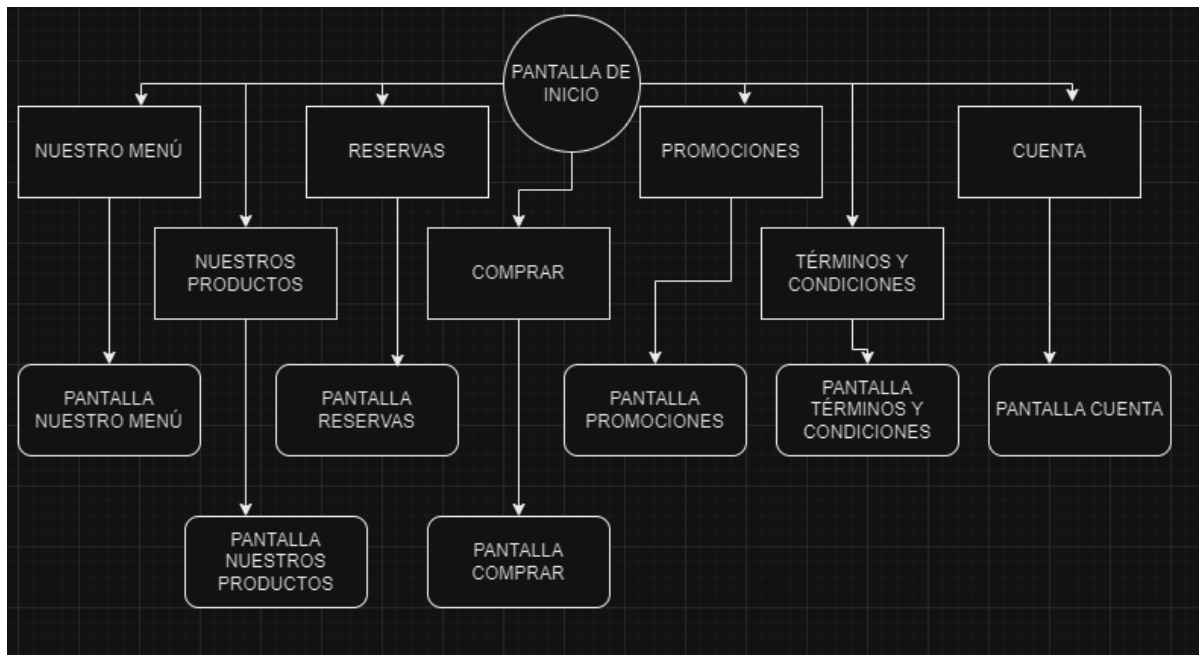
### PANTALLA INICIO SESIÓN



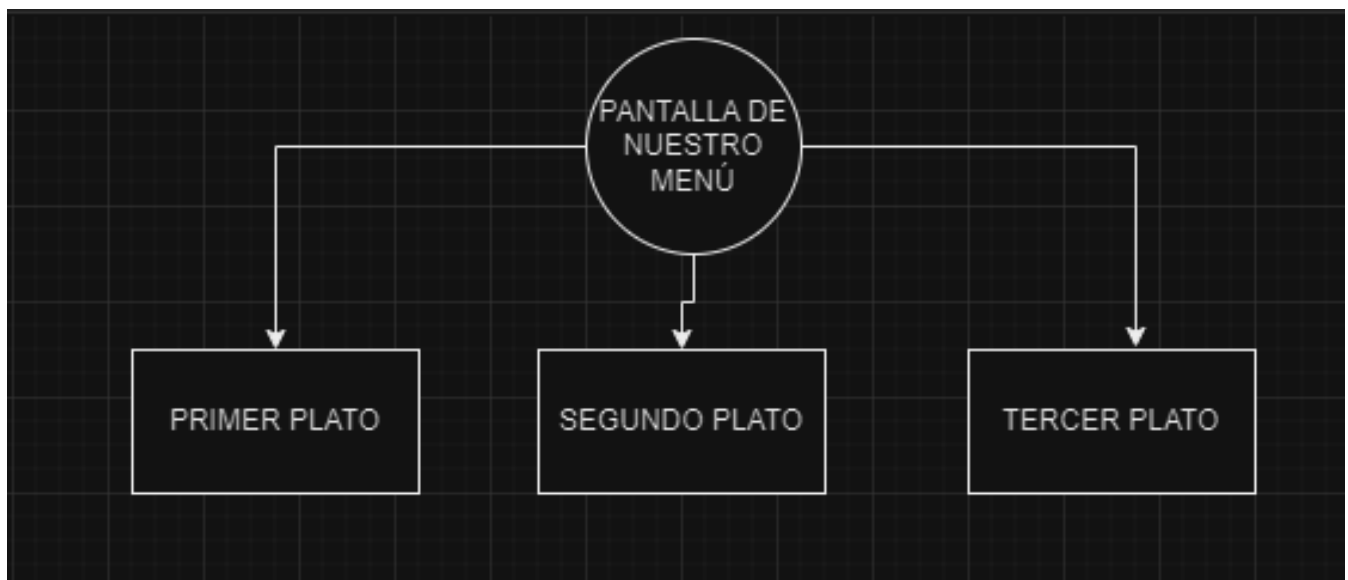
### PANTALLA DE REGISTRO



### PANTALLA DE INICIO

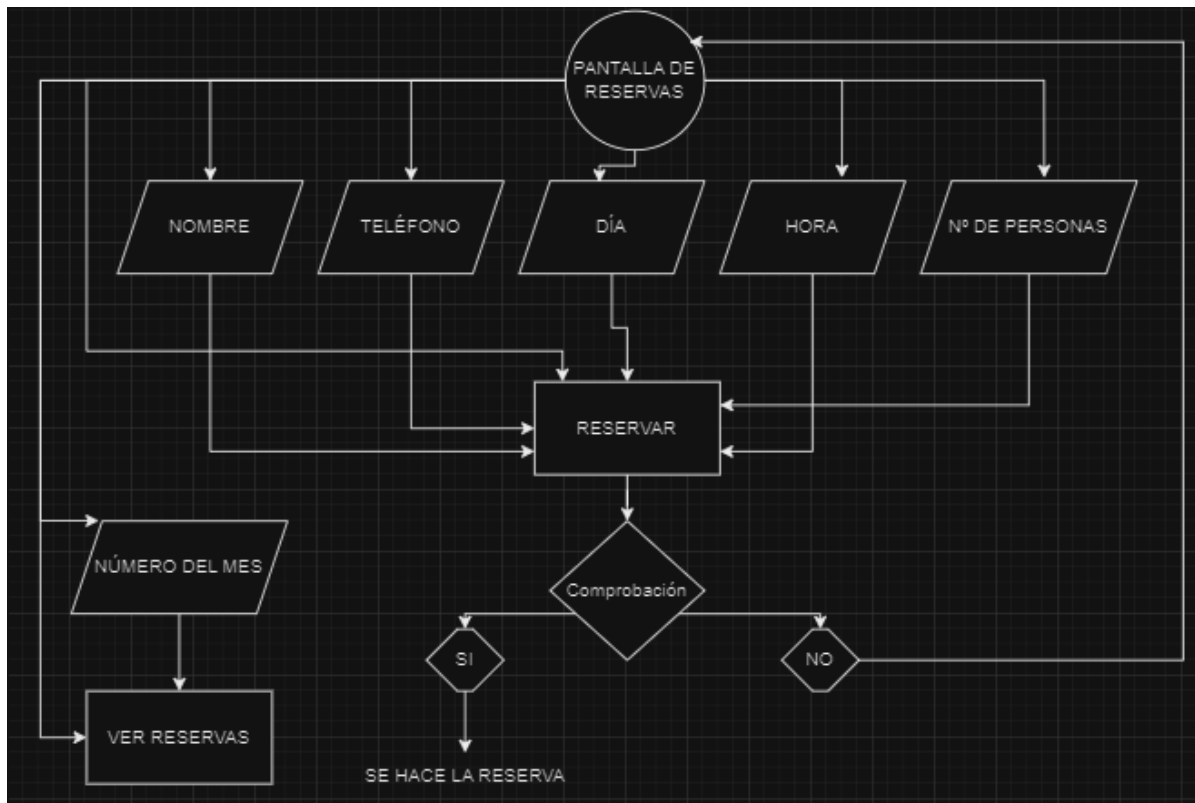


### PANTALLA NUESTRO MENÚ

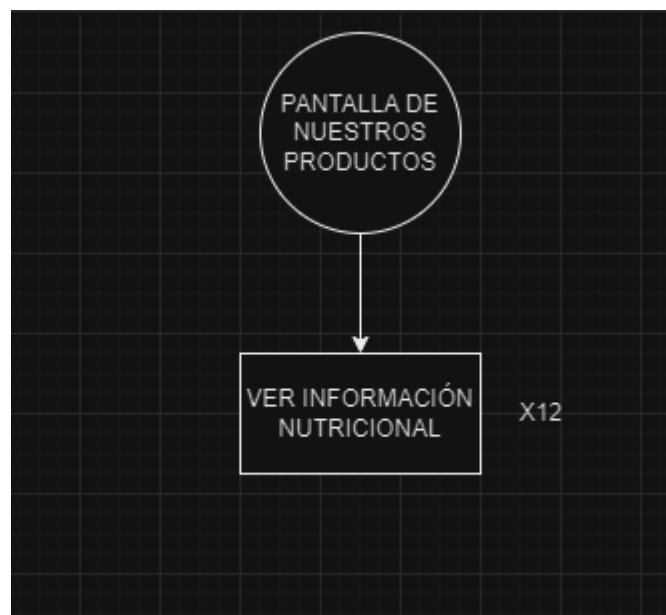




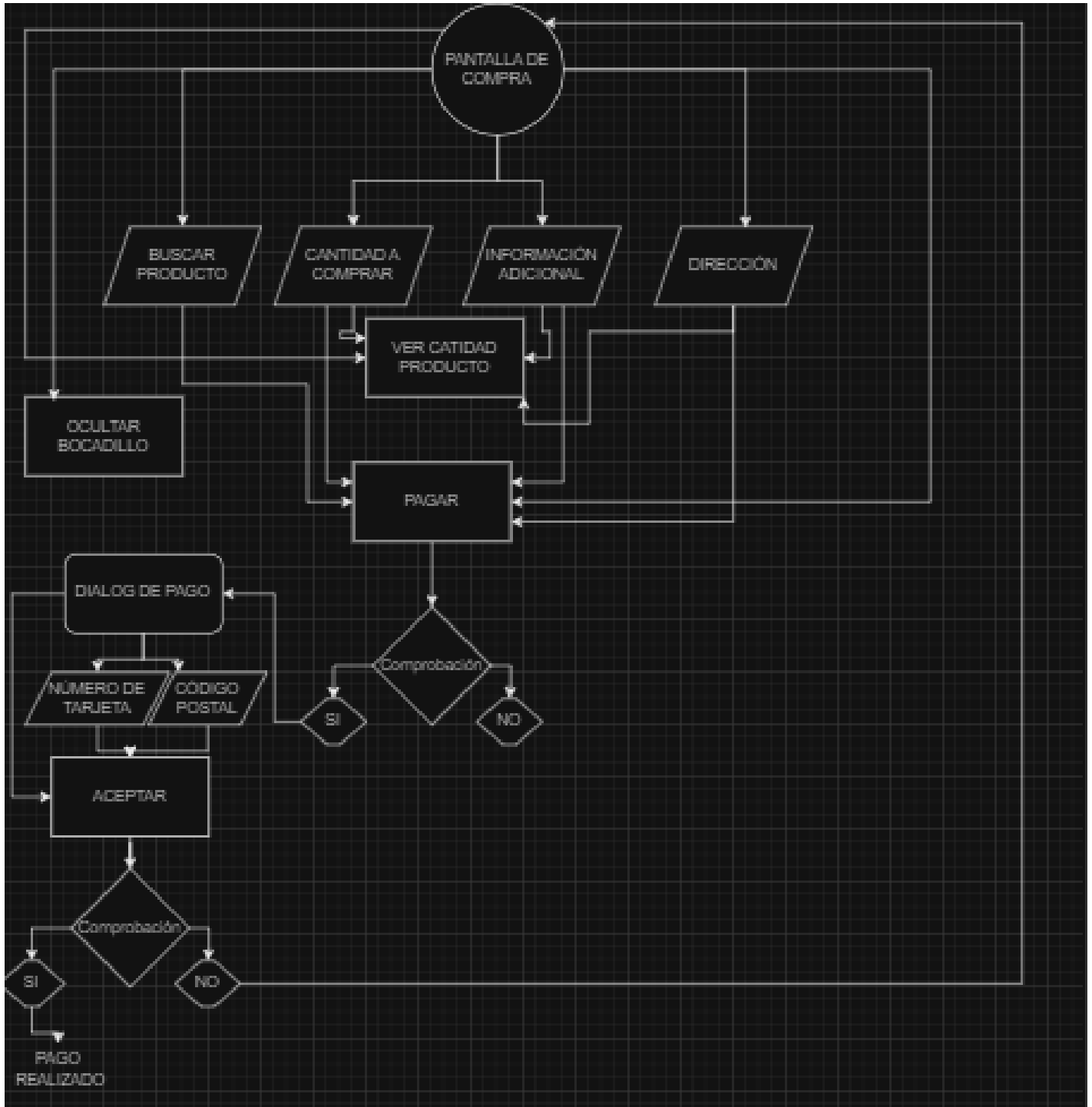
### PANTALLA RESERVAS



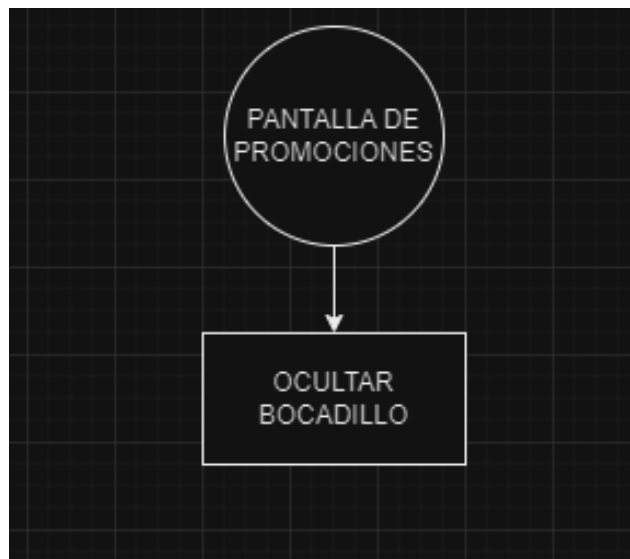
### PANTALLA NUESTROS PRODUCTOS



PANTALLA DE COMPRA



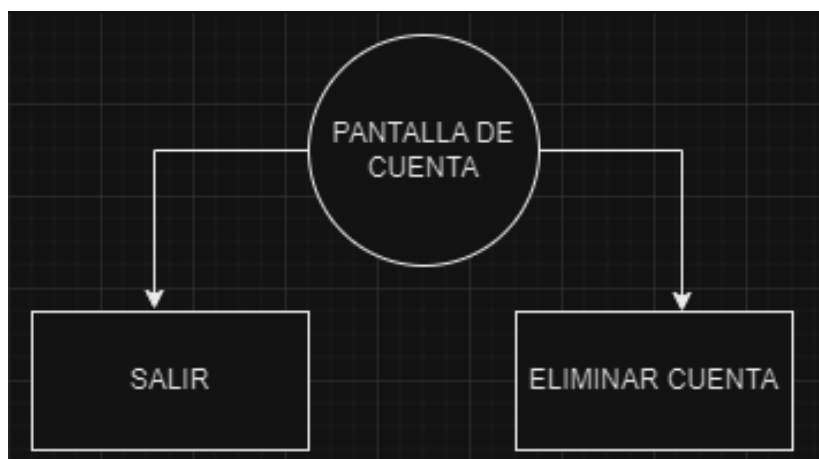
### PANTALLA DE PROMOCIONES



### PANTALLA DE TÉRMINOS Y CONDICIONES



### PANTALLA DE CUENTA



### 6.3.3 DISEÑO DE LA APLICACIÓN

La aplicación consta de una serie de procesos que, unidos entre sí, configuran el BackEnd, por lo que se debe explicar que funcionalidad tienen dichos procesos y cómo afecta cada uno de ellos al desarrollo de la aplicación.

Esas funcionalidades de cada proceso ya han sido citadas y explicadas de forma detallada en los apartados anteriores 4.1 y 6.2.

## 6.4 IMPLEMENTACIÓN

### 6.4.1 ENTORNO DE DESARROLLO

Botones, drawers, dialogs, son algunos de los elementos utilizados en el diseño de la aplicación. Cabe destacar, que también se han utilizado funcionalidades en el código que han hecho un mejor desarrollo de este.

Todas estas “estructuras”, se verán en el manual de usuario y en el propio código de la aplicación, con mucha más atención y detalle, aunque en este apartado se citarán algunas de ellas:

Dividiendo por front y back, distinguimos las siguientes estructuras.

- Estructuras utilizadas en el front:
  - Botones.
  - Drawer.
  - Imágenes.
  - Barra de búsqueda.
  - Campos a rellenar.
  - Dialogs.
  - Etc.
- Estructuras utilizadas en el back:
  - Listas.
  - Objetos.
  - Métodos.
  - Callbacks.
  - Conexiones a API.
  - Consultas a BD.
  - Etc.

### 6.4.2 ESTRUCTURA DEL CÓDIGO

Para el desarrollo de la aplicación, se ha necesitado de librerías/bibliotecas para utilizar funcionalidades o estructuras propias. En Android Studio, existen tres formas de implementar librerías:

- Mediante la carpeta libs: En la carpeta libs (nombre\_del\_proyecto/app/libs/), se pueden descargar archivos JAR o AAR, los cuales son ejecutables que contienen dentro los archivos de las librerías.

- Mediante la ventana External Libraries: A esta ventana se puede acceder pulsando en View->Tool Windows->External Libraries. En dicha ventana se pueden ver las librerías usadas en el proyecto, además de poder añadir nuevas. La vista de las librerías mediante esta ventana se puede realizar a través de dependencias Gradle u otros medios.
- Dependencias Gradle: Son las dependencias que pertenecen a las librerías, en vez de importar la librería completa se implementa la dependencia que se necesite usar. Se puede acceder al archivo donde se encuentran las dependencias mediante la ruta nombre\_del\_proyecto/app/build.gradle.

En la aplicación DeliVibes, el modelo escogido para implementar librerías es el build.gradle, por lo que se han ido implementado dependencias acorde con lo necesitado en el proyecto. El archivo build.gradle consta de las siguientes dependencias:

```
dependencies { this: DependencyHandlerScope

    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.11.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.7.0")
    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0")
    implementation("androidx.navigation:navigation-fragment-ktx:2.5.3")
    implementation("androidx.navigation:navigation-ui-ktx:2.5.3")
    implementation("androidx.drawerlayout:drawerlayout:1.2.0")
    implementation("androidx.coordinatorlayout:coordinatorlayout:1.2.0")
    implementation("androidx.viewpager2:viewpager2:1.0.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    implementation("com.google.firebase:firebase-analytics")
    implementation(platform("com.google.firebase:firebase-bom:32.8.0"))
    implementation("com.google.firebase:firebase-auth")
    implementation("com.google.firebase:firebase-firestore:24.11.0")
    implementation("com.google.code.gson:gson:2.8.8")
    implementation("com.airbnb.android:lottie:4.2.0")
}
```

- androidx.core:core-ktx:1.9.0: Kotlin extensions para la biblioteca de soporte principal de Android.
- androidx.appcompat:appcompat:1.6.1: Proporciona funcionalidades de compatibilidad con versiones anteriores de Android para aplicaciones modernas.
- com.google.android.material:material:1.11.0: Implementa los principios de Material Design en tus aplicaciones.
- androidx.constraintlayout:constraintlayout:2.1.4: Biblioteca para crear diseños flexibles y eficientes.
- androidx.lifecycle:lifecycle-livedata-ktx:2.7.0: Proporciona clases LifecycleOwner y LiveData para una comunicación segura entre componentes de Android.
- androidx.lifecycle:lifecycle-viewmodel-ktx:2.7.0: Proporciona una implementación de ViewModel de Android que se puede compartir entre fragmentos y actividades.
- androidx.navigation:navigation-fragment-ktx:2.5.3: Componentes de navegación para Android, fragmento específico.
- androidx.navigation:navigation-ui-ktx:2.5.3: Componentes de navegación para Android, específico para UI.
- androidx.drawerlayout:drawerlayout:1.2.0: Implementa la funcionalidad del cajón de navegación en las aplicaciones de Android.
- androidx.coordinatorlayout:coordinatorlayout:1.2.0: Proporciona una interfaz para coordinar las vistas en jerarquías de vista complejas.
- androidx.viewpager2:viewpager2:1.0.0: Componente ViewPager2 que permite la navegación mediante gestos deslizantes.
- testImplementation("junit:junit:4.13.2"): Framework de pruebas unitarias para Java.
- androidx.test.ext:junit:1.1.5: Extensión JUnit para pruebas de instrumentación en Android.
- androidx.test.espresso:espresso-core:3.5.1: Framework para escribir pruebas de interfaz de usuario.
- com.google.firebase:firebase-analytics: SDK de Firebase Analytics para recopilar y analizar datos de uso de tu aplicación.
- com.google.firebase:firebase-auth: SDK de Firebase Authentication para autenticación de usuarios.
- com.google.firebase:firebase-firestore:24.11.0: SDK de Firebase Firestore para interactuar con la base de datos en la nube Firestore.
- com.google.code.gson:gson:2.8.8: Biblioteca para serializar y deserializar objetos JSON.
- com.airbnb.android:lottie:4.2.0: Biblioteca para la reproducción de animaciones de Adobe After Effects en aplicaciones nativas de Android.

Todas las dependencias son realizadas por desarrolladores profesionales pertenecientes a una empresa o equipo de desarrollo formal, por lo que no existen importaciones de librerías realizadas por el usuario de la aplicación. Dichas librerías no satisfacen necesidades locales, debido a que contemplan una cantidad de funcionalidades aptas para que muchos desarrolladores las puedan utilizar.

Las dependencias nos brindan una amplia cantidad de funcionalidades, lo cual permite implementarlas de diferentes maneras en el código de la aplicación. Existen varias maneras a seguir en la estructuración del código, permitiendo así saber cuándo, cómo y dónde implementar dichas funcionalidades.

DeliVibes utiliza un código principalmente estructurado-procedimental, lo que supone que, al compilar la aplicación, el código se ejecuta en función del orden en el que está escrito, es decir, que se ejecuta la primera línea, después la segunda... y así hasta la última, siguiendo un orden.

No obstante, existen partes del código en las que se utiliza una POO (programación orientada a objetos), de esta manera se utilizan objetos que representan clases con sus propios métodos, haciendo así una programación más legible y rápida.

### 6.4.3 CUESTIONES DE DISEÑO E IMPLEMENTACIÓN RESEÑABLES

La realización del proyecto ha supuesto un gran desafío, debido a que mucha de la funcionalidad era desconocida (tanto parte back como front), por lo que algunos elementos han sido de gran complejidad manejar. De los cuales resaltan:

- Drawer: Comúnmente conocido como el menú desplegable de la izquierda, fue difícil de manejar, debido a que se necesita un tipo de clase específica para su implementación. Las clases prototipo para usar ese componente son las llamadas *activtys*, son clases independientes en las que no tienen correlación con las demás; es por eso que para utilizar este componente es necesario usar *fragments*, una estructura que depende de una *activity* para que se pueda mostrar. Los *fragments* funcionan como el contenido principal de una *activity*, y cada *fragment* se puede “inflar” para que una sola *activity* pueda ir cambiando el contenido que contiene, por lo que la *activity* principal sería el *drawer* y los diferentes *fragments* se van inflando y desinflando (mostrando y ocultando) en función de la pestaña de la aplicación a la que accedemos.
- Consultas a la BD: Al ser Firebase una base de datos no relacional, no consta de los mismos procesos para acceder que las bases de datos relacionales. La manera de hacer consultas pasa por un procedimiento previo, el cual se encarga de recorrer todos y cada uno de los documentos que hay en la colección, hasta encontrar el documento que coincida con los datos que se pasan en la consulta.
- Layouts: Para el diseño de la interfaz, en Android Studio se utilizan los *layouts*, que son los encargados de cómo va a ser la disposición de los elementos de la interfaz, existen los *LinearLayout* (el cual los elementos van uno a la derecha del otro, sin poder modificar su posición), *RelativeLayout* (el cual nos permite poner los elementos en el lugar deseado) ..., por lo que en algunas interfaces había conflictos sobre qué *layout* utilizar y cómo se deben usar en función de los elementos.

## 6.5 PRUEBAS

Las pruebas son una parte muy importante en el desarrollo, ya que se debe probar todo aquello que se programa y así detectar posibles fallos para posteriormente corregirlos. Al ser DeliVibes una aplicación móvil, no existe la posibilidad de usarse mediante una web, por lo que no se pueden realizar pruebas con los navegadores. Sin embargo, los dispositivos móviles tienen sistemas operativos diferentes, ya sean Android o iOS, y dado que la aplicación es nativa para Android tampoco existe la posibilidad de probar DeliVibes en el sistema operativo iOS.

No obstante, no son las únicas pruebas disponibles que se pueden realizar, las pruebas más eficientes son contra la propia aplicación, es decir, probar diferentes métodos, datos, opciones dentro de la aplicación hasta que se rompa, de esta manera encontraremos fácilmente los fallos.

Dicho esto, se han realizado una serie de pruebas para probar el correcto funcionamiento de la aplicación:

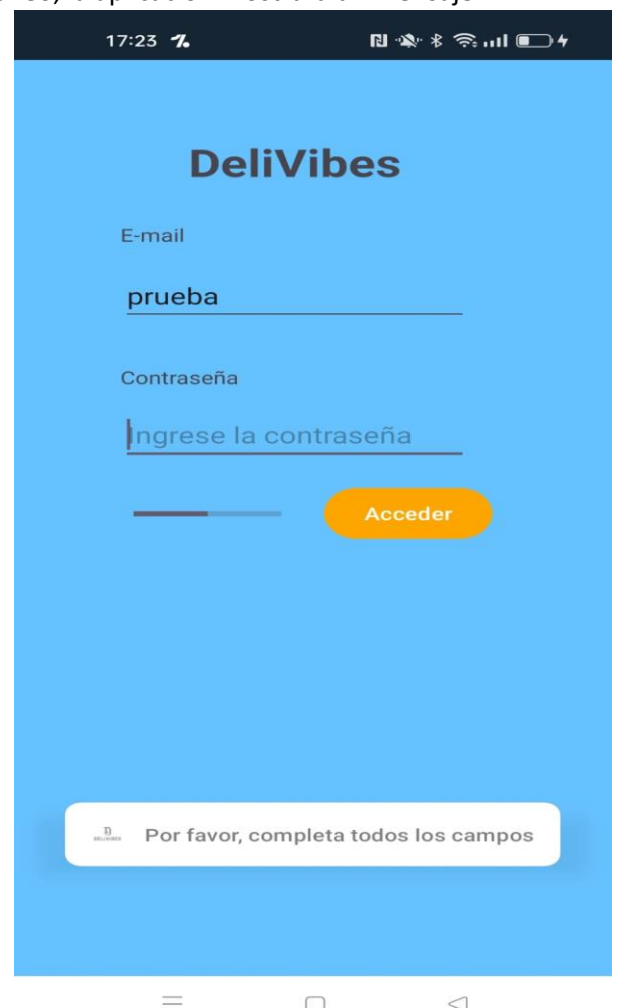
- Prueba 1: Autenticación de usuarios.

El usuario debe entrar a la aplicación mediante el inicio de sesión en caso de que tenga una cuenta creada, nos pedirá el correo y la contraseña.

Como podemos observar, si el usuario intenta acceder con un correo o contraseña erróneo, la aplicación mostrará un mensaje de que hay un error con los datos introducidos.

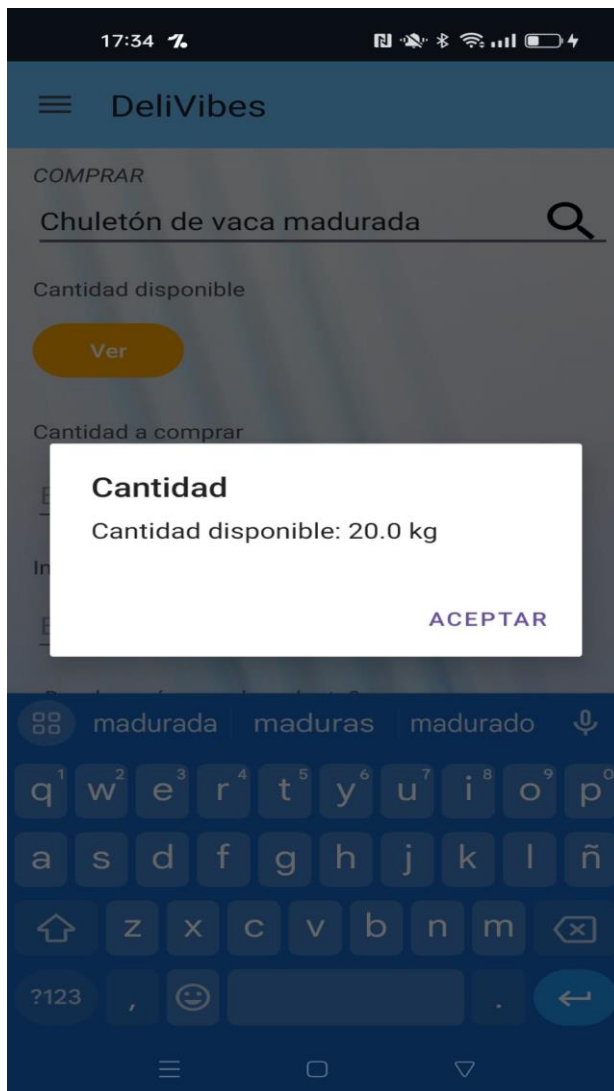
Además, si alguno de los campos está sin seleccionar, también se mostrará ese aviso.

En la aplicación sólo se contempla la posibilidad de que la aplicación la utilice un usuario en cada dispositivo en la que esté instalado, por lo que es imposible la compatibilidad de varios usuarios desde un mismo dispositivo.





- Prueba 2: Búsqueda de la cantidad disponible de un producto.



El usuario tiene la posibilidad de ver qué cantidad restante de producto hay en la tienda.

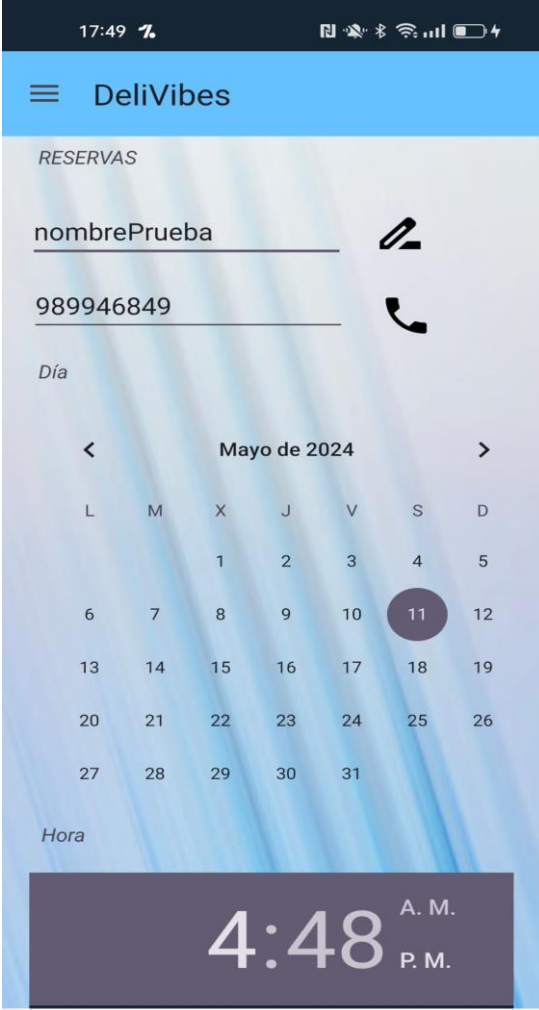
Para dicha búsqueda, el usuario debe escribir el nombre del producto que desee consultar la cantidad, para que pueda pulsar el botón de “Ver”, en el que aparecerá la cantidad del producto.

En cambio, si el usuario decide pulsar el botón de “Ver” sin previamente haber escrito un nombre en la barra de búsqueda, no se podrá acceder a la cantidad de ese producto.



También se muestra el mensaje de “cantidad no disponible” si en la barra de búsqueda escribimos un producto que actualmente no se encuentra en la tienda.

- Prueba 3: Proceso de reserva



17:49

RESERVAS

nombrePrueba

989946849

Día

Mayo de 2024

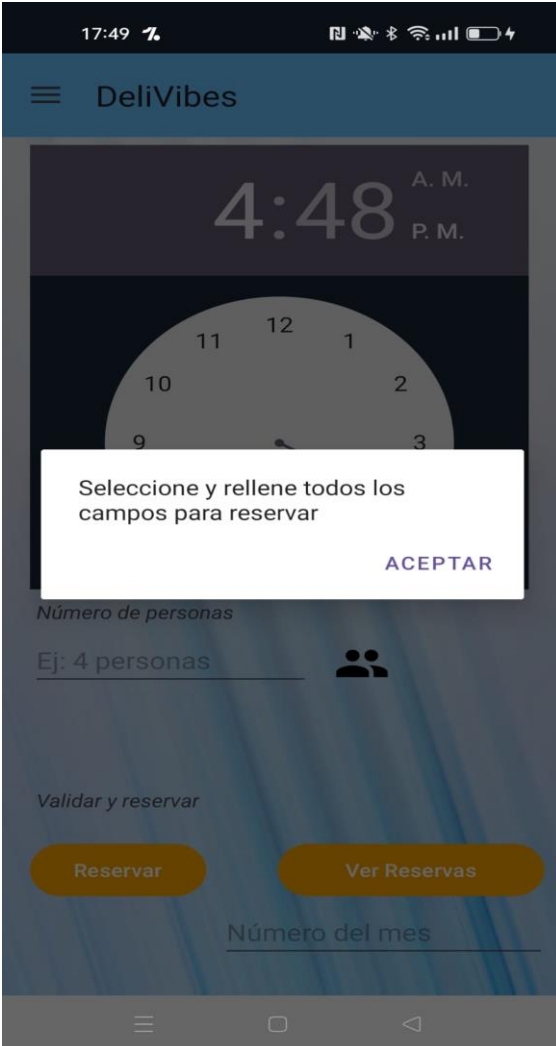
L	M	X	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Hora

4:48 A. M. P. M.

Si todos los campos de la reserva se han rellenado correctamente, se procede a hacer la comprobación de los datos, para comprobar que no haya una reserva con los mismos datos que se han realizado.

La reserva es una de las partes más importantes de la aplicación, ya que permite al usuario elegir el día y la hora para ir a degustar el menú. Sin embargo, el proceso de reserva requiere una inserción de datos para poder efectuarse correctamente. Para saber si todos los campos se han rellenado correctamente, basta con pulsar el botón de “Reservar”. Si todos los campos no se han rellenado, mostrará un error.



17:49

DeliVibes

4:48 A. M. P. M.

Seleccione y rellene todos los campos para reservar

ACEPTAR

Número de personas

Ej: 4 personas

Validar y reservar

Reservar Ver Reservas

Número del mes

Estas sólo son algunas de las pruebas que se pueden realizar en la aplicación, en el manual de usuario (apartado 7.1) se realizarán todas las pruebas posibles.

## 7. MANUALES DE USUARIO

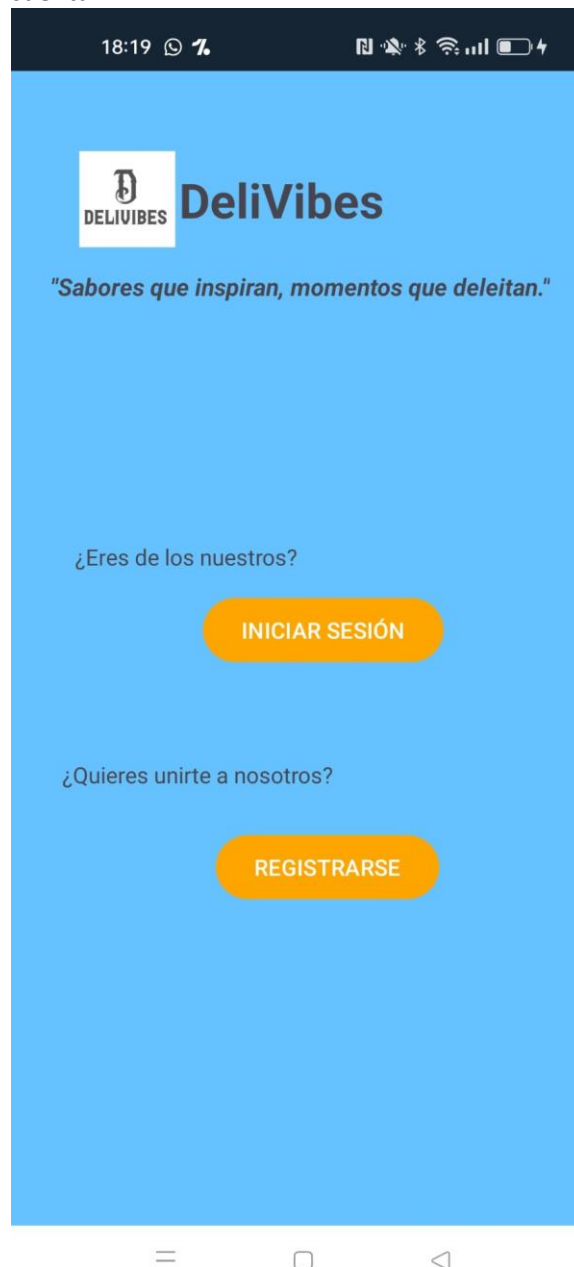
### 7.1 MANUAL DE USUARIO

Manual en el que se explican todas las funcionalidades que tiene la aplicación, así mismo se muestran fotos de las pantallas relacionadas con dichas funcionalidades

#### 7.1.1 PANTALLA DE CARGA

Pantalla que aparece nada más arrancar la aplicación, en la cual da la bienvenida a la misma.

Dicha pantalla ofrece pulsar dos botones, si se pulsa el botón “iniciar sesión” lleva al usuario a la página para iniciar sesión con sus credenciales, por el contrario, si se pulsa el botón de “registrarse”, se accede a la página de registro, para crear una nueva cuenta.

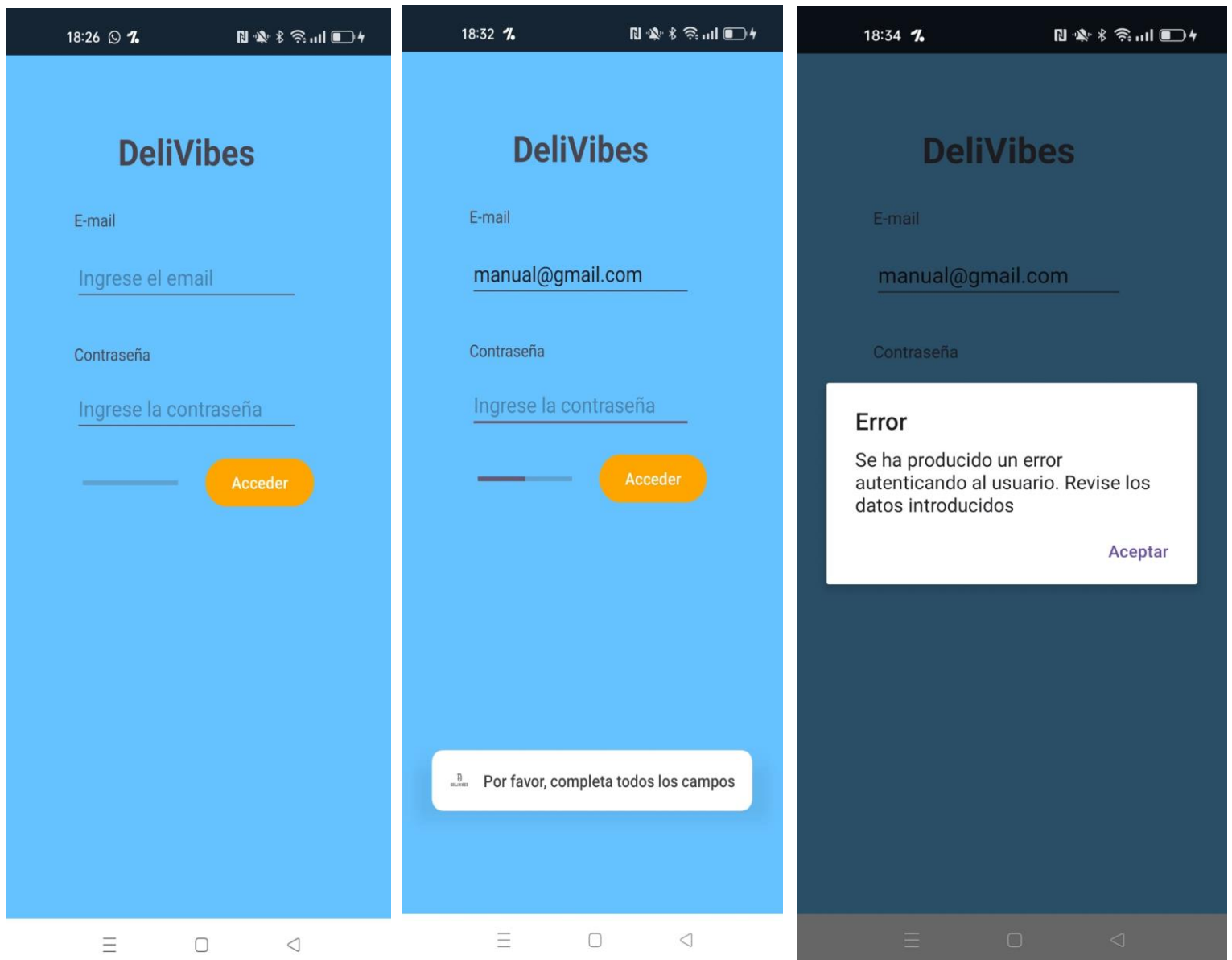


### 7.1.2 PANTALLA DE INICIO DE SESIÓN

Pantalla en la cual se insertan el e-mail y la contraseña propia de cada usuario para el inicio de sesión. Si hay algún campo que no se ha rellenado, muestra un mensaje de error, en el que indica que se tiene que rellenar todos los campos.

Si algún dato es erróneo, se mostrará un mensaje de error en el que nos dirá que hay algún dato que está mal.

Todos estos mensajes se muestran pulsando el botón “acceder”, el cual se encarga de realizar las comprobaciones con los datos insertados. Si la comprobación se ha realizado y los datos son correctos, el usuario accederá al resto de pantallas.

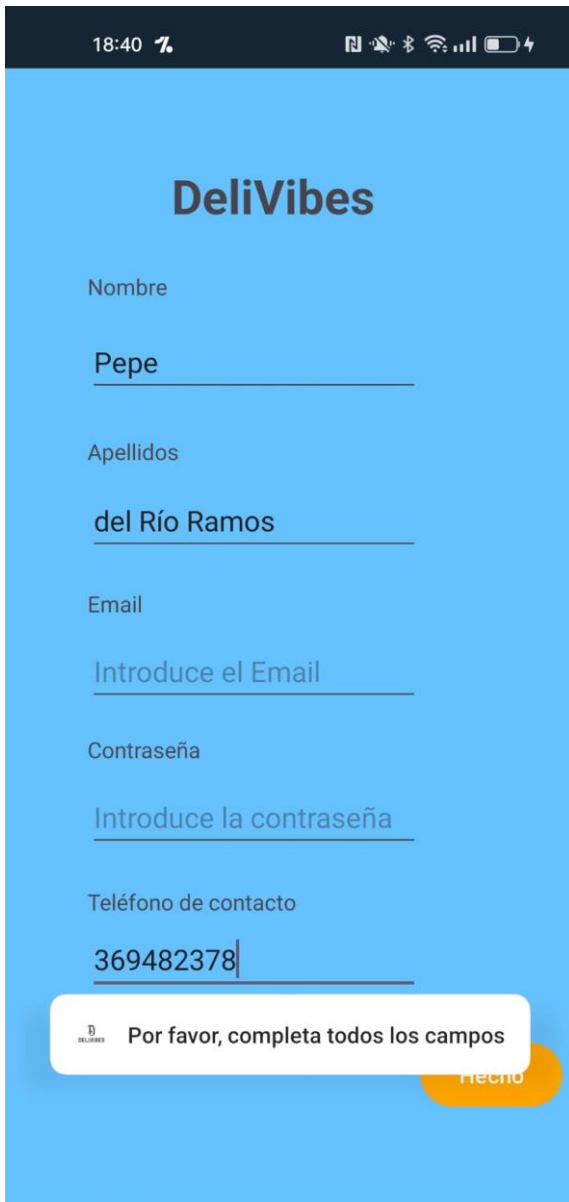


### 7.1.3 PANTALLA DE REGISTRO

Pantalla que permite al usuario crearse una cuenta en la aplicación, consta de rellenar todos los campos necesarios. Una vez rellenados todos los campos, se pulsa al botón “hecho”, que comprueba si toda la información es correcta, si es así se creará una cuenta al usuario.

Si no están todos los campos rellenados, mostrará un error diciendo que se debe rellenar todos los campos.

La contraseña es un campo que requiere unos requisitos, por lo que, si no cumple con dichos requisitos, mostrará un error, al igual que el campo e-mail, que requiere que el correo acabe en .com



18:40

**DeliVibes**

Nombre

Pepe

Apellidos

del Río Ramos

Email

Introduce el Email

Contraseña

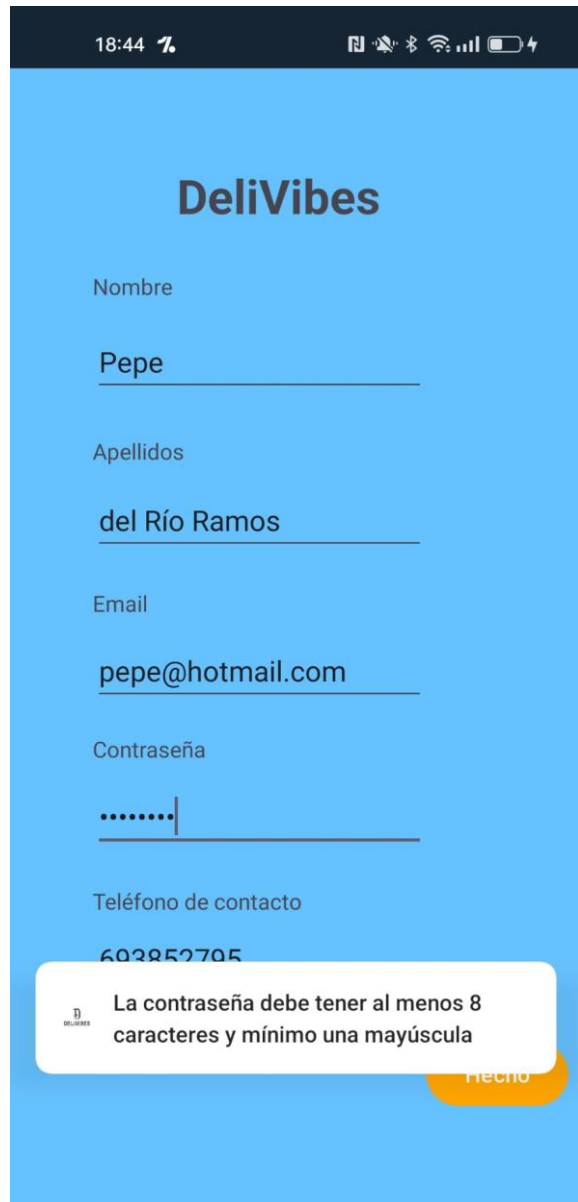
Introduce la contraseña

Teléfono de contacto

369482378

Por favor, completa todos los campos

hecho



18:44

**DeliVibes**

Nombre

Pepe

Apellidos

del Río Ramos

Email

pepe@hotmail.com

Contraseña

.....

Teléfono de contacto

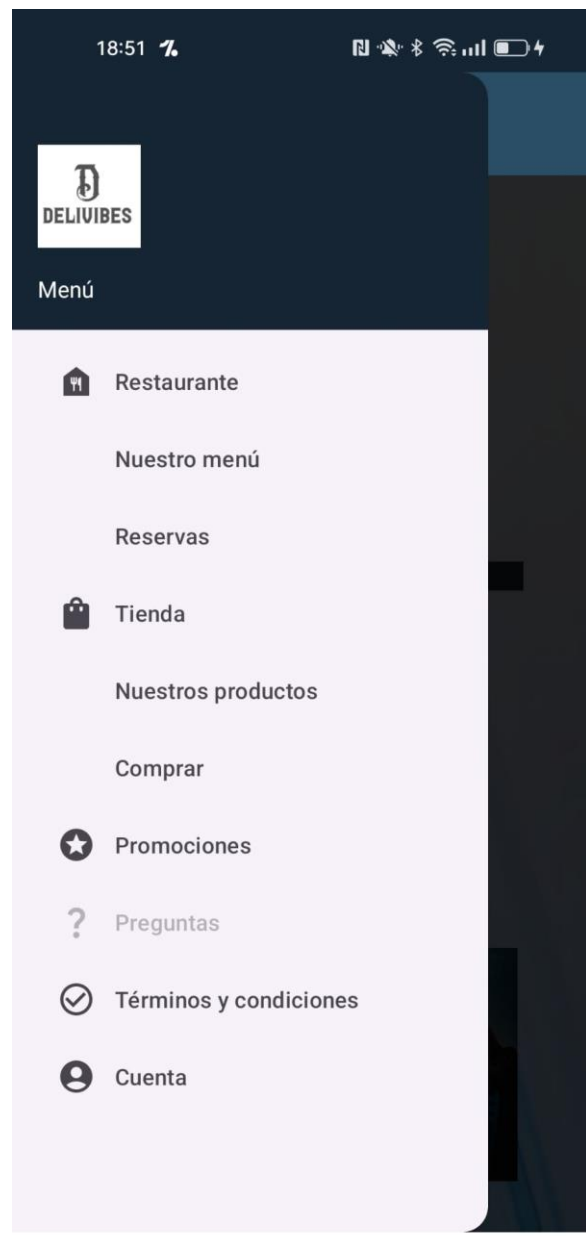
603852705

La contraseña debe tener al menos 8 caracteres y mínimo una mayúscula

hecho

### 7.1.4 PANTALLA PRINCIPAL/INICIO

Pantalla a la que se accede una vez el usuario se ha registrado o iniciado sesión. Esta pantalla nos muestra un pequeño resumen de lo que se puede hacer en la aplicación, como ver el menú, los productos o las promociones, además consta con un menú desplegable en el cual se muestran todas las pantallas a las que se pueden acceder, las cuales son “Nuestro menú”, “Reservas”, “Nuestros productos”, “Compra”, “Promociones”, “Términos y condiciones”, “Cuenta”. Todas las pantallas tienen la posibilidad de ver el menú desplegable y seleccionar la opción deseada.





### 7.1.5 PANTALLA DE NUESTRO MENÚ

Pantalla a la que se accede pulsando “Nuestro menú” en el menú desplegable. En esta pantalla, se muestra el menú que actualmente tiene el restaurante, pudiéndose ver el primer plato, segundo y postre. Para ver dichos platos, basta con pulsar el rectángulo naranja, en el que siempre va a aparecer “Primer plato” nada más acceder a la pantalla (por defecto el usuario va a querer ver primero los primeros platos), por lo que si se pulsa dicho rectángulo, aparecerán las opciones de selección de los tres componentes del menú. Se actualizarán las tarjetas en función de la opción seleccionada, excepto el precio que siempre es el mismo se escoja lo que se escoja.



### 7.1.6 PANTALLA DE RESERVAS

Pantalla a la que se accede pulsando “Reservas” en el menú desplegable.

En esta pantalla se permite al usuario realizar una reserva en el restaurante, para ello es necesario rellenar todos los campos necesarios, los cuales son nombre de la persona que realiza la reserva, teléfono de contacto y número de personas.

El usuario debe conocer y saber que la hora y el día se guardan en la base de datos con el horario actual, es decir, si el usuario no selecciona una hora y fecha, el programa realizará la reserva para el día y fecha actuales de ese momento.

Para realizar la reserva, se debe pulsar al botón de “Reservar”, el cual comprueba dos cosas, la primera de ella es que todos los campos estén rellenos correctamente (si no es así mostrará un mensaje de error), y la segunda es que los datos de la reserva no coincidan con otra reserva, es decir, que el día y la hora no coincidan con otra reserva que tenga esos mismos valores (si esos datos coinciden, mostrará un mensaje de error).

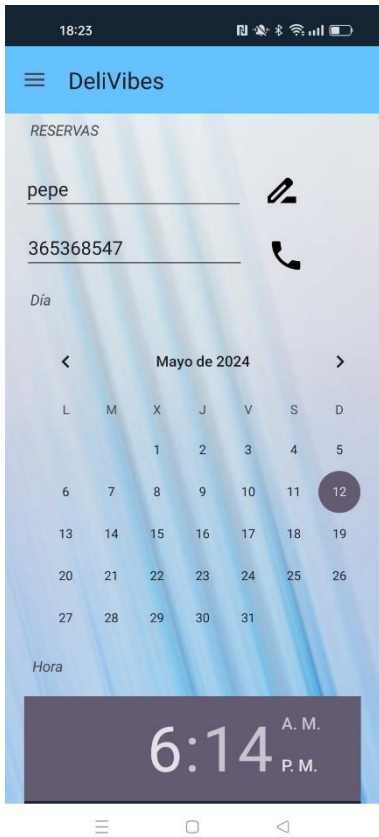
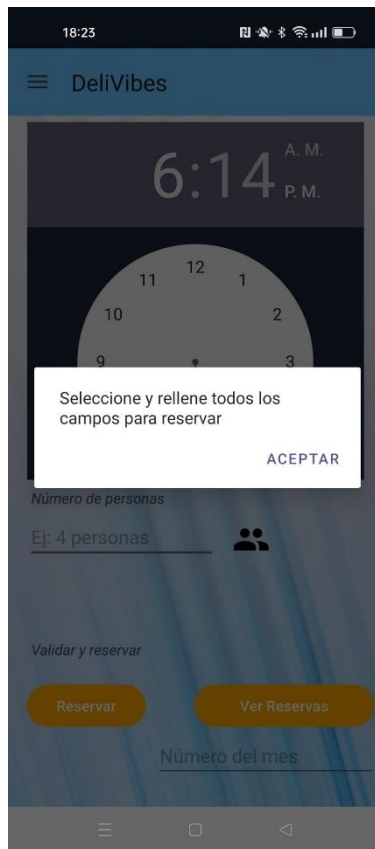
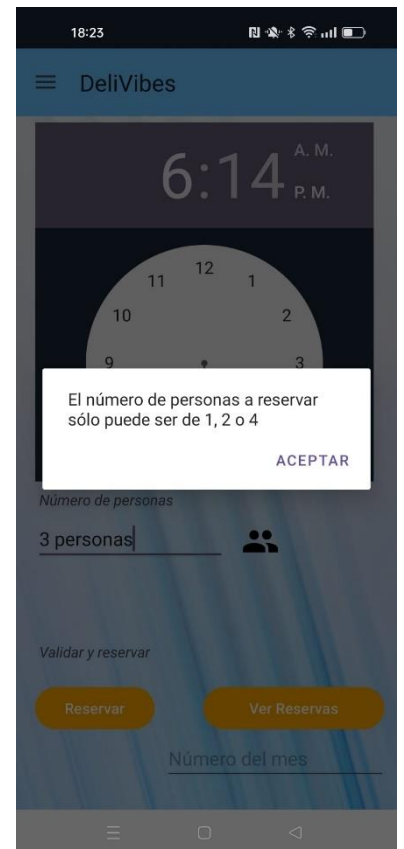
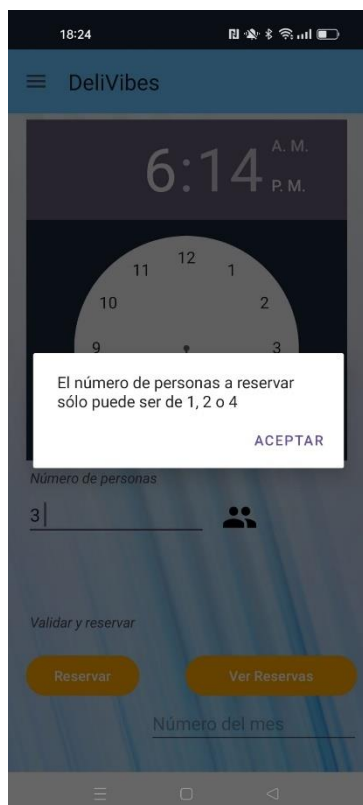
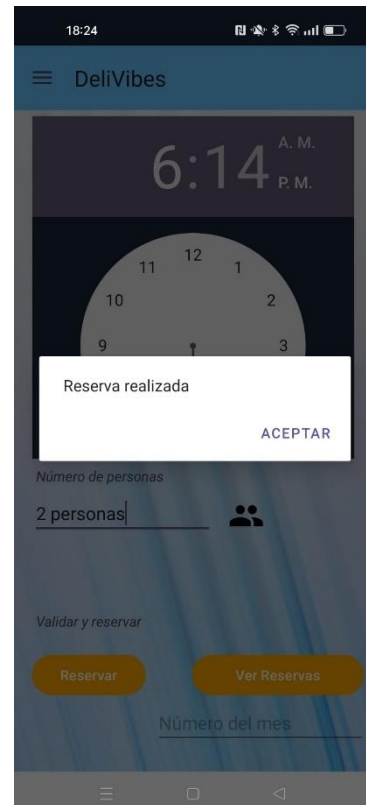
Además, existe otro factor muy importante, sólo se puede realizar la reserva para 1, 2 o 4 personas, por lo que en caso de querer realizar la reserva para un número de personas diferente, mostrará un error. Siguiendo con este campo, el formato es muy explícito, ya que se tiene que escribir el número de personas de una manera concreta para su mejor almacenamiento en la base de datos, lo primero que se debe poner es el número de personas (1,2 o 4), seguido de la palabra “personas”, si el dato introducido no corresponde con el formato impuesto, mostrará un mensaje de error.

En caso de que todo esté correcto, se procederá a realizar la reserva.

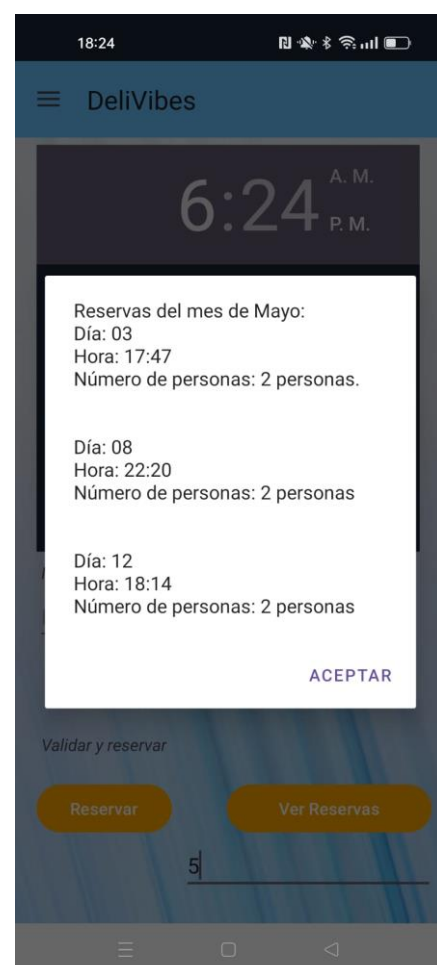
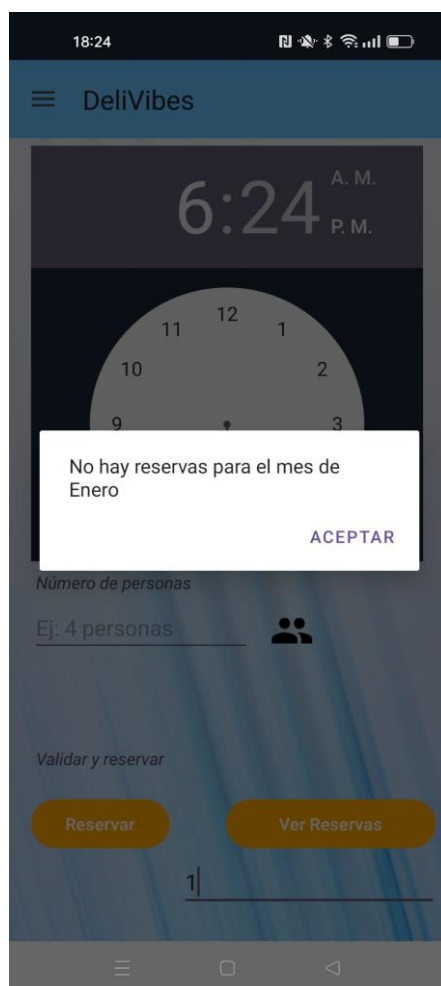
Realizar la reserva no es lo único que se puede hacer en la pantalla, existe un campo que pone “Número del mes” y un botón “Ver Reservas”, dichos elementos sirven para ver que reservas hay en un mes, dando así facilidad al usuario para ver si el día y la hora que quiere reservar están disponibles. Si se pulsa el botón sin antes escribir el número del mes, mostrará un mensaje de error. Si se escribe el número del mes puede pasar dos cosas, que se muestren las reservas de dicho mes, o que no muestre ninguna reserva porque en ese mes no se ha realizado ninguna.



Proceso de reserva:

Proceso de vista de reservas:

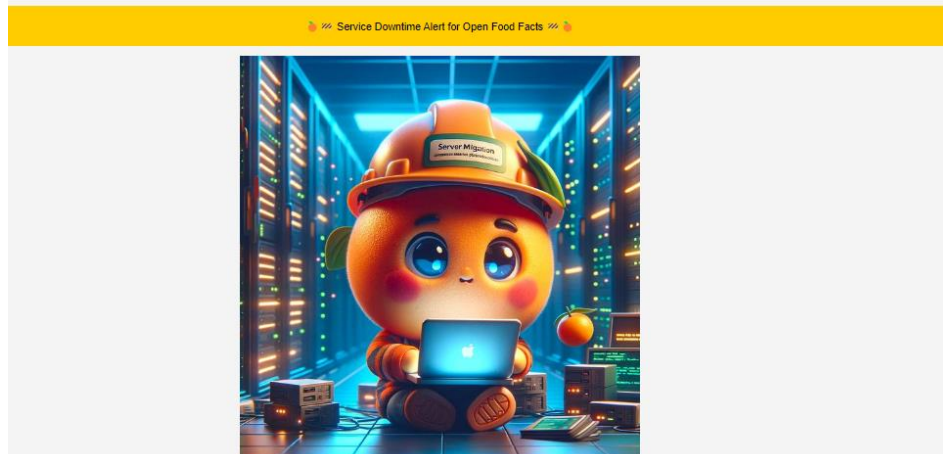


## 7.1.7 PANTALLA DE NUESTROS PRODUCTOS

Pantalla a la que se accede pulsando “Nuestros productos” en el menú desplegable. En esta pantalla, se muestran los productos que hay actualmente en la tienda, pudiendo ver los nombres de los mismos y el precio en kg (o en docenas si son huevos).

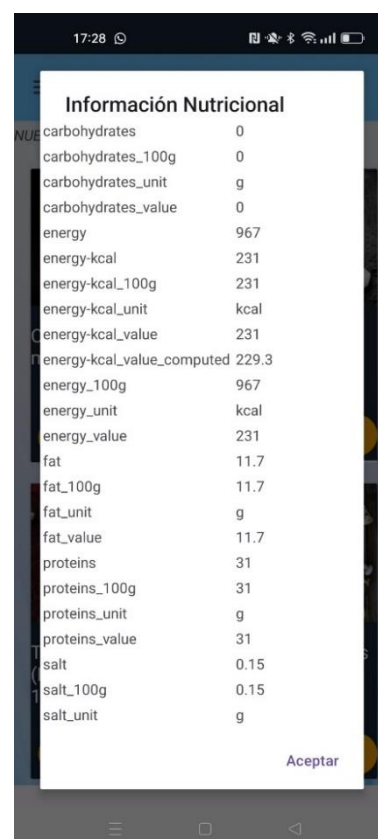
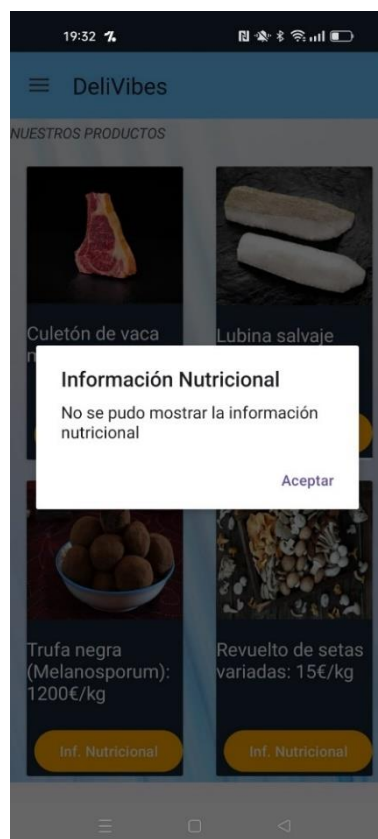
Cada producto consta de un botón “Inf. Nutricional”, en el cual, si se pulsa, se puede ver la información nutricional del producto.

Existe la posibilidad de que la API está en mantenimiento, por lo que no se podrá acceder a la información nutricional y nos dará un error.



### Open Food Facts Maintenance

We are currently undergoing a major migration, and our services, including the Web platform, mobile app, API, and Producer Platform, are temporarily unavailable. The downtime is expected to last for approximately 1 hour. We apologize for any inconvenience this may cause and appreciate your understanding as we work to improve our services. Thank you for your patience!



### 7.1.8 PANTALLA DE COMPRA

Pantalla a la que se accede pulsando “Comprar” en el menú desplegable.

En esta pantalla, el usuario puede comprar el producto que desee, para ello debe poner el nombre del producto en la barra de búsqueda, a continuación, debe pulsar el botón “Ver”, para que muestre al usuario la cantidad restante de ese producto, después se habilitarán el resto de los campos, (cantidad a comprar, información adicional y dirección).

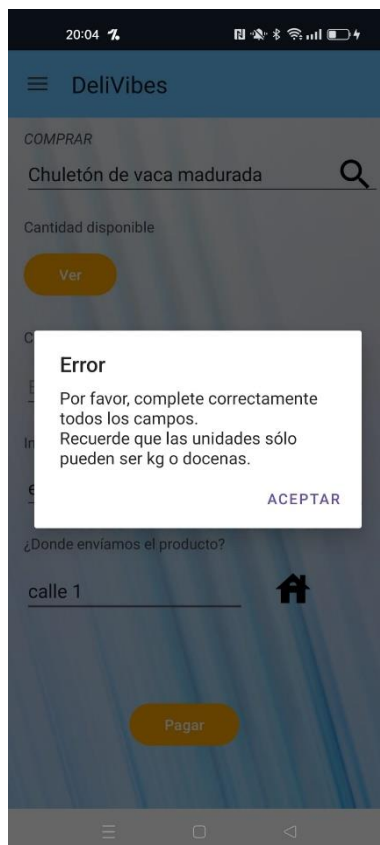
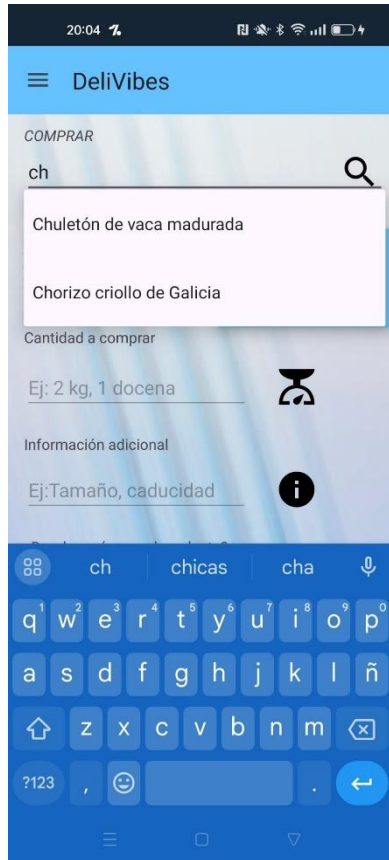
Una vez completado los campos, se debe pulsar el botón “Pagar”, en el cual se comprobará la información añadida. Cabe destacar que, si hay algún campo que no se ha rellenado, se mostrará un mensaje de error, además, como se ha explicado previamente, existen campos que están deshabilitados, para que se vuelvan a activar, se tiene que pulsar el botón “Ver”, el cual muestra la cantidad del producto, se realiza esta técnica para evitar que el usuario compre una cantidad de un producto sin saber la cantidad del mismo.

El campo de cantidad requiere un patrón específico, se debe escribir un total de un número y una palabra, es decir, 2 kg, 3 docenas..., en caso de no ser así mostrará un error debido a un formato incorrecto. Ocurre lo mismo si se escribe una cantidad mayor del producto de la que existe en la tienda, la aplicación mostrará un error diciendo que no hay tanta cantidad de producto.

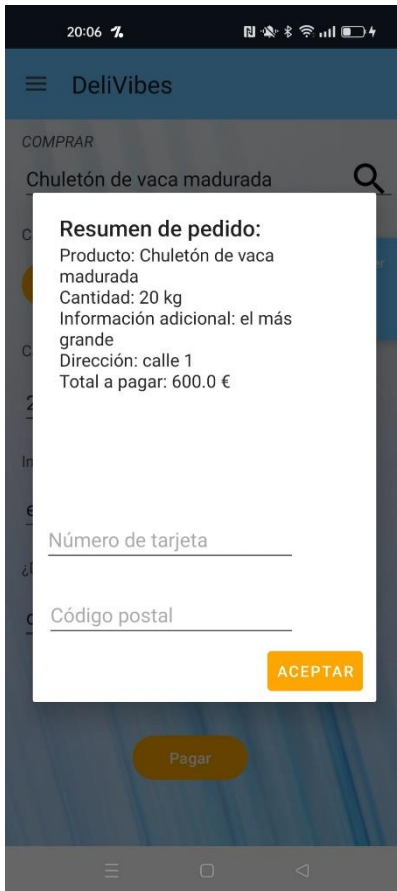
En la parte superior de la pantalla hay un bacadillo, el cual nos recuerda que para que el resto de los campos se deshabiliten debemos pulsar el botón “Ver” con un producto escrito. Dicho bacadillo se puede ocultar pulsando el botón “Ocultar”.

Si al pulsar el botón “Pagar” todo está correctamente, aparecerá un dialog, en el cual mostrará un resumen de lo que se va a pagar, indicando el nombre del producto, la cantidad, la información adicional y la dirección. En dicho dialog debemos rellenar dos campos para efectuar el pago, los cuales son el número de tarjeta y el código postal. Si ambos campos se han rellenado, el pago se realizará con éxito, y la cantidad del producto se actualizará en la base de datos, en caso de no haberse rellenado los dos campos, se mostrará un mensaje de error, diciendo que los campos deben ser rellenados.

Proceso de compra:



Proceso de pago:



20:06

DeliVibes

COMPRAR

Chuletón de vaca madurada

**Resumen de pedido:**  
Producto: Chuletón de vaca madurada  
Cantidad: 20 kg  
Información adicional: el más grande  
Dirección: calle 1  
Total a pagar: 600.0 €

Número de tarjeta

Código postal

ACEPTAR

Pagar



20:07

DeliVibes

COMPRAR

Chuletón de vaca madurada

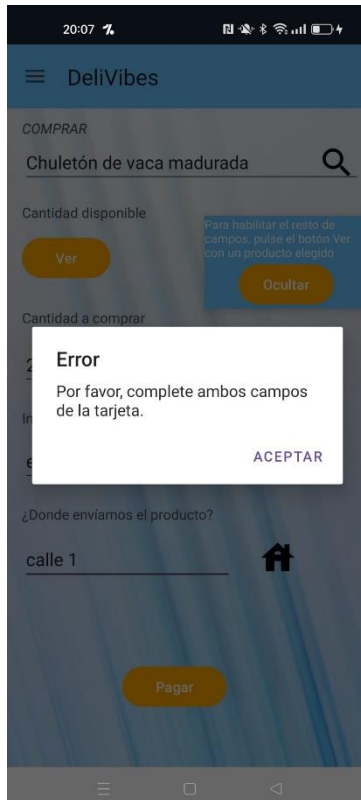
**Resumen de pedido:**  
Producto: Chuletón de vaca madurada  
Cantidad: 20 kg  
Información adicional: el más grande  
Dirección: calle 1  
Total a pagar: 600.0 €

1248438944686497

Código postal

ACEPTAR

Pagar



20:07

DeliVibes

COMPRAR

Chuletón de vaca madurada

Cantidad disponible

Ver

Ocultar

Cantidad a comprar

**Error**  
Por favor, complete ambos campos de la tarjeta.

ACEPTAR

¿Donde enviamos el producto?

calle 1

Pagar



20:07

DeliVibes

COMPRAR

Chuletón de vaca madurada

Cantidad disponible

Ver

Ocultar

Cantidad a comprar

**Pago**  
¡Pago realizado con éxito!

ACEPTAR

¿Donde enviamos el producto?

calle 1

Pagar



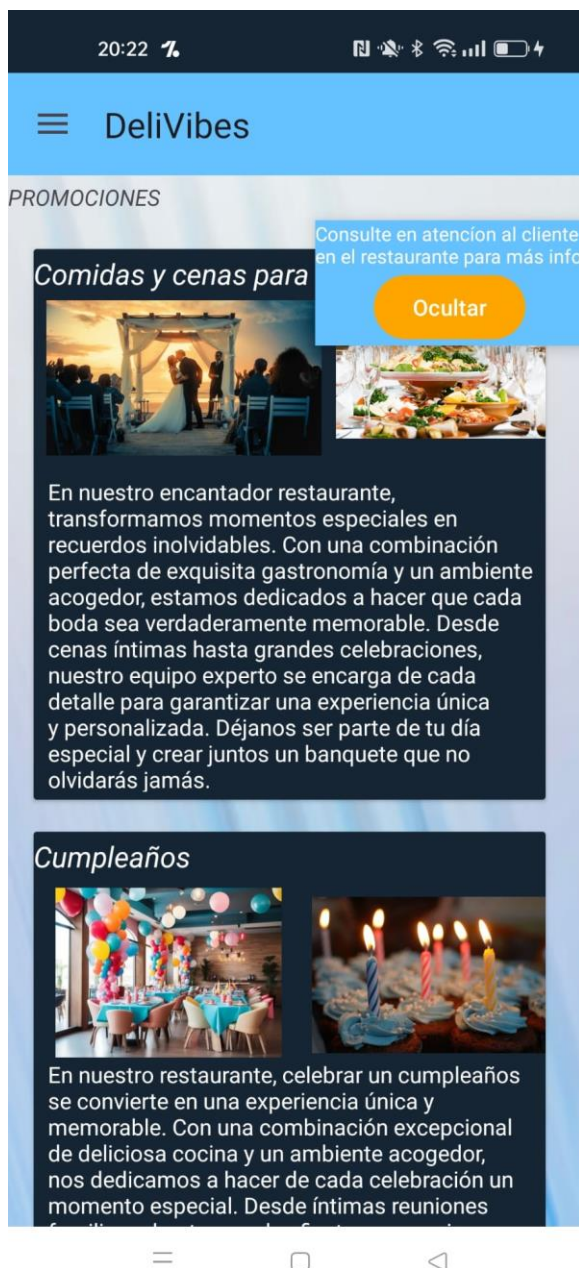
### 7.1.9 PANTALLA DE PROMOCIONES

Pantalla a la que se accede pulsando “Promociones” en el menú desplegable.

En esta pantalla, el usuario puede consultar y leer todas las promociones que ofrece el restaurante, entre ellas se encuentran comidas y cenas para bodas, cumpleaños y Navidad y fin de año. Cada promoción tiene una breve descripción de la misma, en la que se puede leer de qué consta cada una de ellas.

Además, la pantalla cuenta con un bocadillo, el cual indica que para obtener más información, se debe de acceder a atención al cliente en el propio restaurante.

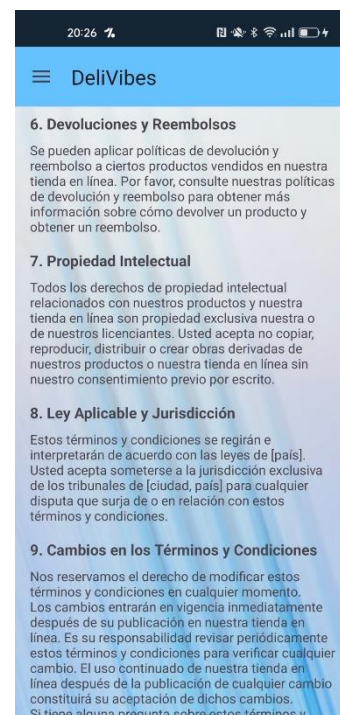
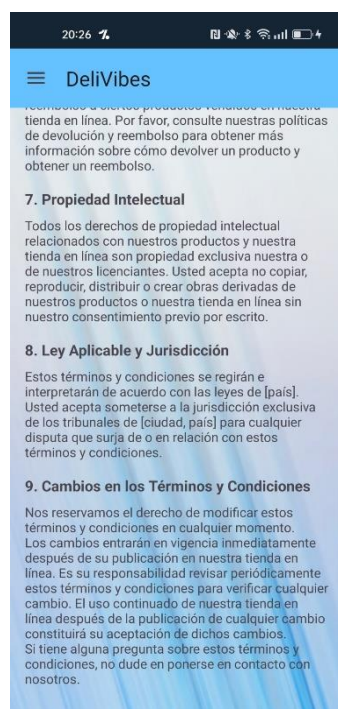
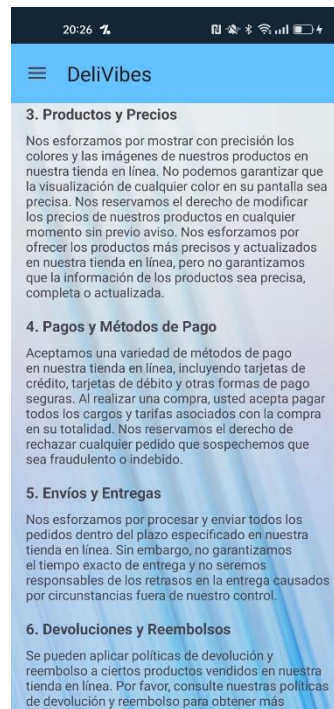
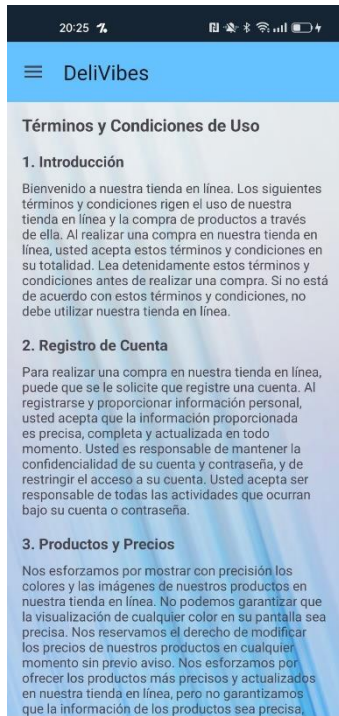
Dicho bocadillo se puede ocultar pulsando el botón “Ocultar”.



## 7.1.10 PANTALLA DE TÉRMINOS Y CONDICIONES

Pantalla a la que se accede pulsando “Términos y condiciones” en el menú desplegable.

En esta pantalla, el usuario puede leer los términos y las condiciones del restaurante y de la tienda, para de esta manera saber cómo se rigen dichos establecimientos.



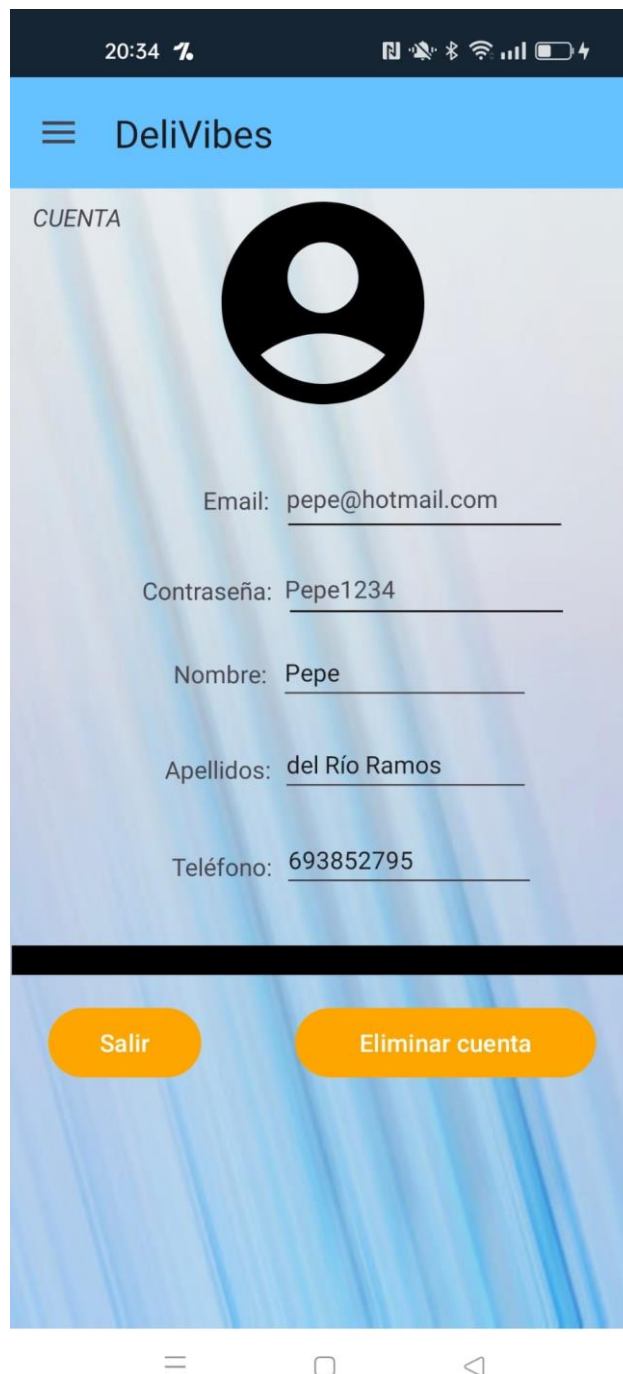


### 7.1.11 PANTALLA DE CUENTA

Pantalla a la que se accede pulsando “Cuenta” en el menú desplegable.

En esta pantalla, el usuario puede ver sus datos personales, es decir, los datos con los que se registró en la aplicación, los cuales son Email, Contraseña, Nombre, Apellidos y Teléfono.

Además, la pantalla consta de dos botones, el primero es “Salir”, el cual si se pulsa se cierra la sesión actual, volviendo a la pantalla de carga. El segundo botón es “Eliminar cuenta”, el cual si se pulsa nos cierra la sesión actual y vuelve a la pantalla de carga, además nos elimina la cuenta de la base de datos, por lo que no se puede volver a iniciar sesión con esa cuenta.



## **7.2 MANUAL DE INSTALACIÓN**

Manual que permite al usuario conocer cómo debe de instalarse la aplicación.  
Dicho manual puede entenderse también como la forma de desplegar la aplicación, por lo que viene referenciado en el apartado 4.3.2.

## **8. CONCLUSIONES Y POSIBLES AMPLIACIONES**

### **8.1 CONCLUSIONES**

Un proyecto ambicioso, costoso y satisfactorio.

Desde el primer momento, sabía de las complicaciones que tenía el desarrollo del proyecto, ya que en un tiempo de 30 horas, la amplitud de la aplicación no iba a ser muy grande, aún así decidí hacer unas 80 horas para que la aplicación tuviese el mayor contenido posible relacionado con la temática de la misma.

No fue fácil desarrollar DeliVibes, al elegir Kotlin como lenguaje de desarrollo, tenía que aprender primero cómo funcionaba y que funcionalidades se podrían implementar, lo cual hizo que el progreso fuese muy lento al principio. Además, al implementar una base desconocida como es Firebase, sabía que me iba a llevar tiempo comprender cómo era su funcionamiento, ya que al ser una base de datos no relacional, no estaba acostumbrado a trabajar con ese tipo de BD.

Los objetivos se han cumplido en la medida de lo posible, ya que la idea principal de la aplicación se ha realizado con éxito en un tiempo más que asequible, aunque algunas funcionalidades más que quería añadir no pudieron realizarse debido al tiempo necesitado para su desarrollo.

Concluyendo, ha sido una aplicación difícil de desarrollar, ya que el lenguaje y la base de datos utilizada no habían sido vistos, por lo que requería comprender cómo funcionaban previamente, lo cual se hizo mediante vídeos de YouTube y chatgpt, además eso conllevó mucho tiempo, lo cual llevó a sobrepasar el tiempo estimado de la aplicación a 80 horas. Os objetivos se han cumplido, habiendo hecho así la aplicación acorde a las ideas principales.

Estoy orgulloso de la aplicación que ha sido desarrollada.

### **8.2 POSIBLES AMPLIACIONES**

La aplicación consta de una gran cantidad de funcionalidades, lo cual ha requerido bastante tiempo, aun así, si el tiempo impuesto hubiese sido más elevado, la aplicación hubiera constado de más funcionalidad y pantallas, las cuales serían ampliaciones en un futuro para la misma.

Algunas de las ampliaciones serían:

- Implementar un historial en la pantalla de cuenta: La pantalla tendría una opción para ver el historial de reservas y compras que ha hecho el usuario.
- Preguntas: Pantalla que está actualmente en la aplicación, pero deshabilitada. La idea es implementar un chatbox, en el cual el usuario pueda preguntar cualquier cuestión relacionada con el restaurante/ tienda, y el sistema le contestaría con una respuesta adecuada a la pregunta formulada.
- Cancelación: Ya que la aplicación permite la realización de la reserva, una ampliación sería poder cancelar la reserva, para ello se le enviaría al usuario un número de reserva, el cual debe guardar e insertar en el apartado de cancelar reserva, dicho apartado de cancelación se ubicaría en la pantalla de reservas.

- Devolución de productos: El usuario puede comprar productos, por lo que una idea a desarrollar sería la posibilidad de devolverles en un plazo de tiempo, se le otorgaría al usuario un número de pedido, el cual debe de insertar en el apartado de devoluciones, que se ubicaría en la pantalla de los productos.

## 9. BIBLIOGRAFÍA

<https://www.youtube.com/watch?v=t5yyc1XfQrs&t=11s>  
<https://www.youtube.com/watch?v=WTkiwXDh8Aw&t=11s>  
<https://www.youtube.com/watch?v=t5yyc1XfQrs&t=11s>  
<https://www.youtube.com/watch?v=KYPc7CAYJOw&t=1665s>  
<https://www.youtube.com/watch?v=D1v7vX2DMbk&t=468s>  
<https://www.youtube.com/watch?v=KYPc7CAYJOw&t=1678s>  
<https://www.youtube.com/watch?v=dpURgJ4HkMk&t=988s>  
<https://www.youtube.com/watch?v=KYPc7CAYJOw&t=1686s>  
<https://www.youtube.com/watch?v=3GymExBkKjE&t=12415s>  
<https://www.youtube.com/watch?v=K4CGYiQu52s&t=81s>  
<https://www.youtube.com/watch?v=KBo1pyugG88&t=18s>  
[https://www.youtube.com/watch?v=sSL6a\\_ivRVk&t=24s](https://www.youtube.com/watch?v=sSL6a_ivRVk&t=24s)  
<https://www.youtube.com/watch?v=z3ieN39SKQo&t=1s>  
[https://www.youtube.com/watch?v=vJapzH\\_46a8&t=23582s](https://www.youtube.com/watch?v=vJapzH_46a8&t=23582s)  
<https://www.youtube.com/watch?v=T3ugOYTRF7c&t=1379s>  
<https://chatgpt.com/?oai-dm=1>

## 10. ANEXOS

Se han utilizado diferentes herramientas extras para la realización del proyecto:

- Tailorbrands: Página web utilizada para la creación del logo de la aplicación
- Lootie: Página web utilizada para la creación de animaciones. Además en dicha página web existe la posibilidad de descargar las aplicaciones que otros desarrolladores han creado, siendo de forma gratuita.
- BeFunky: Página web dedicada a la edición de imágenes, la cual se ha utilizado para la edición de las imágenes de la aplicación, presenta muchas ventajas, debido a que no es necesario la descarga de aplicaciones como Photoshop o GIMP para la edición de las fotos.
- Stack Overflow: Página web en la que los desarrolladores exponen sus errores en el código para que otros desarrolladores se los puedan solucionar, de esta manera es muy probable que el error en el código que tenga un desarrollador se ha resuelto en esa página por otra persona. Se ha utilizado para solución de errores de compilación o de librerías.
- Logcat: Perteneciente a Android Studio, logcat permite ver los logs, es decir, la consola, debido a que cuando se ejecuta una aplicación es este IDE, no se muestra por consola las operaciones que se van realizando o los errores que van surgiendo.
- Github: GitHub es una plataforma basada en la web utilizada para el control de versiones y la colaboración en proyectos de desarrollo de software. Permite a los desarrolladores almacenar y gestionar sus repositorios de código, rastrear los cambios realizados en el código a lo largo del tiempo y coordinar el trabajo con otros miembros del equipo. Herramienta utilizada para ver código relacionado con la aplicación.