TEC | Tecnológico de Costa Rica

Documentación Tarea Programada 2:
Recetas Bombur

IC-4700

Lenguajes de Programación

Profesor:
Andrei Fuentes Leiva

Estudiantes:

- Mauricio Quirós Bravo - 2017083146
- Paul Villafuerte - 2017142374
- Jose Pablo Feng - 2017105644

Fecha de entrega:
7 de junio de 2019

Índice

1. Introducción

El propósito del presente documento es el de especificar el planeamiento y desarrollo del software *Recetas Bombur,* realizado para la segunda tarea programada del curso de Lenguajes de programación IC-4700, por los estudiantes Mauricio Quirós, Paul Villafuerte y José Pablo Feng; el primer semestre de 2019. Asimismo, se detalla un manual técnico y manual de usuario final para los aspectos mas relevantes del sistema.

El sistema desarrollado se trata de una aplicación móvil para Android que le permite al usuario crear y compartir sus recetas favoritas, así como tener la oportunidad de conocer otras recetas publicadas por otros usuarios.

2. Planificación y calendario

Para mantener un control del avance, además de tener una manera de planificar los tiempos delegados para implementar cada característica del proyecto, se utilizó un calendario tipo Gantt, al que todos los miembros del equipo podían tener acceso de consulta y modificación.

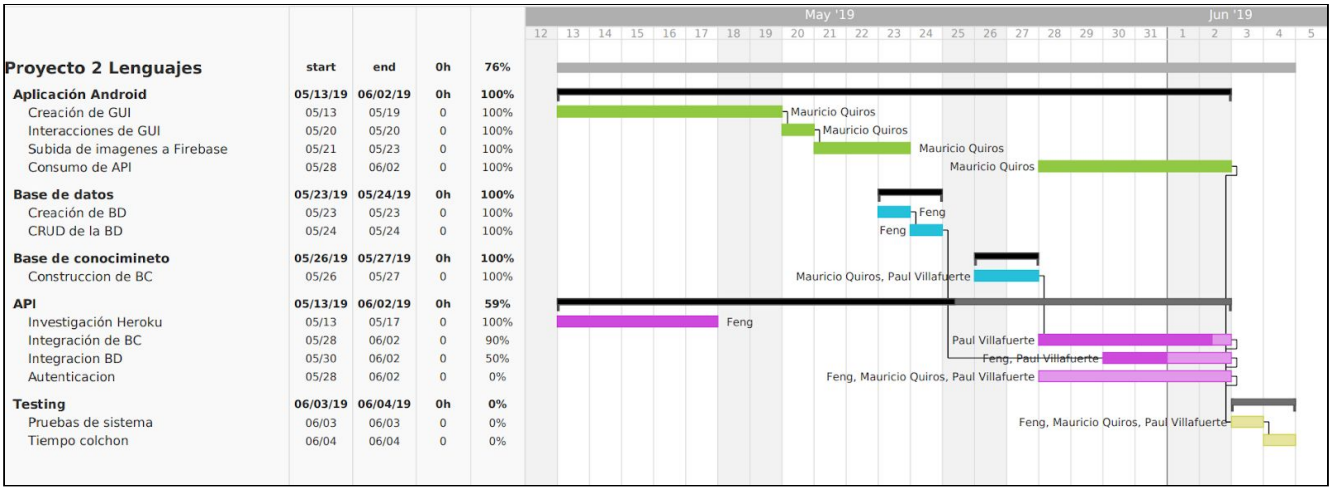


Figura 0: Calendario Gantt del proyecto

Para mantener un control de las tareas que se encontraban por ser realizadas se trabajó a lo interno con un sistema de SCRUM no documentado.

En el transcurso de las semanas de desarrollo se calendarizan varias reuniones informales tanto físicas como virtuales, donde se estipulaba que todos los miembros

del trabajo asistieron de manera obligatoria. Todas las reuniones se llevaron a cabo de acuerdo a lo planeado y todos los puntos fueron tratados con éxito.

En cuanto a la organización y repartición de trabajo se estipularon tres áreas de desarrollo, el desarrollo del app, el desarrollo del backend y el desarrollo y despliegue de la base de datos. El desarrollo de la aplicación móvil estuvo principalmente dirigida por Mauricio Quiros. El desarrollo del backend fue principalmente implementado por Paul Villafuerte. La creación y despliegue de la base de datos estuvo a cargo de José Pablo Feng.

Todos los miembros del equipo estuvieron involucrados en la investigación e implementación de los temas desconocidos como lo eran el despliegue en *Heroku*, la autenticación por medio de tokens y los conceptos básicos de la creación de un backend con *Flask*.

3. Desarrollo del sistema

En esta sección se detallarán los aspectos técnicos mas relevantes para la implementación del proyecto.

3.1- Vision general

El sistema está compuesto por cuatro secciones, la aplicación Android; utilizada como medio para que el usuario se comunique con el sistema, backend con Python + Flask; utilizado como un API para recibir las consultas de los clientes, a la par de un mecanismo de consulta de Prolog donde se almacena y consulta la información de las recetas, y una base de datos; cuyo propósito es el de almacenar la información relevante a los usuarios además de procesar la lógica de inicio de sesión.

3.2- Arquitectura de la aplicación

La aplicación se estructura mediante una arquitectura muy común para los sistemas Android llamada *Clean + MVPI*. Esta, resumidamente, se trata de partir de un diagrama de clases y encontrar las relaciones y entidades que engloban comportamientos comunes y transferirlos a paquetes en Android. Una vez que se llega a la estructura deseada, cada entidad tendrá su propia sub-arquitectura *MVPI*, donde:

- El **modelo** es una capa utilizada para abstraer los objetos relevantes.
- La **vista** se conforma del layout y de su actividad controladora, y es la encargada de recibir todos los inputs de usuario y su validación

- El **presentador** es el encargado de recibir la información de la vista y pasarla al interactor.
- El **interactor** se encarga de realizar la implementación de la lógica. Si fuera necesario, este se conecta con una capa extra que maneje la persistencia de los datos necesarios.

Es importante destacar que cada una de las capas se debe implementar utilizando interfaces para mantener una alta cohesión y un bajo acoplamiento. De esta manera, se crea la arquitectura cerrada implementada en la aplicación

3.3- Restricciones de diseño

La aplicación será desarrollada específicamente utilizando Android con un sdk 21 como mínimo y utilizando Java. El backend será implementado utilizando Python + Flask. La base de datos será construida utilizando PostgreSQL. Se utilizará Heroku como plataforma de despliegue para el backend.

3.4- Librerías y dependencias

Las siguientes son las librerías y dependencias necesarias para el correcto funcionamiento del sistema:

- **Android**
  - Volley
  - Gson
  - Glide
  - Firebase core
  - Firebase storage
  - PinchToZoom

- **Backend**
  - Flask
  - Pyswip
  - SWI-Prolog compiler
  - Gunicorn
  - Itsdangerous
  - Jinja2
  - Markupsafe
  - Werkzeug
  - Psycopg2

3.5- Funcionalidades del sistema

El sistema deberá ser capaz de realizar las siguientes funcionalidades:

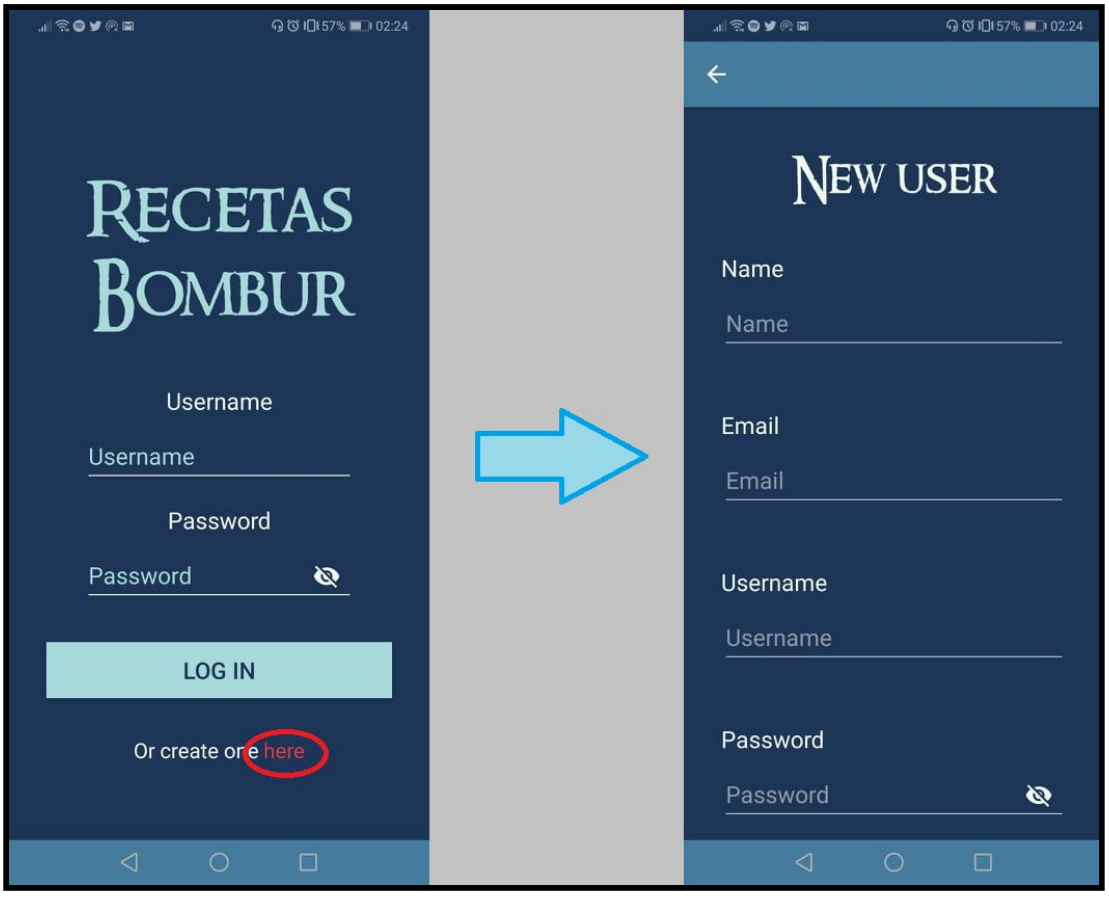
- Creación de usuarios y su registro en la base de datos
- Capacidad de iniciar y salir de una sesión
- Creación de recetas con todas sus características, incluida la subida de imágenes
- Visualización de todas las recetas creadas por los usuarios
- Búsqueda de recetas por nombre, tipo o ingredientes
- Consultar el detalle de una receta

4. Manual de usuario final

En esta sección se detallan y se ilustran con mayor profundidad las funcionalidades del sistema enunciadas en el punto *3.5*.

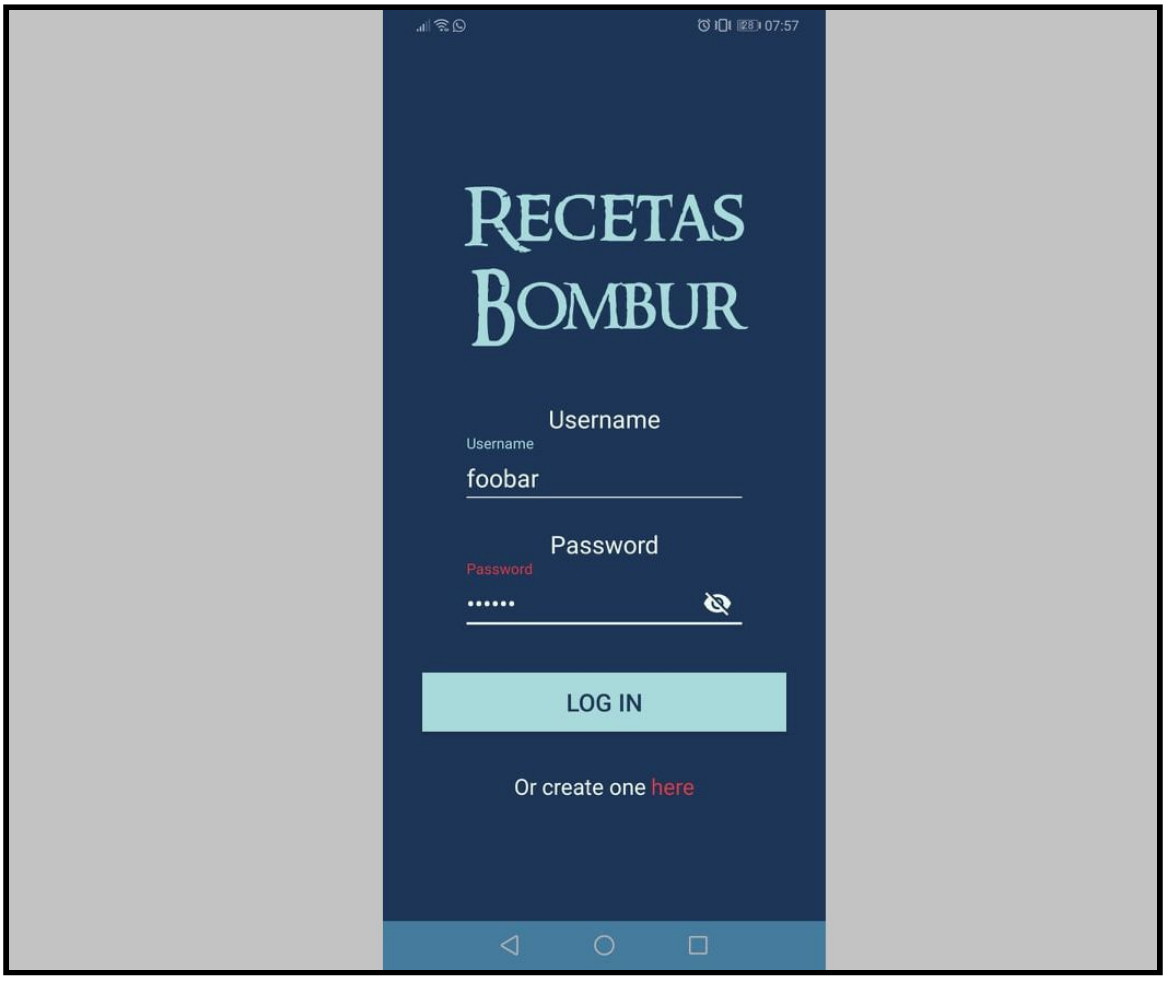
4.1- Creación de usuarios

Desde la pantalla de inicio de sesión, al iniciar la aplicación, se muestra un texto resaltado que indica *o cree una aquí*. Si cliqueamos este texto se nos muestra la pantalla de la creación, donde se llena la información solicitada y se confirma para crear su nuevo usuario.



4.2- Inicio y cierre de sesión

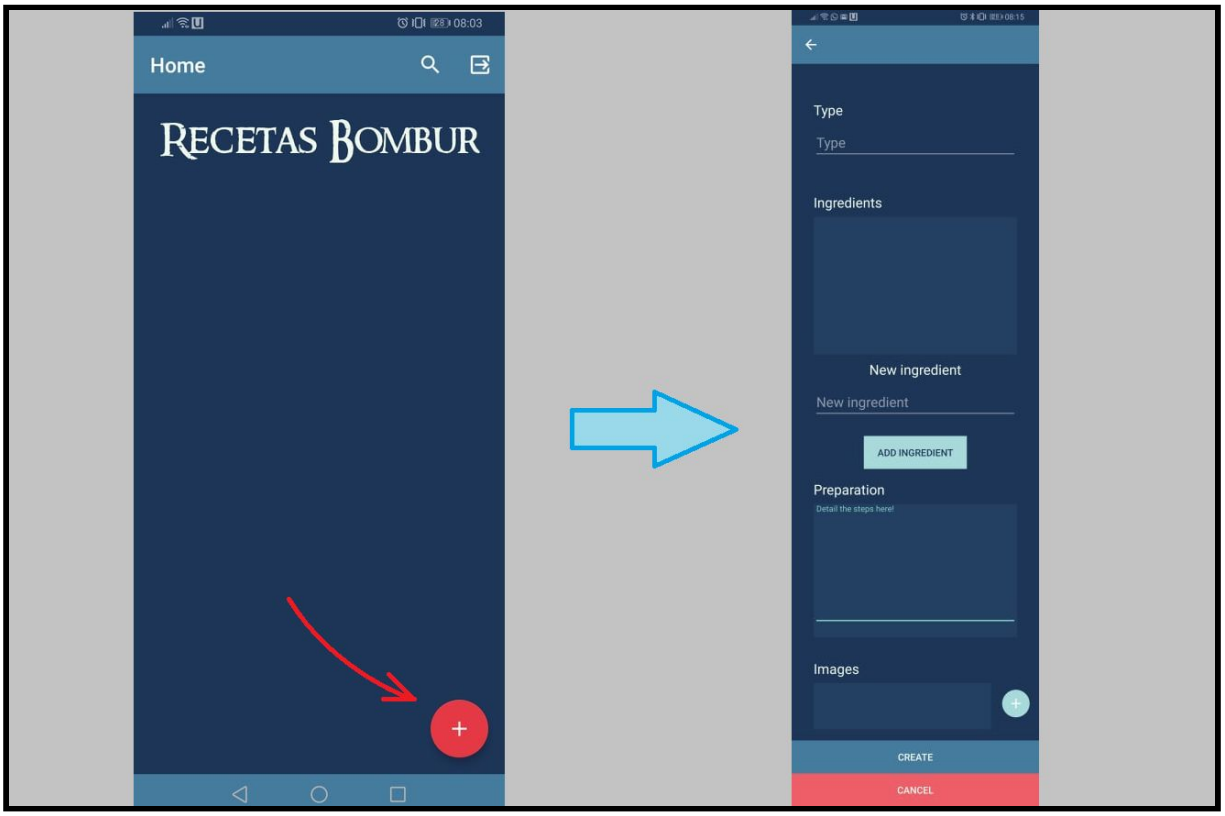
Una vez que se tenga una cuenta creada, podemos utilizar las nuevas credenciales para iniciar sesión en el sistema. Para esto, desde la página de inicio de sesión se ingresa el nombre de usuario y la contraseña. Una vez que hayan sido ingresadas, entramos con el botón de Log in.



Cuando se quiera cerrar una sesión, basta con presionar el icono de *Log out* encontrado en la barra superior en la pantalla principal.

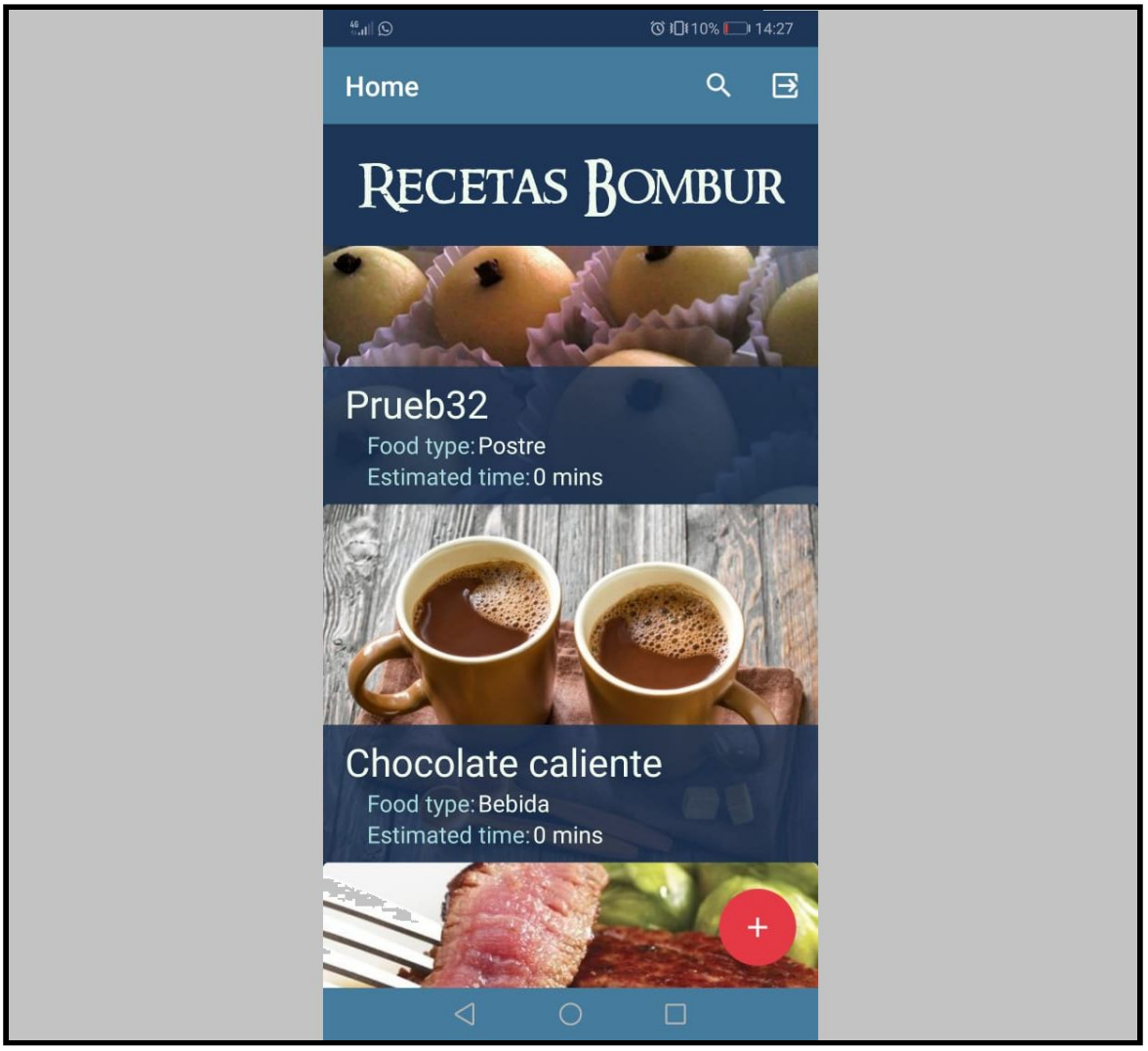
4.3- Creación de recetas

Para crear una nueva receta, desde el menú principal, cliqueamos el botón con símbolo + lo que nos lleva a una nueva pantalla que nos solicita llenar la información de la receta. Una vez que esta información sea ingresada, hacemos click en el botón *Crear receta*.



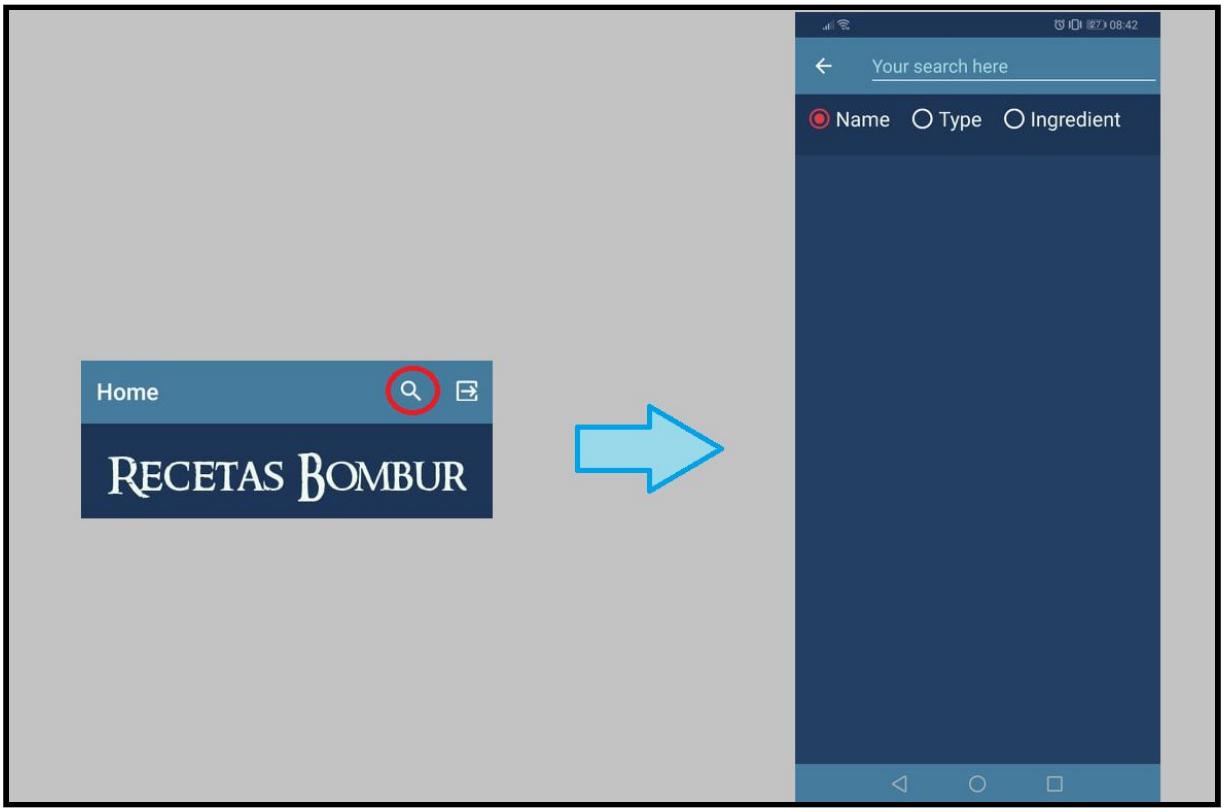
4.4- Visualización de todas las recetas

Para ver todas las recetas de la aplicación basta con iniciar sesión y una lista con todas las recetas será mostrada en la página de inicio.



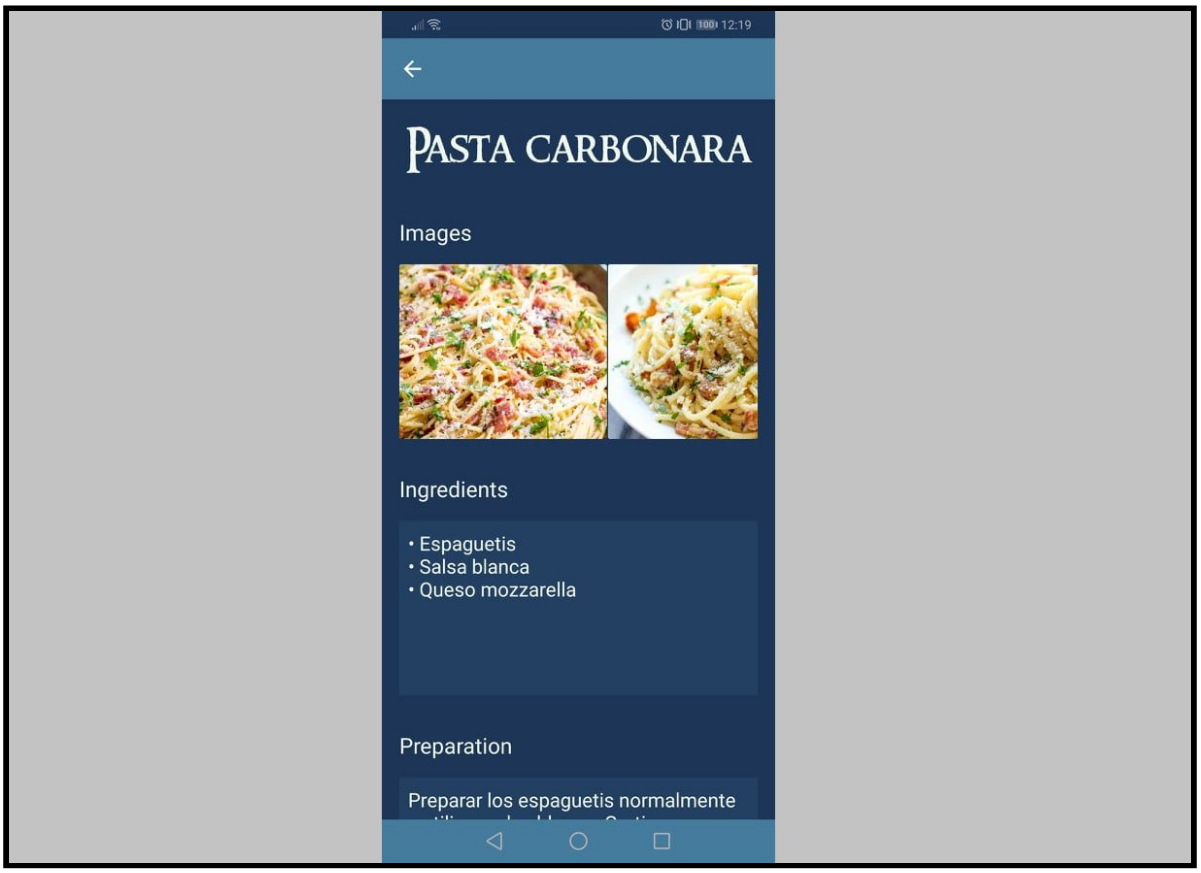
4.5- Búsqueda de recetas

La aplicación ofrece la posibilidad de encontrar recetas por medio de una búsqueda textual, por categorías las de *Nombre*, *Tipo* e *Ingrediente*. Para realizar una búsqueda debemos, desde el menú principal, hacer click en el icono de *Buscar* y esto nos llevará a la pantalla de búsqueda, donde seleccionaremos el tipo de búsqueda e introduciremos el parámetro de texto para la búsqueda.



4.6- Detalle de una receta

Para ver toda la información de una receta, basta con cliquear la imagen de una receta desde la página de inicio o desde la pagina de busqueda despues de encontrar un resultado. Esto nos lleva a una nueva pantalla donde se encuentran todas la imágenes de la receta, sus ingredientes y preparación.



Referencias

- Procfile:
  - https://devcenter.heroku.com/articles/procfile

- Requirements.txt:
  - https://devcenter.heroku.com/articles/python-pip

- pyswip consult:
  - https://github.com/yuce/pyswip

- heroku plugins:install heroku-repo:
  - https://www.xavierriley.co.u/compiling-packages-from-source-on-heroku-using-buildpacks/

  - https://devcenter.heroku.com/articles/heroku-postgresql#heroku-postgres-ssl

- Tutorial Python and PostgreSQL:
  - https://www.youtube.com/watch?v=2PDkXviEMD0

- Column encryption:
  - https://dbtut.com/index.php/2018/10/01/column-level-encryption-with-pgcrypto-on-postgresql/
  - https://www.tutorialspoint.com/postgresql/postgresql_java.htm

- pgAdminInfo:
  - https://medium.com/@vapurrmaid/getting-started-with-heroku-postgres-and-pgadmin-run-on-part-2-90d9499ed8fb

- StoredProcs in pgAdmin4:
  - http://www.postgresqltutorial.com/postgresql-create-function/

  - http://www.postgresqltutorial.com/postgresql-jdbc/call-postgresql-stored-function/

  - http://initd.org/psycopg/docs/cursor.html

  - https://jwt.io/introduction/

- Random string generator:
    - https://pynative.com/python-generate-random-string/

- Android HTTP requests with Volley:
    - https://developer.android.com/training/volley/simple.html#java

- PySwip:
    - https://github.com/yuce/pyswip

- Python + Flask
    - https://www.youtube.com/watch?v=s_ht4AKnWZg

    - https://www.youtube.com/playlist?list=PLXmMXHVSvS-AFMUmbBeIfL3PqTvgw8ygb