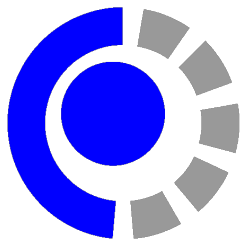


# NOMBRE MATERIA

## TITULO PRACTICO

Manuel Latorre FAI-1931  
manuel.latorre@est.fi.uncoma.edu.ar

Segundo cuatrimestre 2023



**Facultad de Informática**  
UNIVERSIDAD NACIONAL DEL COMAHUE



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. RESTful API	1
1.2. Patrón MVC	2
1.3. Ejecución Otras consideraciones	2
<b>2. Cliente</b>	<b>3</b>
2.1. Patron Flux	4
2.2. Las vistas	4

## Índice de figuras

1. Modelo entidad relación de la base de datos implementada	1
2. Lista de pruebas realizadas a la API con Postman	3
3. Vista de inicio de sesión	5
4. Vista de inicio de sesión con validación	6
5. Vista de registro	7
6. Vista de recetas	8
7. Selección de categorías para filtrar	9
8. Vista de recetas con categorías filtradas	10
9. Vista de favoritos	11
10. Vista de favoritos con una categoría de recetas seleccionada para la cual aun no existen favoritos	12
11. Vista de información de receta	13
12. Vista de calificación de receta	14
13. Vista de información de receta con cajon de ingredientes a mitad de pantalla	15
14. Vista de calificación información de receta con cajon de ingredientes completamente abierto	16
15. Vista de creación de receta	17
16. Vista de creación de receta	18
17. Vista de creación de receta con validaciones	19
18. Vista de creación de receta	20
19. Vista de creación de receta seleccionando la categoría	21
20. Vista de creación de receta definiendo los ingredientes y pasos a seguir	22
21. Vista recetas en donde se ve la nueva receta agregada	23
22. Vista información de la receta creada	24
23. Vista información de la receta creada	25
24. Vista cierre de sesión	26

# 1. Introducción

En el presente informe se desarrollará el proceso de implementación de una aplicación Android con React Native. Dicha aplicación busca la carga y visualización de recetas de cocina, permitiendo a usuarios registrarse e iniciar sesión para poder subir, calificar y marcar como favoritas sus propias recetas. Se hará uso de una RESTful API para, desde el cliente, poder obtener toda la información relativa a las recetas. Las principales tecnologías utilizadas para el desarrollo del cliente son React Native para el desarrollo de la interfaz gráfica, Redux para el manejo de estados y Cloud Storage for Firebase para el almacenamiento y recuperación de imágenes. Las tecnologías utilizadas para la API son JavaScript, Node.js, Express.js y MySQL entre otras.

## 1.1. RESTful API

La RESTful API fue diseñada en primera instancia durante el cursado de la materia Laboratorio de programación utilizando JavaScript, Node.js y Express.js. Para la implementación de este trabajo se realizó una nueva API usando las mismas tecnologías y tomando como base la anterior mencionada con el fin de mejorar aspectos organizativos y funcionales.

Además con el fin de tener información persistente se agregó una base de datos MySQL para el almacenamiento de la información, considerando que las imágenes a utilizar iban a ser cargadas en un almacenamiento en la nube por lo que solo es necesario almacenar el enlace a estas para poder accederlas luego.

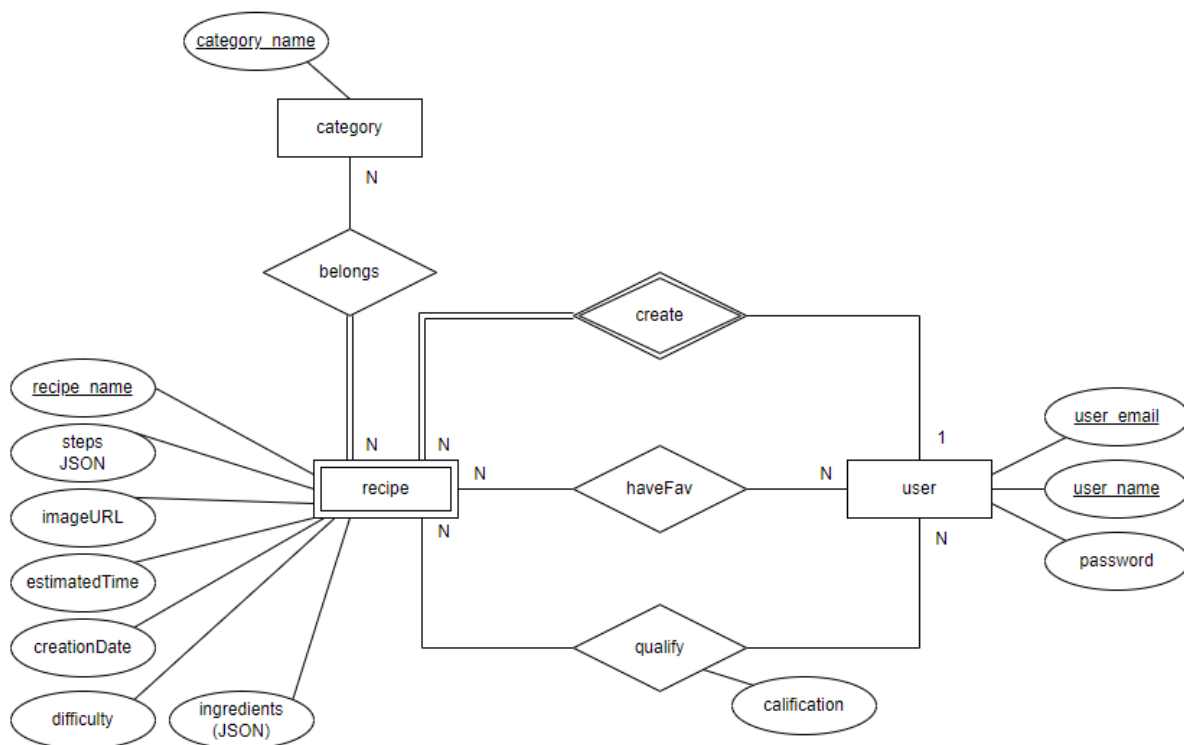


Figura 1: Modelo entidad relación de la base de datos implementada

La implementación cuenta con método de población de la base de datos que puede ser ejecutado con el comando `npm run populate`. Este se encargara de cargar con usuarios, recetas, categorías y favoritos a la base de datos para agilizar su prueba.

## 1.2. Patrón MVC

Como patron de diseño para la implementacion se utilizo Model View Controller (MVC). Este consiste en las siguientes capas:

- **El modelo:** Representa los datos y la lógica de negocio de la aplicación. En el caso de esta implementación se corresponde con la capa de servicios que se encarga de realizar las consultas a la base de datos de forma directa. Puede encontrarse en el directorio `/services`
- **La vista:** Es la capa de presentación de la aplicación, que se encarga de mostrar los datos al usuario. En el contexto de una API RESTful, esta capa se corresponde con las respuestas que envía el servidor a las peticiones realizadas por los clientes. Puede encontrarse en el directorio `/routes`
- **El controlador:** es la capa que se encarga de manejar las peticiones del usuario, interactuar con el modelo para obtener los datos necesarios y devolver la respuesta adecuada. En este caso se utilizaron rutas y controladores que reciben peticiones, validan los datos y solicitan informacion a los servicios correspondientes. Puede encontrarse en los directorios `/controllers` y `/middlewares/validators`

## 1.3. Ejecución Otras consideraciones

Una vez configurado el `.env` de manera adecuada la API puede ponerse en marcha con el comando `npm run dev`.

Otras bibliotecas importantes para la implementación de la api son:

- **Bcrypt:** Para realizar el cifrado de las contraseñas de los usuarios
- **Dotenv:** Para definir variables de entorno y facilitar asi la configuración de la API y la base de datos. Es conveniente revisar dicho archivo para configurar adecuadamente la API e iniciar la ejecución.
- **JSON Web Token (JWT):** como mecanismo de autenticación y autorización con los clientes

Para el realizar pruebas se hizo uso de Postman, en la figura [2] se puede observar la lista de pruebas realizadas las cuales también se corresponden con los diferentes servicios que brinda la API.

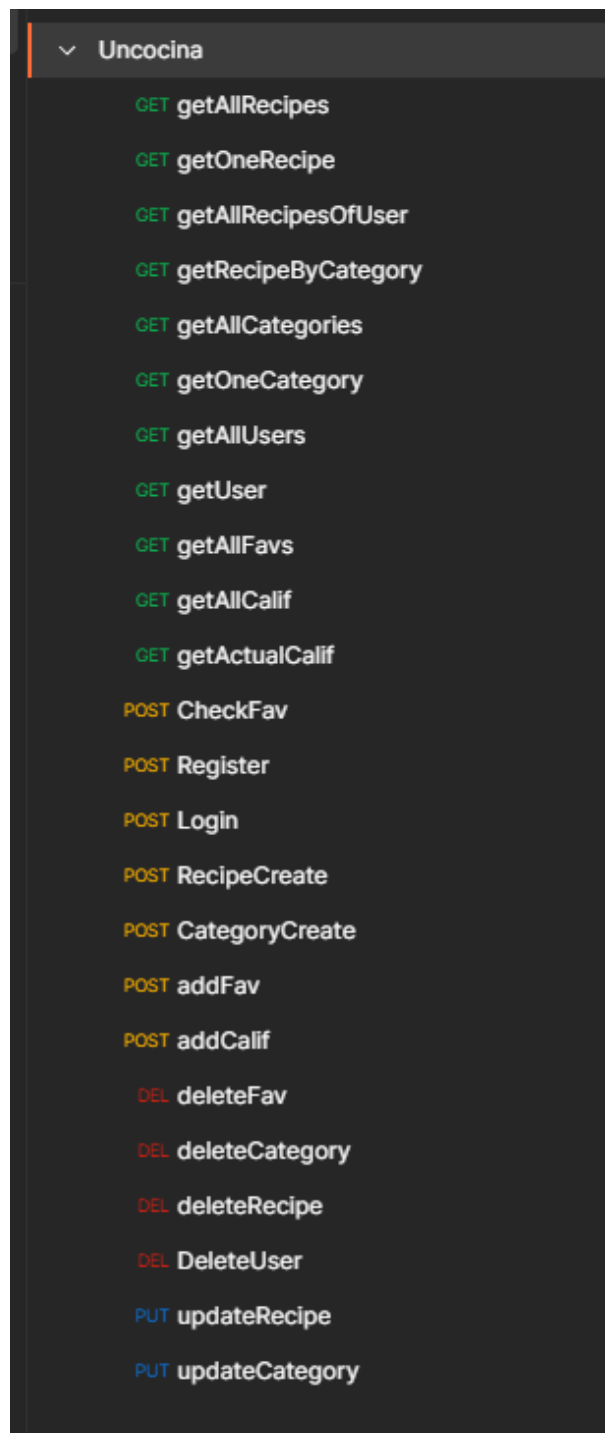


Figura 2: Lista de pruebas realizadas a la API con Postman

## 2. Cliente

El cliente fue implementado con React Native utilizando la herramienta Expo. Este se sirve de información provista por la API a través de peticiones realizadas con Axios.

## 2.1. Patron Flux

Se optó por un patrón Flux para el desarrollo debido a su conveniencia para la implementación de aplicaciones que manejan grandes cantidades de datos o que necesitan ser actualizadas en tiempo real.

El patrón Flux consta de cuatro elementos principales: el Store, el Dispatcher, las Actions y la View. La Store es una especie de base de datos en la que se almacenan los datos de la aplicación (para esto se utilizó Redux Toolkit). El Dispatcher es el encargado de recibir y distribuir las acciones que se realizan en la aplicación. Las Actions son objetos que describen una acción específica que se realizará en la aplicación, y la View es la interfaz de usuario que se muestra al usuario.

El patrón Flux utiliza un flujo de datos unidireccional, lo que significa que la información se mueve en una dirección desde las Actions hasta la View. Cuando un usuario realiza una acción en la aplicación, se crea un objeto Action que describe la acción que se ha realizado. El Dispatcher recibe esta acción y la distribuye a la Store, donde se actualiza el estado de la aplicación. Finalmente, la View se actualiza para reflejar el estado actual de la aplicación.

## 2.2. Las vistas

**Vista de inicio de sesión:** Permite el inicio de sesión por parte de los usuarios con un email y contraseña. Realiza una validación en caso de que falte ingresar algún dato, el email no se encuentre registrado o la contraseña no sea válida. Figura 3 y Figura 4

**Vista de registro:** Permite el registro de usuarios a la plataforma. Realiza el mismo tipo de validaciones que la pantalla de inicio de sesión. Figura 5

**Vista de recetas:** Una vez el usuario se registra o inicia sesión es redireccionado automáticamente a esta vista, aquí se mostrara todas las recetas de la base de datos permitiendo ordenarlas por fecha de publicación y filtrar por categorías. Además como la API soporta paginación las recetas se van cargando a medida que se “scrollea” hacia abajo. Figura 6, Figura 7 y Figura 8

**Vista de favoritos:** Esta vista reutiliza el componente utilizado para la vista de recetas para mostrar las recetas seleccionadas como favoritas por los usuarios. Puede accederse a través del navbar ubicado en la parte baja de la pantalla. Figura 9 y Figura 10

**Vista información de receta:** Si en las vistas de recetas o favoritos se presiona sobre alguna de las recetas se llegará a la vista de información de dicha receta. Aquí se pueden ver listados los pasos de la receta, ingredientes, duración, imagen, se puede realizar una calificación sobre la receta y agregarla a favoritos. Además se utilizó el componente externo “@gorhom/bottom-sheet” para poner los ingredientes en un cajón deslizable. Figura 11, Figura 12, Figura 13 y Figura 14

**Vista de creación de receta:** En esta vista se solicitarán todos los datos necesarios para crear una nueva receta. Se realizarán las validaciones correspondientes y una vez cargada la receta exitosamente se redireccionará a la vista de recetas donde se podrá ver publicada. Figuras [15],[16],[17],[18],[19],[20],[21],[22],[23]

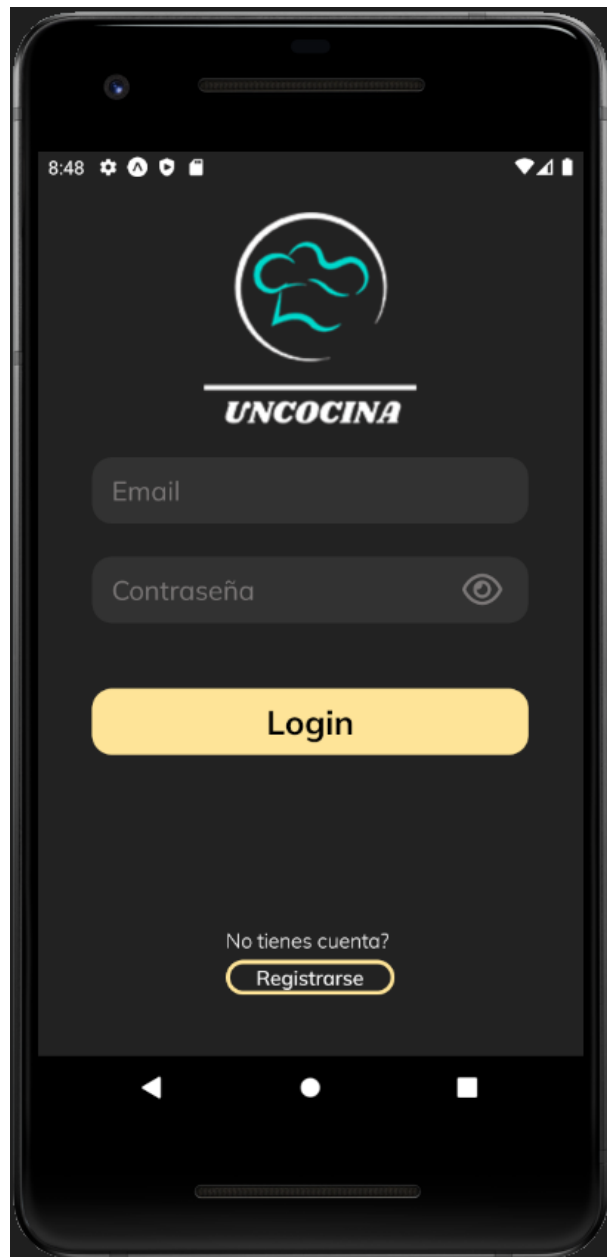


Figura 3: Vista de inicio de sesión

**Vista de cierre de sesión:** Finalmente se cuenta con una vista para cerrar la sesión, al hacerlo redirecciona automáticamente a la vista de inicio de sesión. Figura 24

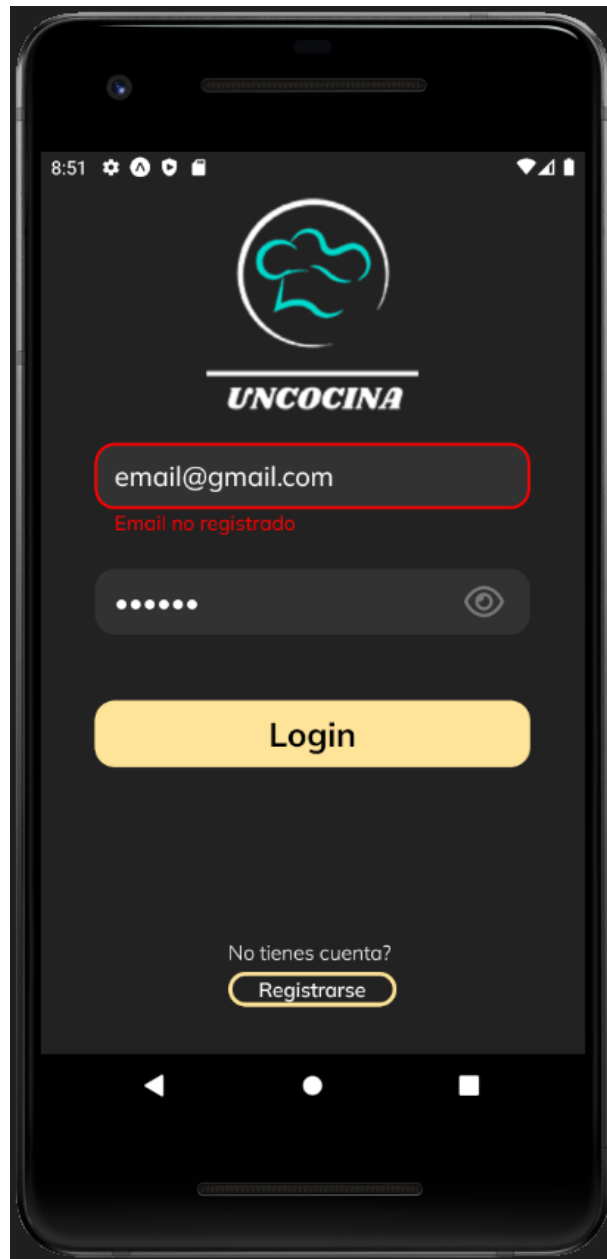


Figura 4: Vista de inicio de sesión con validación



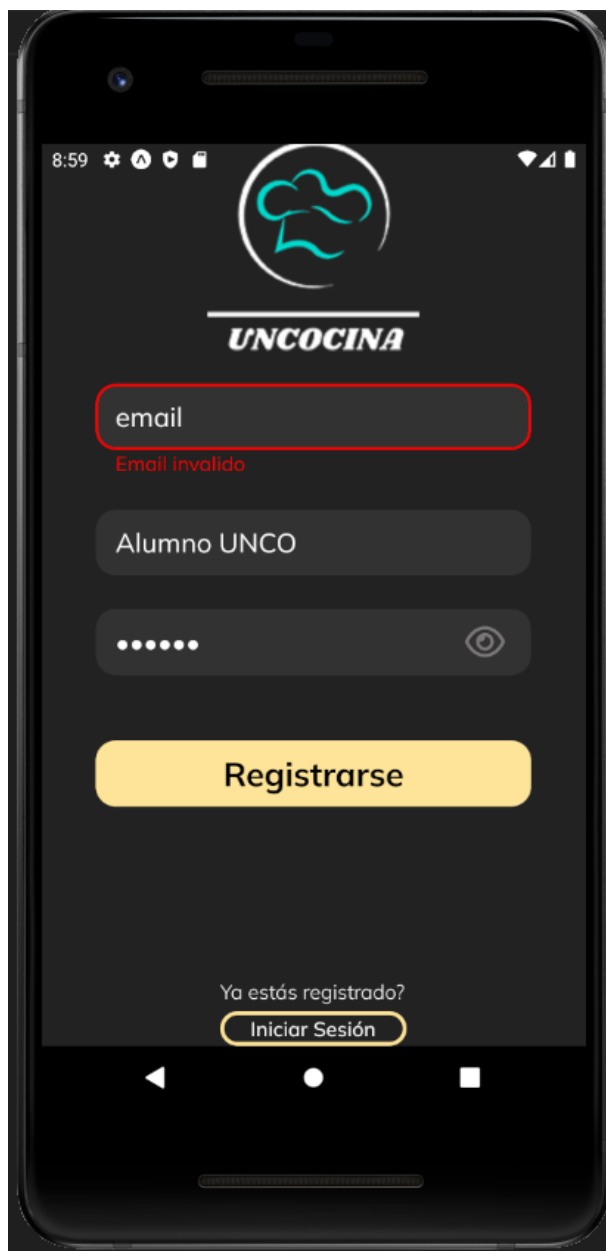


Figura 5: Vista de registro

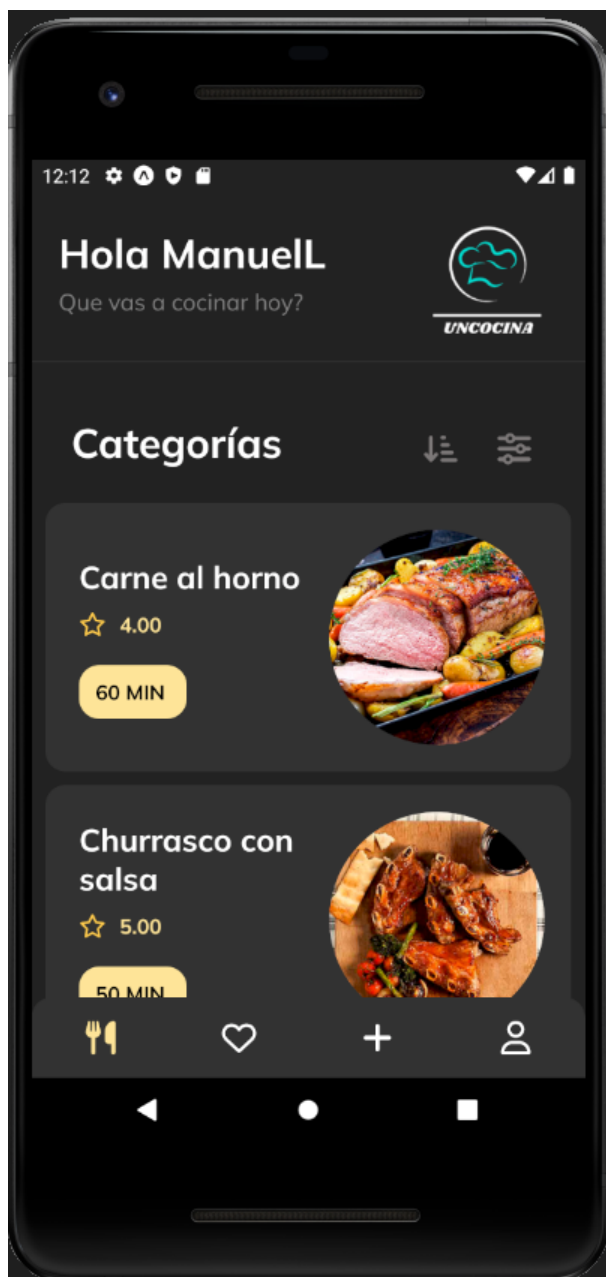


Figura 6: Vista de recetas



Figura 7: Selección de categorías para filtrar

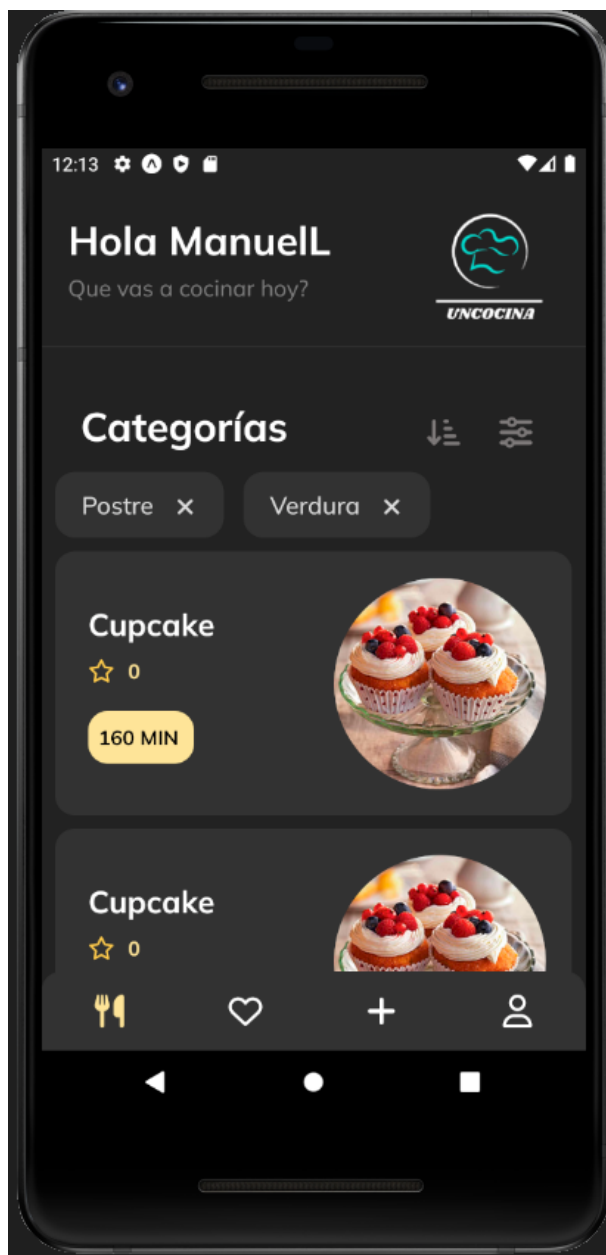


Figura 8: Vista de recetas con categorías filtradas

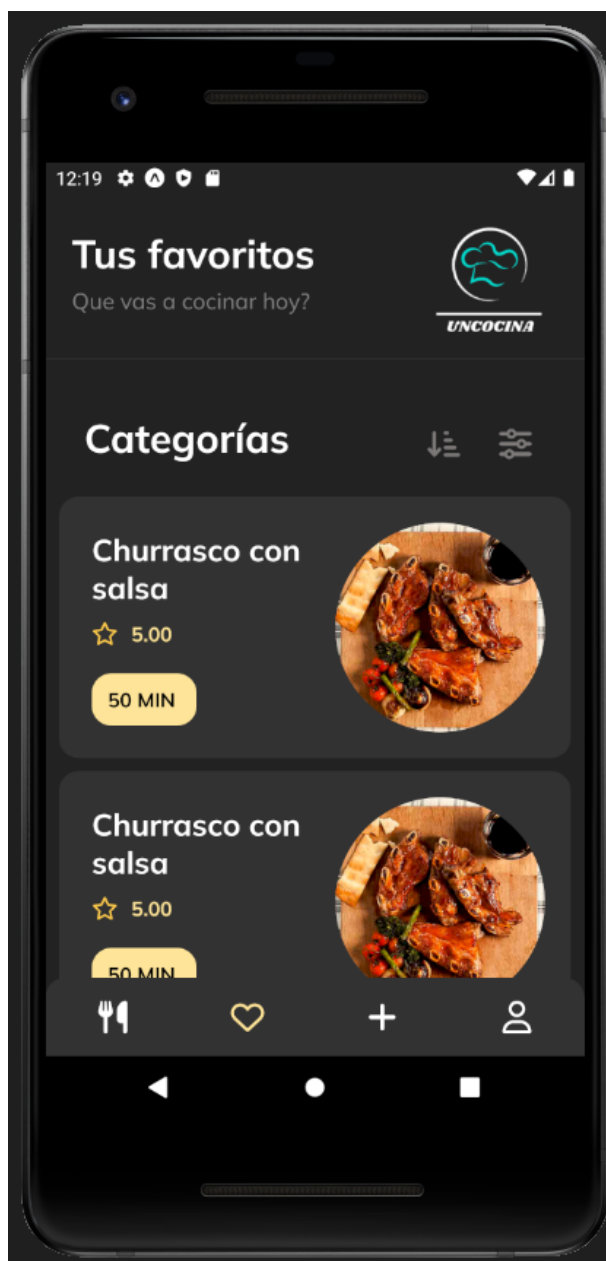


Figura 9: Vista de favoritos



Figura 10: Vista de favoritos con una categoría de recetas seleccionada para la cual aun no existen favoritos

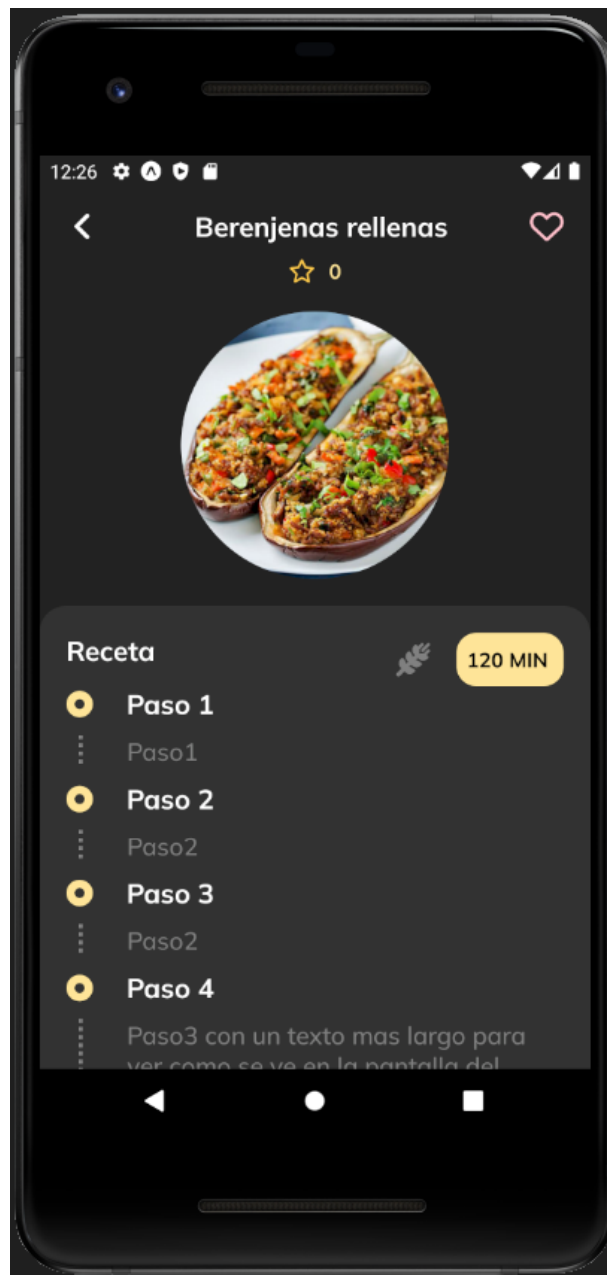


Figura 11: Vista de información de receta

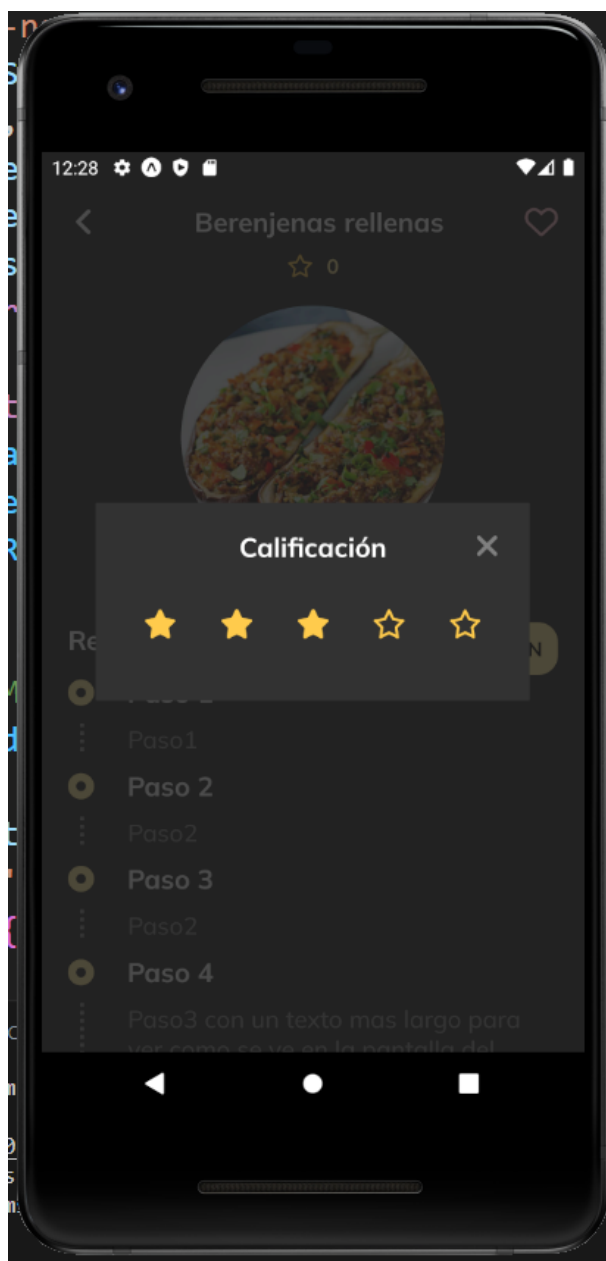


Figura 12: Vista de calificación de receta



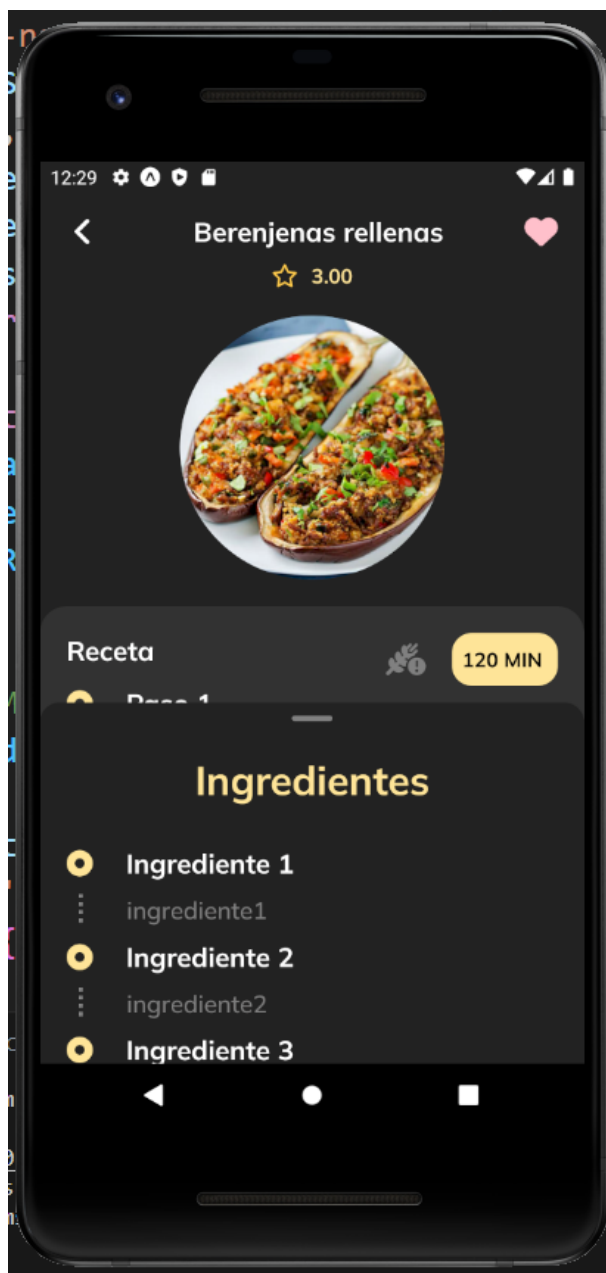


Figura 13: Vista de información de receta con cajon de ingredientes a mitad de pantalla



Figura 14: Vista de calificación información de receta con cajon de ingredientes completamente abierto



Figura 15: Vista de creación de receta



12:39

Categoría

Dificultad

Ingredientes

Ingrediente 1

+

Pasos

Paso 1

+

Publicar

Figura 16: Vista de creación de receta



Figura 17: Vista de creación de receta con validaciones

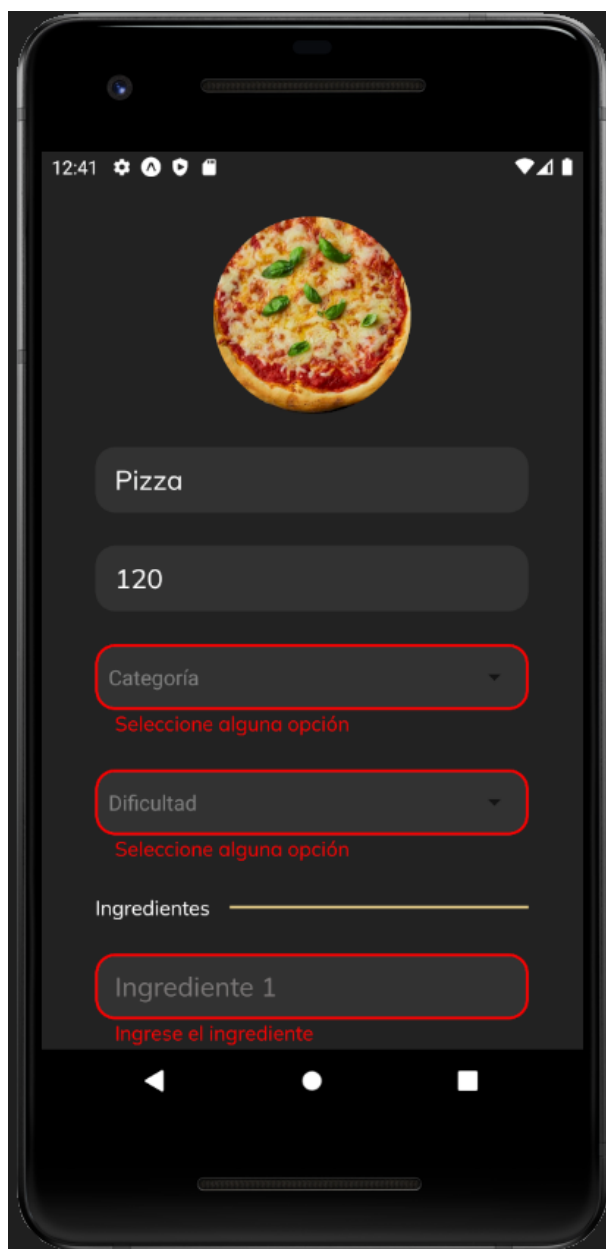


Figura 18: Vista de creación de receta

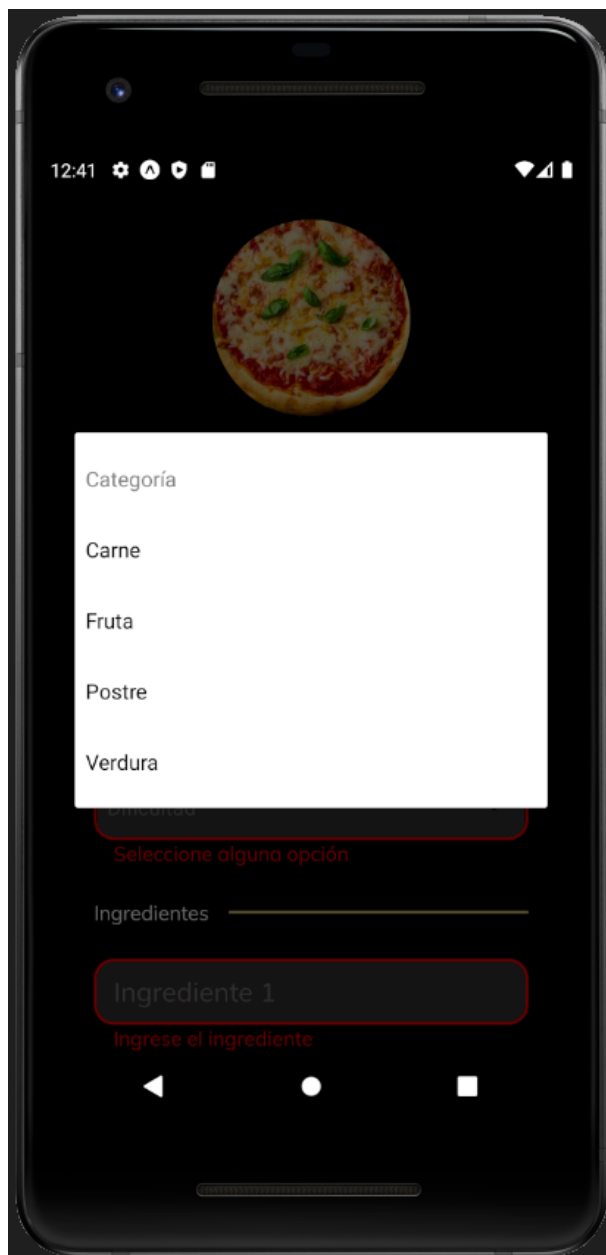


Figura 19: Vista de creación de receta seleccionando la categoría

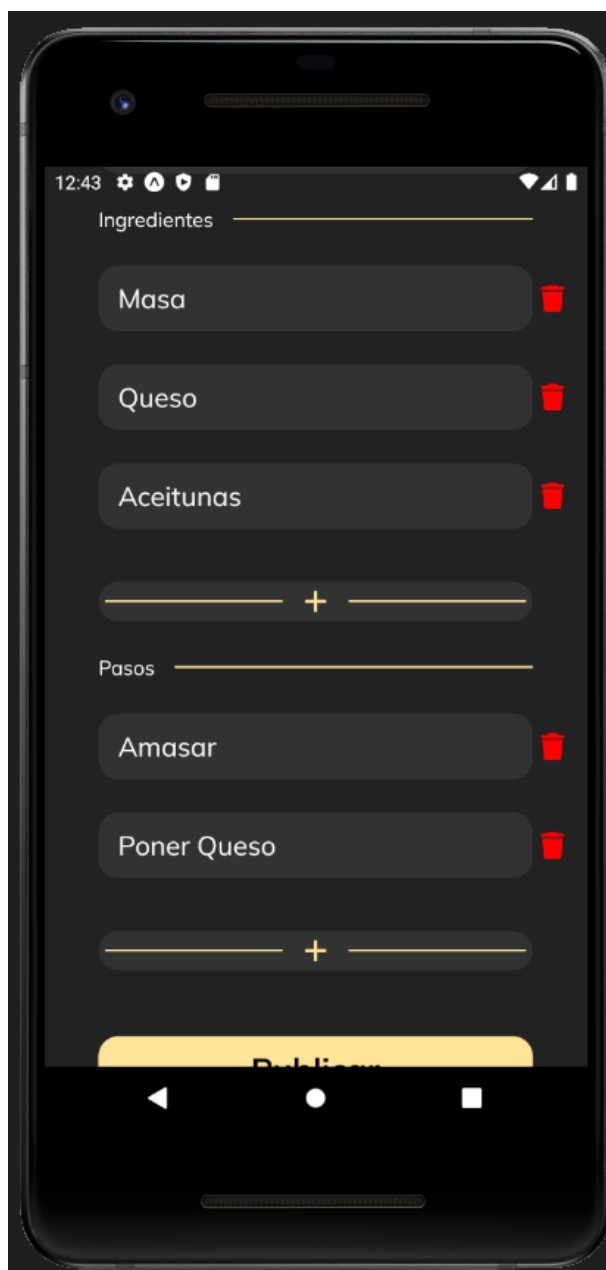


Figura 20: Vista de creación de receta definiendo los ingredientes y pasos a seguir



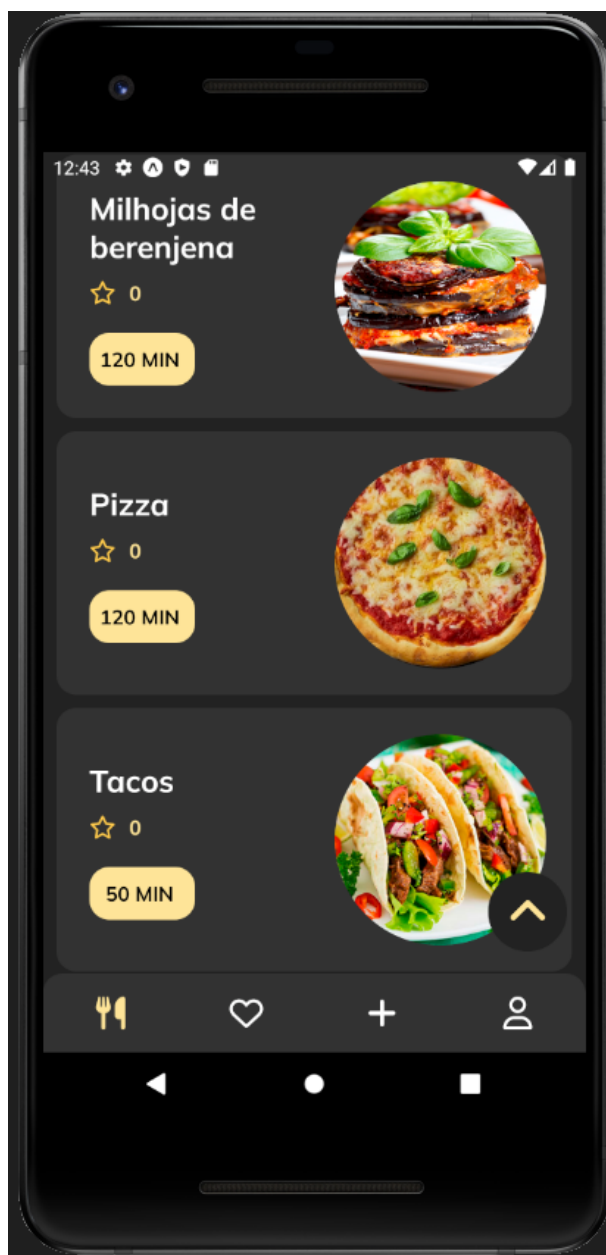


Figura 21: Vista recetas en donde se ve la nueva receta agregada

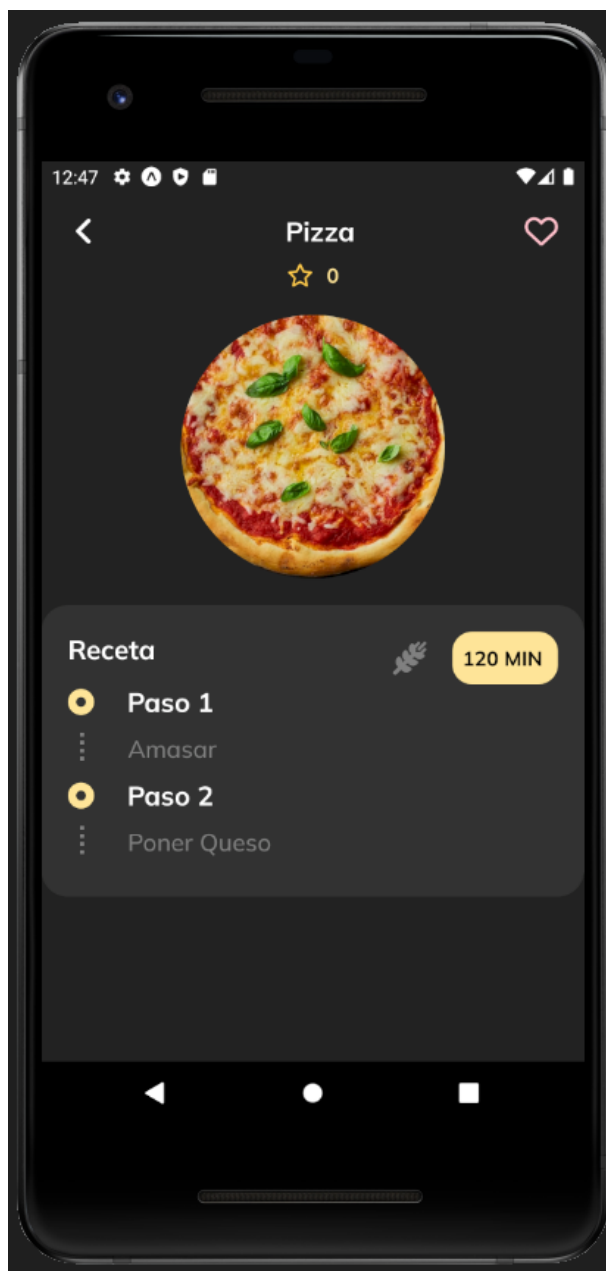


Figura 22: Vista información de la receta creada

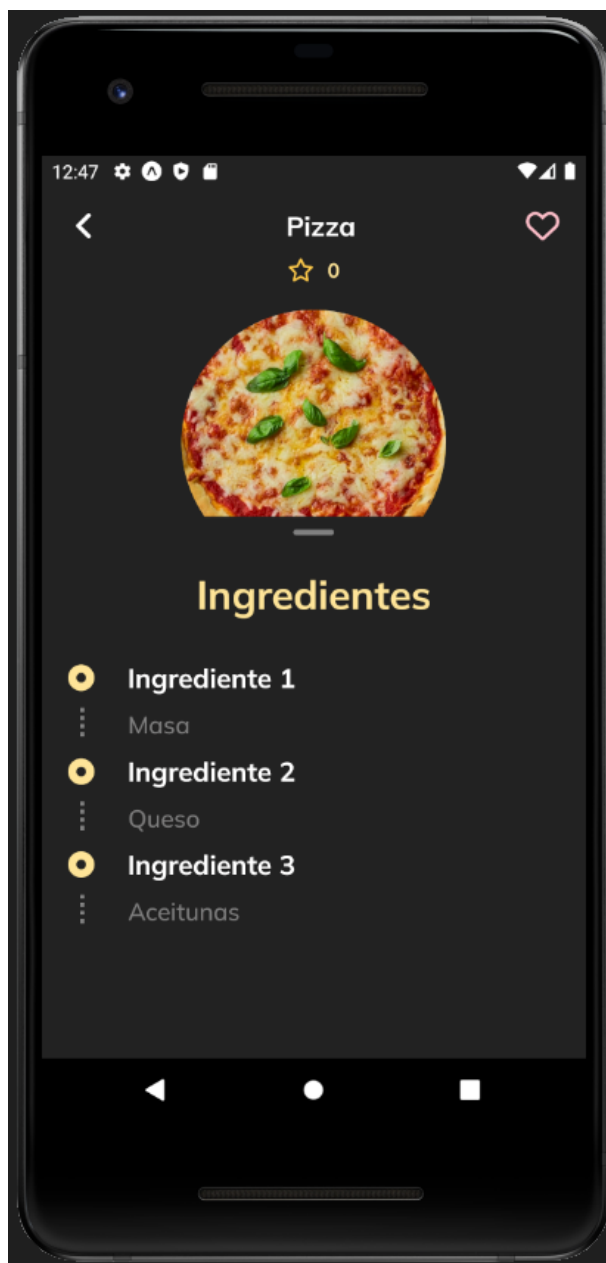


Figura 23: Vista información de la receta creada



Figura 24: Vista cierre de sesión