

CICLO FORMATIVO DE GRADO SUPERIOR

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROYECTO DE FIN DE GRADO



[KuEat Food Delivery]

Curso: 2021/22

AUTOR: [Fouad Horri Ahcynat]

Introducción

Motivación y objetivos

La principal motivación para realizar esta aplicación, ha sido debido a que un familiar, que vive en mi país (Marruecos), me comento que un amigo suyo que tiene un negocio de comida a domicilio, esta interesado en que alguna compañía le haga una pequeña aplicación en la que se puedan mostrar todas las comidas con las que cuenta la empresa y mostrar todos los productos que esta

compañía ofrece, para que así los clientes se puedan descargar esta aplicación y puedan ver todas las comidas que ofrecen.

Cuando mi familiar me lo comento, se me ocurrió la idea de realizar una aplicación de envíos de comida a domicilio, lo que es conocido actualmente como una app de Food Delivery, por lo tanto comencé a buscar que tecnologías podrían ser interesantes para desarrollar una app de este estilo, y por lo tanto estuve pensando entre React Native o Flutter, y finalmente me decante por Flutter, ya que esta tecnología me interesaba mucho y siempre me ha gustado el desarrollo para IOS/Android.

Además, a este app pensé en integrarle categorías de comidas, integrar categorías de bebidas entre otras muchas, por lo tanto, tendremos un Backend que se encargara de todo ello, crear nuevas comidas/bebidas/categorías.

Esta app integrara un sistema de carrito, para que podamos ir añadiendo productos a nuestro pedido.

La app contara con un diseño muy amigable para el usuario y bastante sencilla de utilizar.

Entre mis principales objetivos, esta el aprender a utilizar Flutter y su lenguaje de programación Dart, ya que me parece un Framework muy llamativo y que deja resultados muy llamativos y limpios en cuanto al diseño de aplicaciones móviles.

Descripción de la aplicación

Se trata de una aplicación para Android/IOS, que funciona como una aplicación de envíos y reparto de comida a domicilio, con esta app también vamos a poder consultar los ingredientes de las comidas, vamos a poder ver la forma en la que se realizan estas comidas que ofrece la empresa detrás del envío de dicha comida.

La aplicación también nos permitirá escanear comidas, ya sea introduciendo imágenes desde nuestra galería o abriendo la cámara para poder hacerle una foto a alguna comida, y así poder saber donde comer dicha comida. Al igual que consultar los ingredientes y forma de hacer dicha comida.

Especificación de requisitos

Requisitos funcionales

- En cuanto a los requisitos funcionales, podremos ver que la aplicación al iniciarse, se conectará a internet, ya sea vía datos móviles o por Wifi, al iniciarse por primera vez la aplicación, nos solicitará que iniciemos sesión con nuestro correo y contraseña, esa será la primera pantalla que aparecerá en la aplicación.
- Para el correcto funcionamiento de la aplicación, vamos a necesitar de dar permisos de acceso a la ubicación, permisos de acceso al almacenamiento local, a la gestión de la cámara, entre otros permisos.

- Una vez que hayamos entrado por primera vez a la aplicación, vamos a poder consultar las distintas comidas que se encuentran en la pantalla principal, vamos a poder navegar entre las comidas populares, y comidas recomendadas, al pulsar sobre alguna de estas comidas vamos a poder acceder a los detalles de esta comida. Vamos a poder consultar el precio de la comida, una descripción de dicha comida, lugar del que proviene dicha comida.
- De cada comida vamos a poder añadir una cantidad de dichas, por lo tanto vamos a contar con un carrito con funcionalidad completa, vamos a poder añadir distintas comidas al carrito, donde se ira sumando el total de la compra a pedir, también se irán añadiendo las distintas comidas que vamos seleccionando conforme vamos navegando por las comidas recomendadas y popular, cuando el usuario haya decidido el pedido que va a comprar, puede confirmar el pedido, una vez confirmado pasara a la pantalla de inicio de sesión.
- En la pantalla de inicio de sesión, el usuario va a poder crear una nueva cuenta dentro de la app, o también puede iniciar sesión, con una cuenta existente con la que se haya registrado con anterioridad.
- Para el registro del usuario vamos a solicitar el numero de teléfono, correo electrónico, nombre, y por ultimo contraseña, todos estos campos estarán validados para corregir errores. No se podrán registrar dos usuarios con el mismo correo electrónico, con el mismo numero de teléfono.
- El usuario va a poder dirigirse a la pantalla de usuario, donde podrá consultar su información, ya sea nuestro numero de teléfono registrado dentro de la app, nuestro nombre y correo electrónico dentro de la app.
- Por otro lugar, dentro del apartado de usuario vamos a poder encontrar un apartado de dirección, el usuario al pulsar sobre dicha sección, va a poder entrar a Google Maps, para poder buscar su dirección, y así poder seleccionar dicha dirección como dirección de entrega, también va a poder seleccionar otras direcciones alternativas, como lugar de trabajo o hogar, en el apartado de Google Maps, gracias a la API de Google Maps, podemos indexar las direcciones y así poder seleccionar la dirección que nosotros deseemos.
- Contaremos con un historial de pedidos, ya que cada usuario al confirmar una compra de un pedido, dicho pedido se le guardara dentro de su usuario, y así siempre va a poder consultar el numero de pedidos que ha realizado, el tipo de pedido que ha realizado, y la cantidad de comida que ha realizado, con su fecha y hora.
- En la pantalla de home, el usuario contara con un botón para poder acceder al escáner de comidas, cuando accede a dicha pantalla va a poder subir una imagen desde su galería o realizar una foto con su cámara a alguna de las comidas que desea de comer o consultar sus ingredientes y descripción , por lo tanto el usuario subirá una imagen o hará una foto, y con ayuda de inteligencia artificial va a poder saber que tipo de comida es y por norma general los ingredientes de dicha comida, y se le recomendará un lugar donde puede ir a comer dicha comida.

Requisitos no funcionales

- **ARQUITECTURA**

En cuanto a los requisitos que debe de cumplir el dispositivo para ejecutar la aplicación, serán los siguientes,

el dispositivo debe de contar con al menos 4GB de memoria interna, para poder instalar correctamente la aplicación y todas sus dependencias.

La aplicación es compatible con los sistemas Android entre el SDK 12 y la 24 que es la última versión del SDK de Android, la versión del sistema operativo Android será la 9.0 a partir de esta se podrá ejecutar la aplicación, ya que la versión 9.0 es la versión mínima de sistema operativo para poder ejecutar la app.

El dispositivo debe de contar con al menos 2GB de Memoria RAM para poder ejecutar la aplicación correctamente y que esta funcione de manera correcta sin trabas.

La aplicación no se podrá ejecutar en dispositivos IOS.

El dispositivo debe de contar con un procesador Quad Core a 1.2GHz como mínimo para correr la aplicación.

El dispositivo debe de contar con servicios de localización GPS para poder utilizar los mapas dentro de la aplicación.

Para el correcto funcionamiento de la aplicación, el dispositivo debe de contar con cámara.

- **SEGURIDAD**

En el dispositivo vamos a tener todos los datos del usuario cifrados, para la seguridad del usuario.

La aplicación estará protegida contra accesos no autorizados. Todos los datos dentro de la aplicación, ya sea el historial de pedidos, la contraseña del usuario, todo ello está cifrado y cuenta con una alta seguridad, para evitar el robo de datos.

- **CONECTIVIDAD**

El dispositivo debe de contar con conexión a internet en el primer inicio, ya que es necesario para poder cargar los datos de la base de datos del servidor.

El dispositivo debe de dar permisos de acceso al almacenamiento local, también debemos de tener permisos para poder acceder al control de GPS y mapas.

- **ORGANIZACION DE CLASES**

Las clases de la aplicación están organizadas según su función por medio de paquetes en la estructura del proyecto.

- **RESOLUCIONES DE PANTALLA**

La aplicación está diseñada para poder ejecutarse en resoluciones de pantalla de 1920x1080,

pantallas HD. La app, se adapta al tipo de pantalla del dispositivo pero para una mejor experiencia del usuario se recomiendan pantallas en HD.

- **OTROS**

La aplicación móvil informará al usuario de los posibles errores que se puedan producir durante el uso de la aplicación.

La aplicación proporcionará un buen manejo en cuanto al acceso concurrente de los usuarios.

La aplicación debe de tener las notificaciones activadas, ya que se recibirán noticias en forma de notificaciones push, por lo tanto debe de tener las notificaciones activadas.

Wireframe e Interfaz Gráfico

- Incluiré un boceto de la apariencia que tendrá la aplicación, describiré brevemente los distintos ítems que aparecen en la pantalla:

ICONO DE LA APP



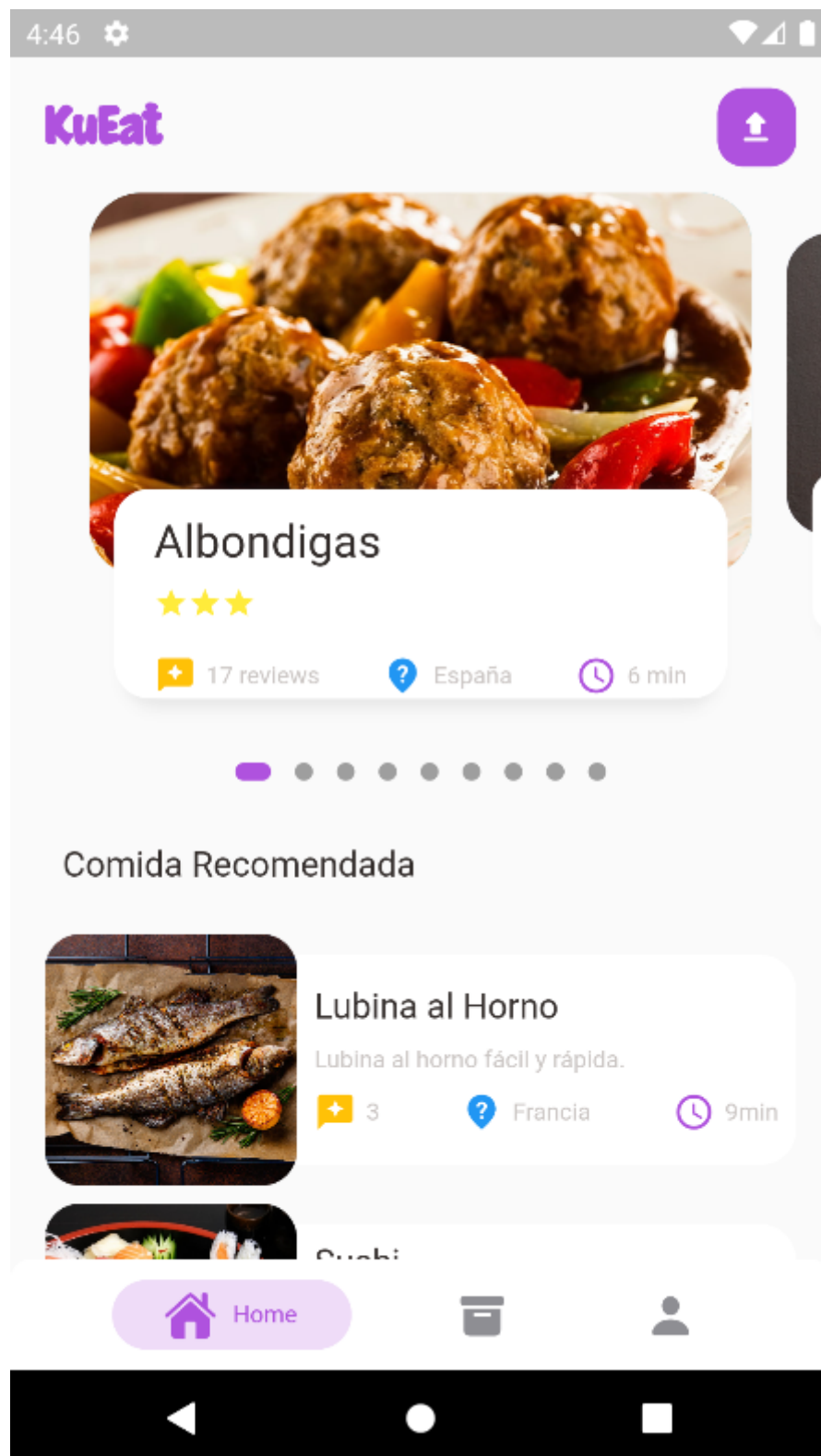
SPLASH SCREEN



Best Food Delivery

- Pantalla de inicio de carga de la app.

HOME SCREEN



- En esta pantalla de home vamos a poder encontrar la sección principal de la app, donde podemos observar que en la parte inferior dentro de la app, contamos con la barra de navegación para dirigirnos a las distintas pantallas dentro de la app, también encontraremos dos secciones principales, la sección de comidas populares en la parte superior de la pantalla, y en la parte central encontramos las comidas recomendadas, también encontraremos el botón para acceder al escáner de imágenes, y justo su izquierda encontraremos el título de la aplicación.

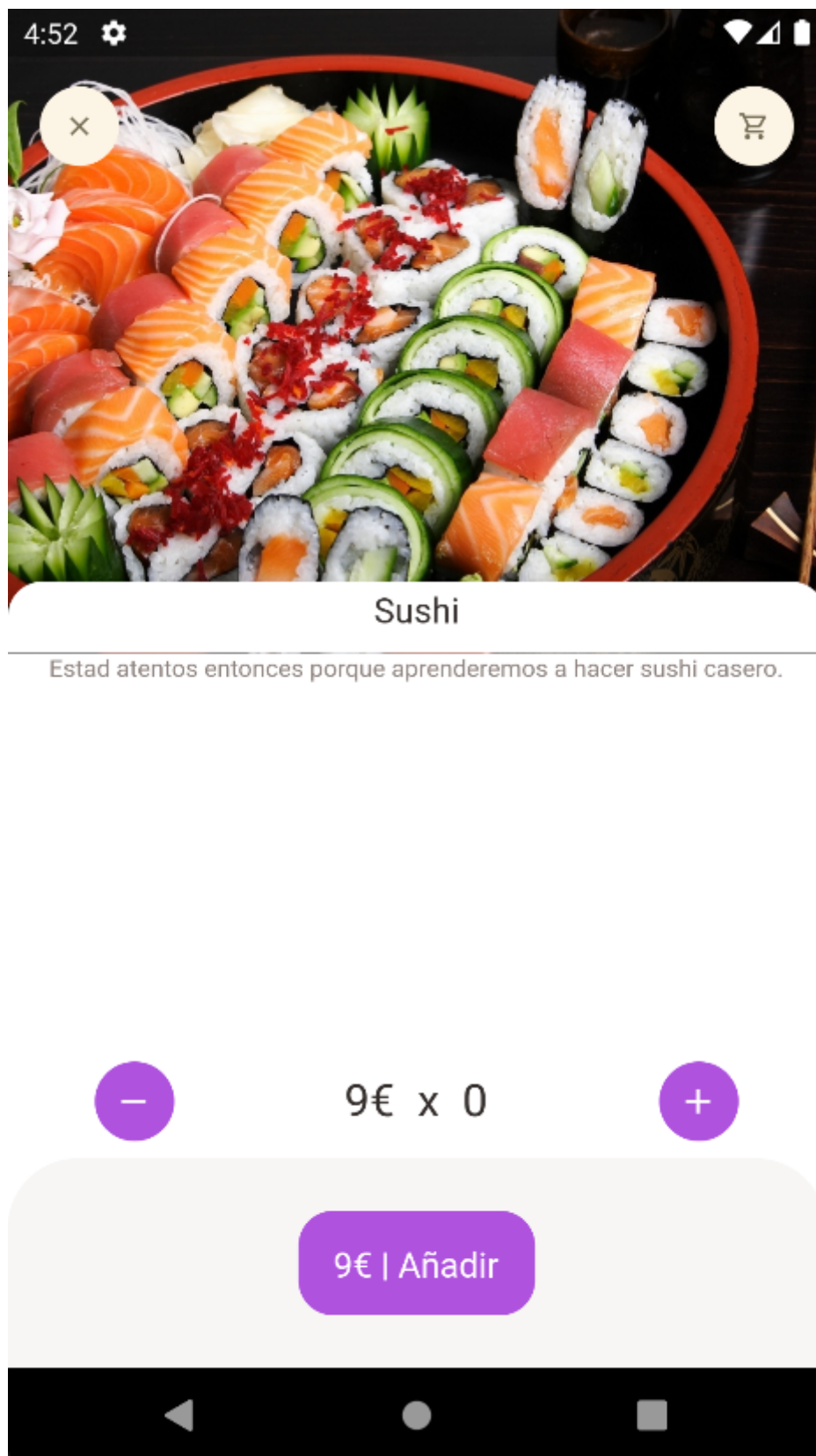
DETALLES DE COMIDA POPULAR



- Si pulsamos sobre una comida popular accederemos a la pantalla de detalles de dicha comida, en esta pantalla vamos a poder consultar la descripción de la comida, la imagen de la comida, vamos a poder ver el número de reviews que tiene esta comida, el lugar de procedencia de dicha comida, y el tiempo medio en el que se tarda en hacer dicha comida. Contaremos con un botón en la parte inferior derecha donde se muestra el precio de dicho producto y pulsando sobre este botón, podremos añadirlo a nuestro carrito. En la sección izquierda inferior, contamos con dos botones para aumentar la

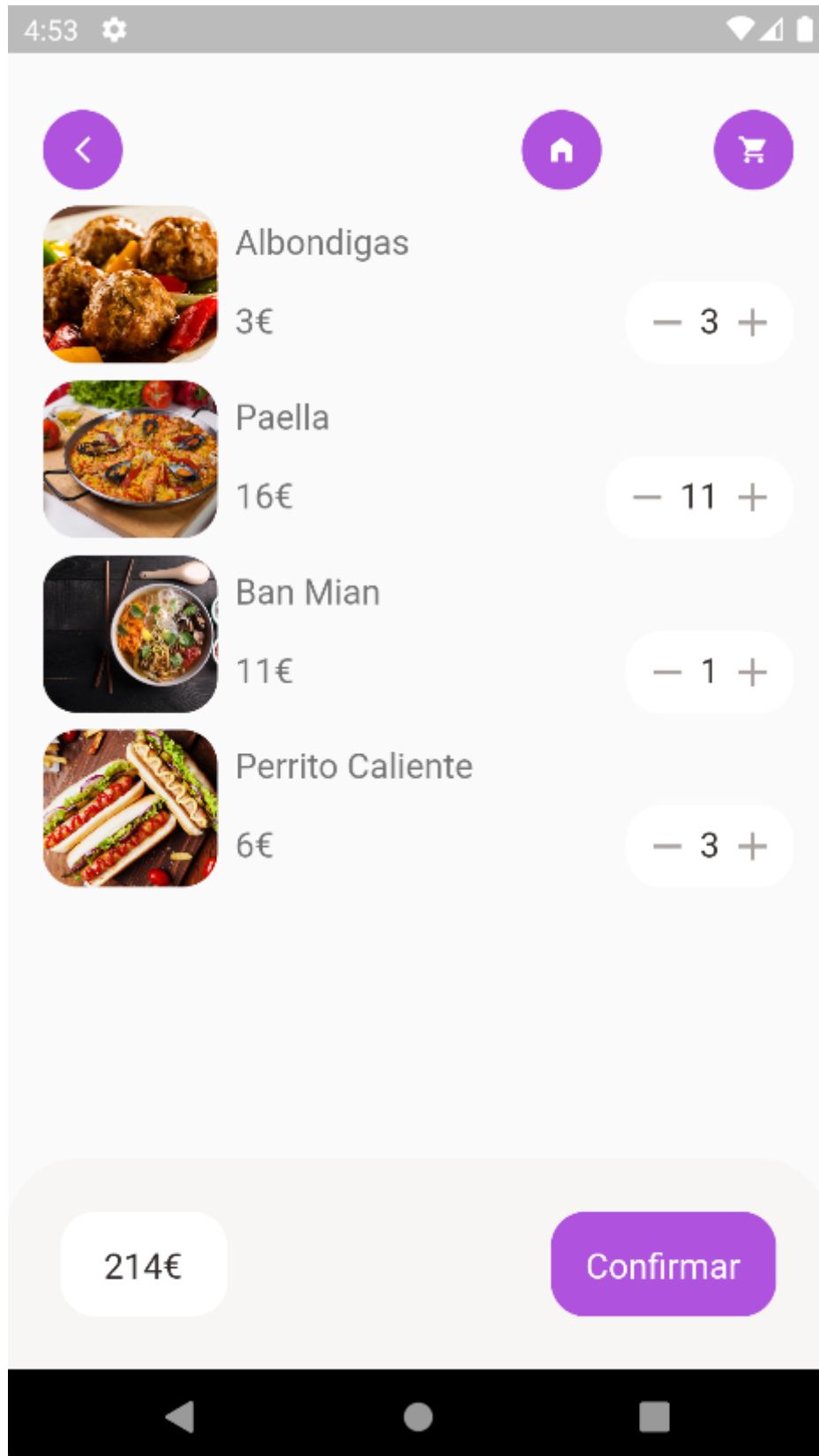
cantidad de este producto o disminuirla, el limite por pedido esta en 20 productos. En la sección superior contaremos con el botón para acceder al carrito, al igual que el botón para poder volver a la pantalla anterior a ésta.

DETALLES DE COMIDA RECOMENDADA



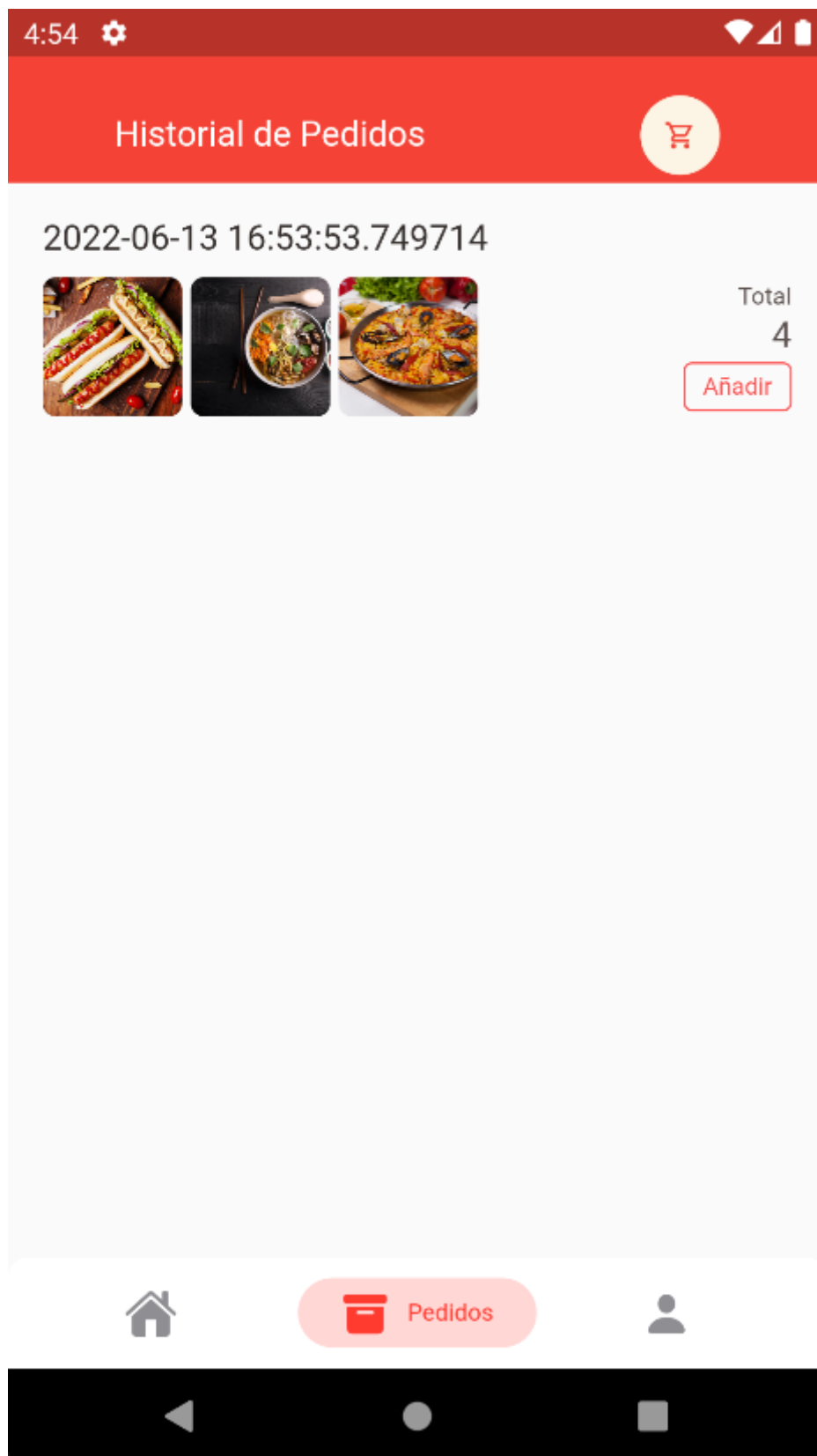
- Al pulsar sobre una comida recomendada entraremos a la pantalla de descripción de dicha comida recomendada, la imagen de la comida, en esta pantalla vamos a poder añadir o restar productos recomendados, al igual que añadirlos al carrito, en la parte superior vamos a poder acceder al carrito de compra y volver a la pantalla anterior.

CARRITO DE LA COMPRA



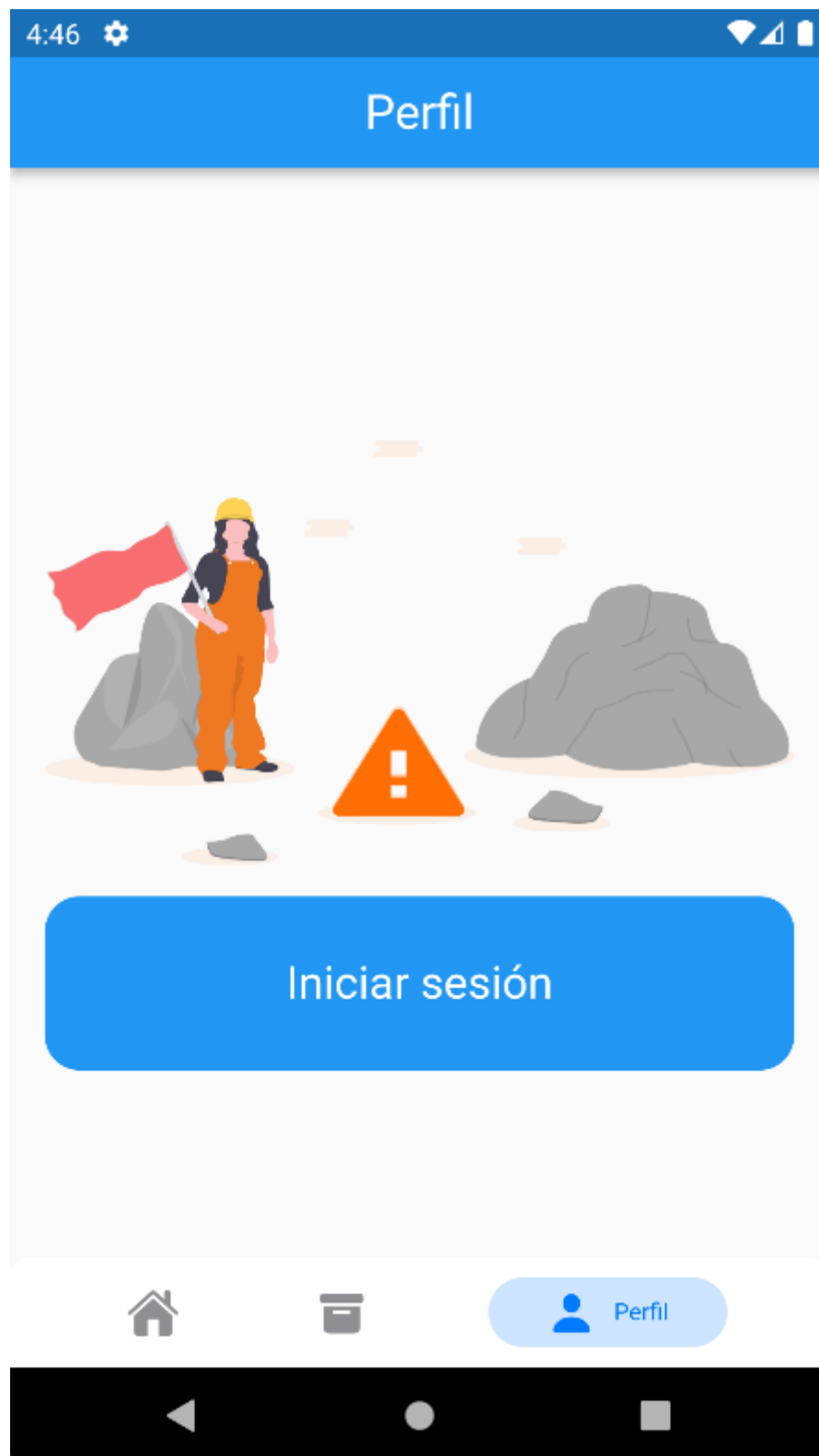
- En esta ventana veremos todos los productos que hemos ido añadiendo a nuestro carrito, desde esta pantalla vamos a poder añadir mas productos a cada uno de los que ya tenemos en el carrito, veremos una pequeña imagen de cada uno de los productos que hemos ido añadiendo, nombre, precio y numero de productos de este pedido, en la parte inferior veremos el total del pedido. También contaremos con el botón para volver a nuestra pantalla de home, también para poder volver a la ventana anterior a esta. A la hora de pulsar sobre confirmar el pedido, este se añadirá a nuestro historial de pedidos.

HISTORIAL DE PEDIDOS



- En esta ventana vamos a poder consultar todos los pedidos que hemos hecho desde que descargamos la aplicación, si el pedido sigue existiendo vamos a poder añadir mas elementos a este pedido, si en el caso de que el producto que se pidió no existe, no podremos añadir mas productos de este. También podremos consultar la fecha en la que se hizo este pedido.

PERFIL SIN INICIAR SESIÓN



- Al pulsar sobre el icono de usuario dentro de la barra de navegación, si estamos registrados se nos pedirá que iniciemos sesión, si no es así nos pedirá que iniciemos sesión pulsando sobre el botón azul.

INICIAR SESIÓN



Bienvenido

Iniciar Sesión



Teléfono



Contraseña

Iniciar Sesión

No tienes cuenta? **Crear Cuenta**



- En esta pantalla contaremos con un pequeño diseño donde vamos a poder introducir nuestras credenciales de registro, por lo tanto, podremos introducir nuestro teléfono y nuestra contraseña, si estamos registrados vamos a poder acceder y dirigirnos a la pantalla principal, pero si no lo estamos, debemos de pulsar sobre crear cuenta para crear una nueva cuenta.

CREAR CUENTA



Email



Contraseña



Nombre



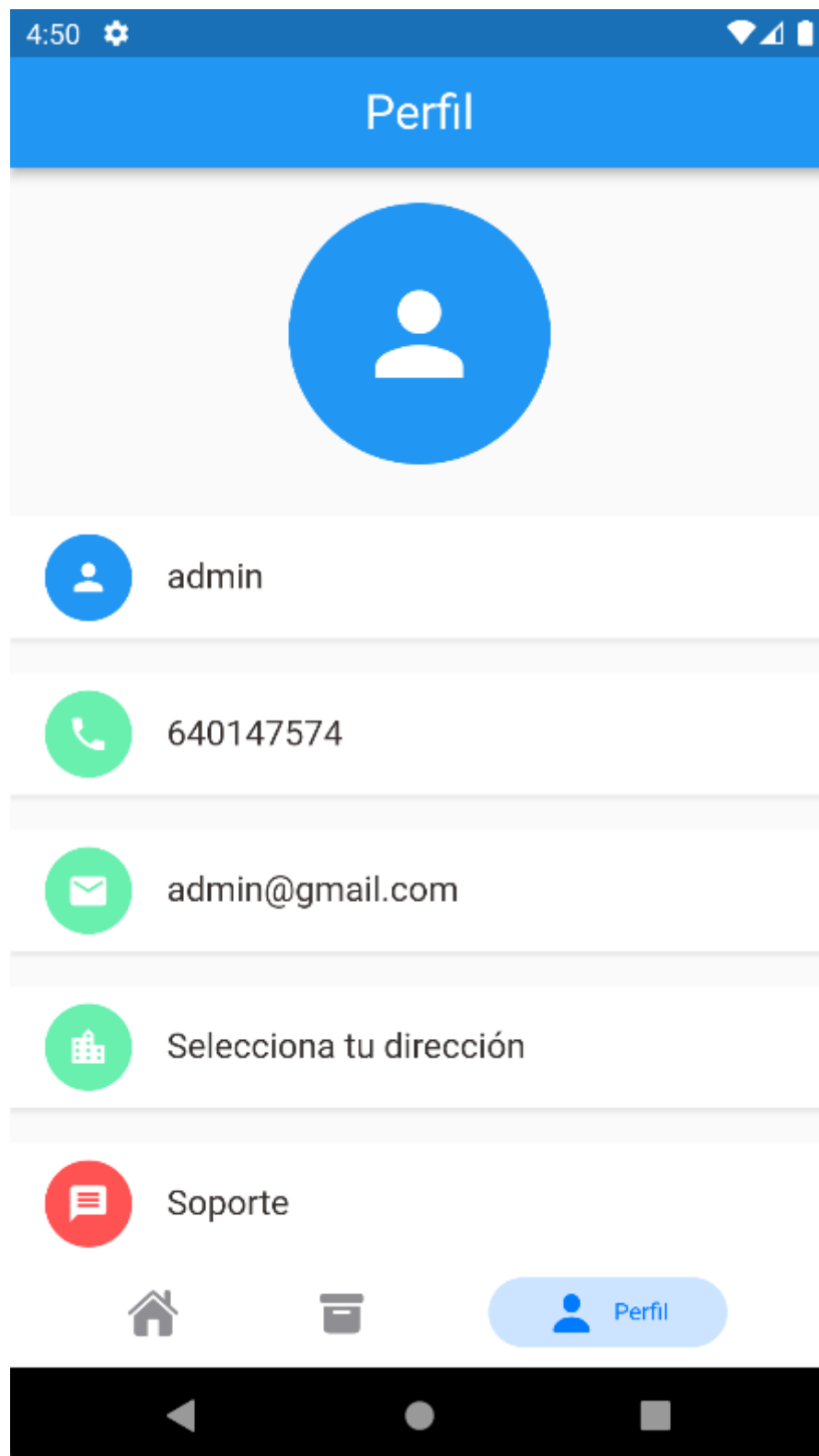
Teléfono

Registrarse

Ya estas registrado?

- En esta pantalla vamos a poder registrarnos, para registrarnos debemos de introducir un correo electrónico válido, una contraseña, nuestro nombre y teléfono. Finalmente solo debemos de pulsar en Registrarse, si estamos registrados podemos acceder rápidamente a la pantalla de inicio de sesión pulsando sobre Ya estas registrado?.

PERFIL DE USUARIO



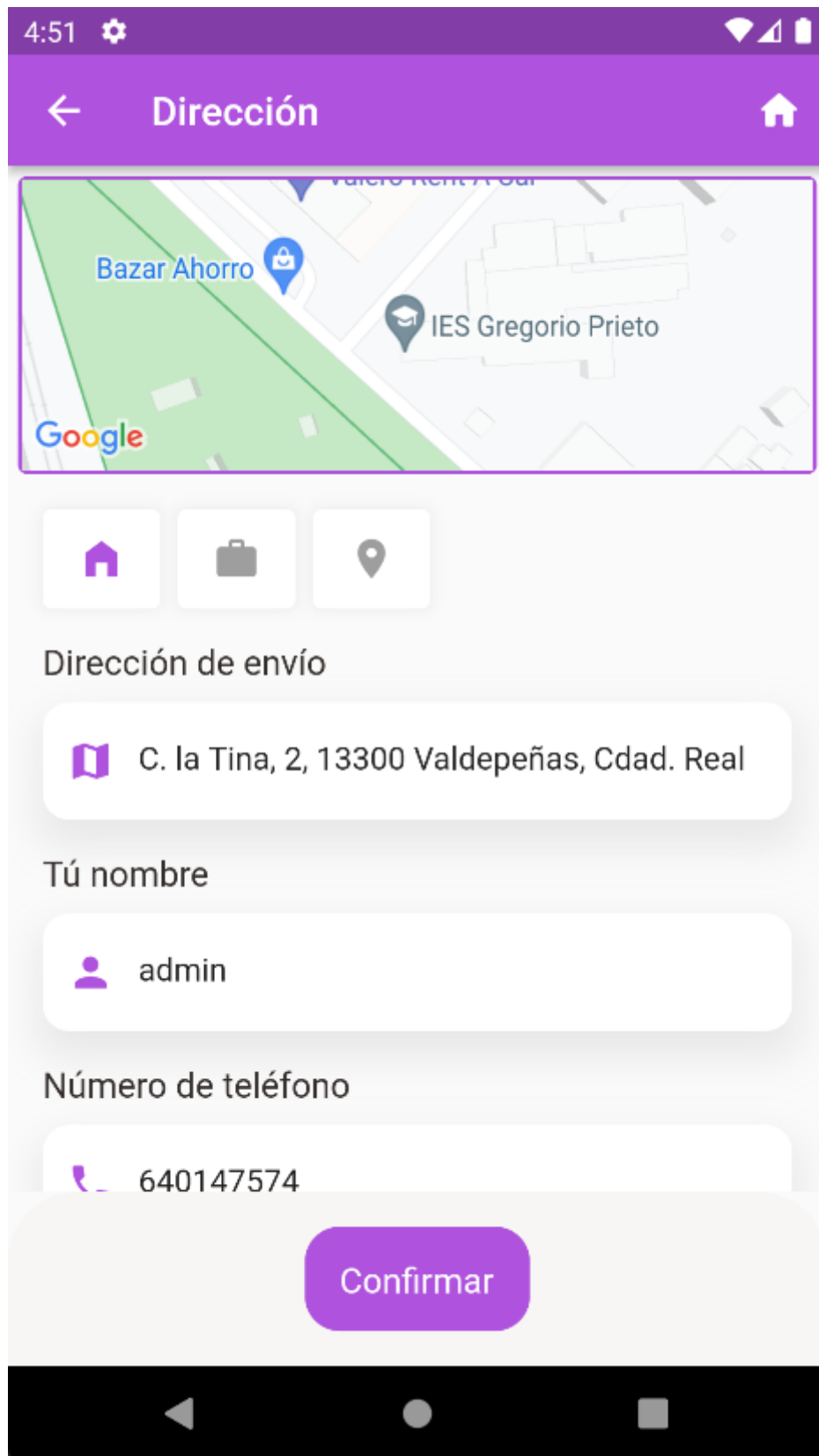
- En esta pantalla veremos que cuando hemos iniciado sesión vamos a poder encontrar aquí todos nuestros datos, vamos a poder contactar con soporte, pulsando en soporte, seleccionar nuestra dirección de envío, y finalmente podremos encontrar el botón de cerrar sesión, para salir de nuestra cuenta.

IA SOPORTE TECNICO 24/7



- Al pulsar sobre soporte técnico, vamos a poder hablar con un robot que se encarga de contestar todas nuestras dudas que tengamos sobre la app, por lo tanto contaremos con un textBox donde podremos introducir el mensaje que queremos enviar, y pulsaremos en el botón de enviar, para enviar el mensaje. En la parte superior contaremos con un botón para poder volver a nuestra pantalla de home.

SELECCIONAR DIRECCIÓN



- Al pulsar sobre mi dirección vamos a poder acceder a la pantalla de dirección, en esta pantalla vamos a poder seleccionar nuestra dirección con ayuda de Google Maps, por lo tanto nos deslizaremos por el mapa para seleccionar nuestra dirección, también se muestran los datos de contacto, como nuestro nombre y nuestro número de teléfono. Una vez que hayamos seleccionado nuestra dirección, vamos a poder pulsar sobre el botón de confirmar, para confirmar nuestra dirección. En la parte superior, contamos con un botón para poder volver a nuestra pantalla de inicio.



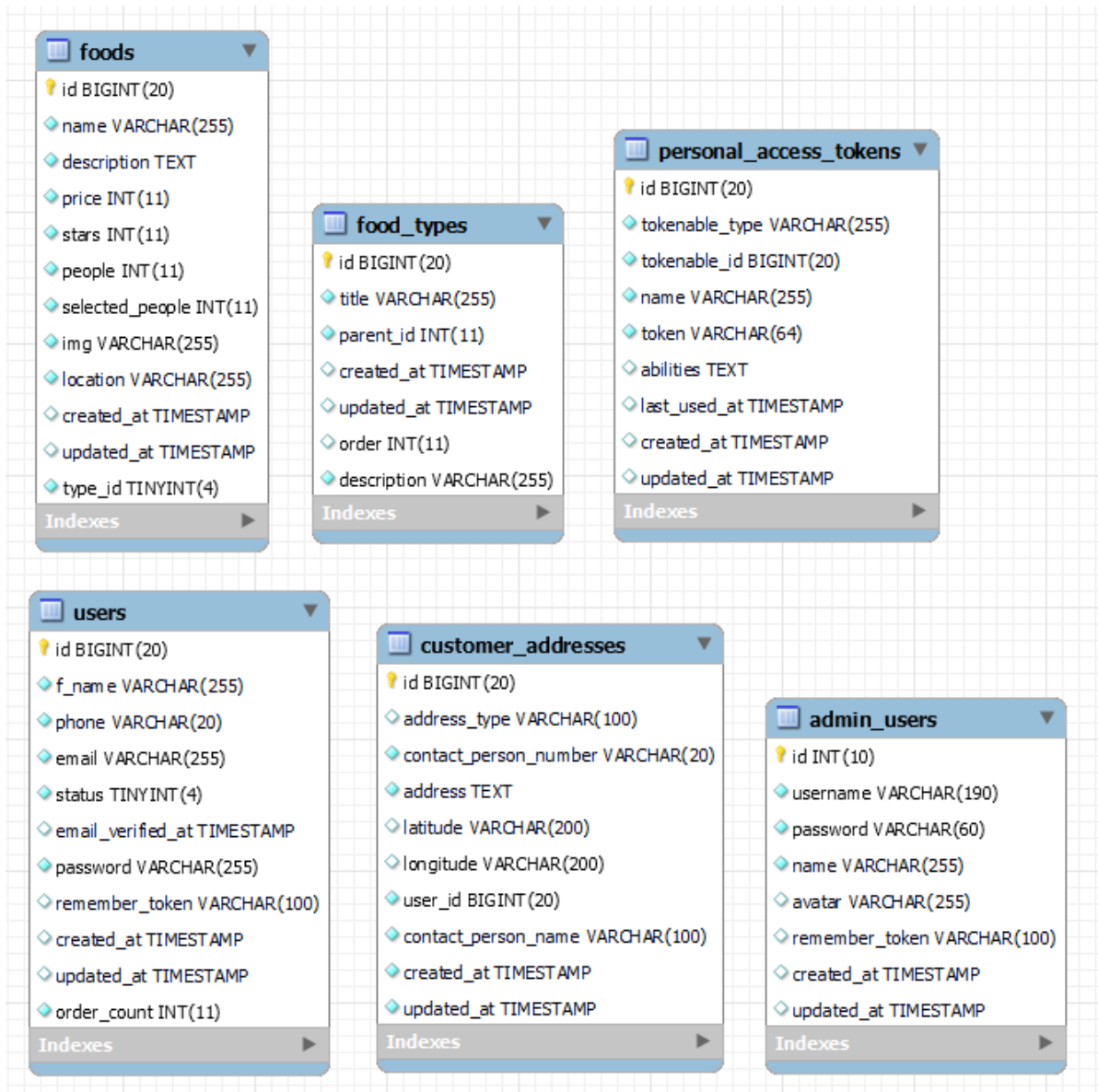
- En esta pantalla vamos a poder ver la inteligencia artificial que he integrado en mi app, esta es la apariencia de la pantalla antes de escanear cualquier comida, en esta pantalla vamos a poder subir una imagen desde nuestra galería, o vamos a poder realizar una foto con nuestra cámara. Cuando hayamos seleccionado una imagen la pantalla cambiara al siguiente estado.



- Cuando hayamos subido alguna imagen vamos a poder ver que nos dirá que comida es, por lo tanto la IA nos dice que comida es, también nos dirá el porcentaje de seguridad que tiene de que esa comida es la correcta, de acuerdo a la comida que se muestre, nos mostrará una descripción u otra , pulsando en el botón de mapa, nos dirigirá a Google Maps para que podamos ver en que zona cercana podemos comer dicha comida. En la parte superior vamos a poder pulsar sobre el botón de home para poder irnos a la pantalla de inicio.

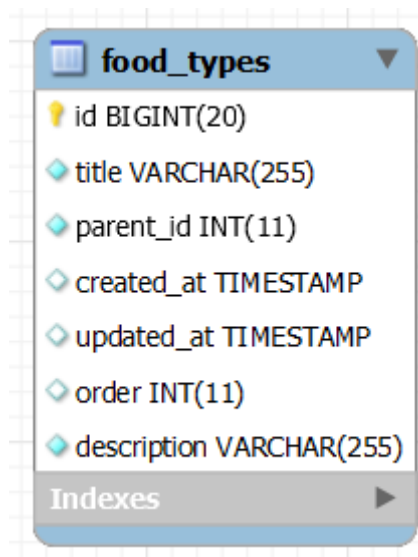
Análisis Funcional

Diagrama de la Base de Datos



- La base de datos solo tiene 6 tablas, he optado por una base de datos relacional, ya que tienes mas control sobre todos los campos, y así puedes asignar los valores a la hora de crearla mas fácilmente. Mi base de datos se compone de las siguientes tablas:

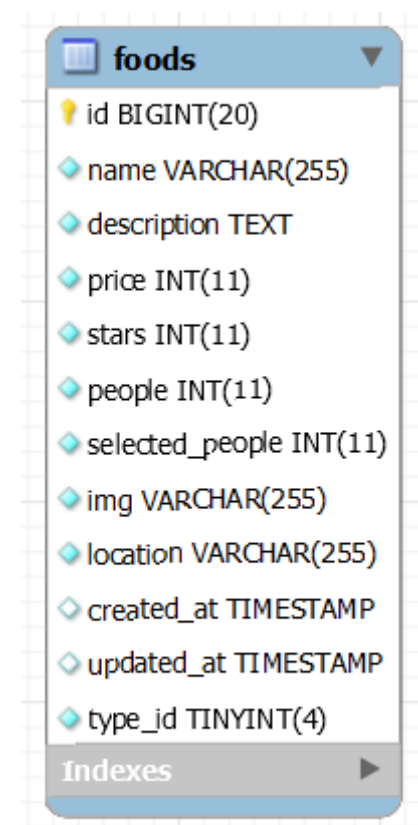
FOODS TYPES



food_types	
id	BIGINT(20)
title	VARCHAR(255)
parent_id	INT(11)
created_at	TIMESTAMP
updated_at	TIMESTAMP
order	INT(11)
description	VARCHAR(255)
Indexes	

- En esta tabla guardo los tipos de comidas, se compone de los campos de titulo de la comida, un identificador de comida, otro identificador para saber a que comida pertenece, la fecha de creación, fecha de actualización, una descripción, y por ultimo también guardo el numero de pedidos de la categoría de comida.

FOODS

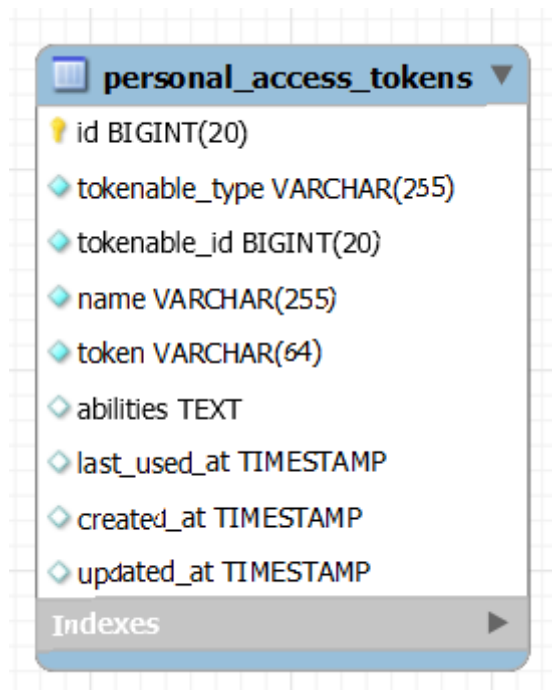


foods	
id	BIGINT(20)
name	VARCHAR(255)
description	TEXT
price	INT(11)
stars	INT(11)
people	INT(11)
selected_people	INT(11)
img	VARCHAR(255)
location	VARCHAR(255)
created_at	TIMESTAMP
updated_at	TIMESTAMP
type_id	TINYINT(4)
Indexes	

- En esta tabla guardo datos sobre la comida, se compone de los siguientes campos, contiene un identificador de la comida, el nombre comida, descripción de esta comida, el precio de esta comida, numero de estrellas que tiene esta comida, people se refiere al campo de numero de reviews, selected_people al numero al tiempo que se tarda en hacer dicha comida, guardo la url también de la imagen de la comida en img, en

location guardo la procedencia de la comida, fecha de creación y fecha de actualización, en cuanto al type_id se refiere a que categoría pertenece esta comida.

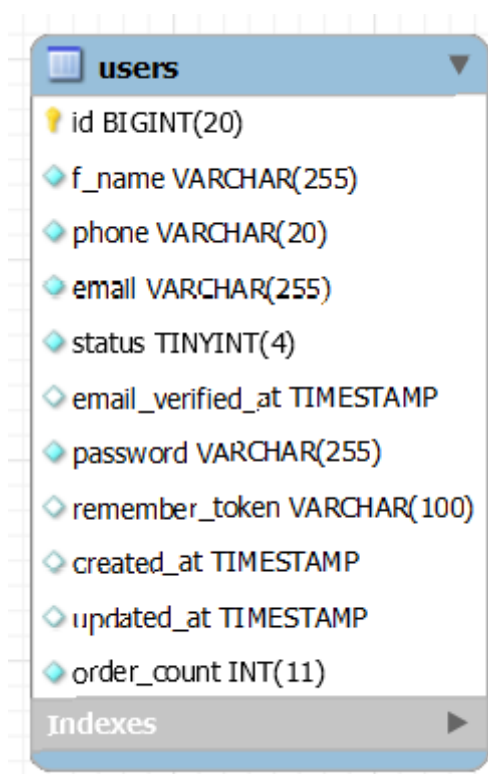
TOKENS



personal_access_tokens	
id	BIGINT(20)
tokenable_type	VARCHAR(255)
tokenable_id	BIGINT(20)
name	VARCHAR(255)
token	VARCHAR(64)
abilities	TEXT
last_used_at	TIMESTAMP
created_at	TIMESTAMP
updated_at	TIMESTAMP
Indexes	

- En esta tabla almaceno los tokens que genero para cada uno de los usuarios, guardamos el tipo de token, el id de dicho token, y las fechas de creación y actualización.

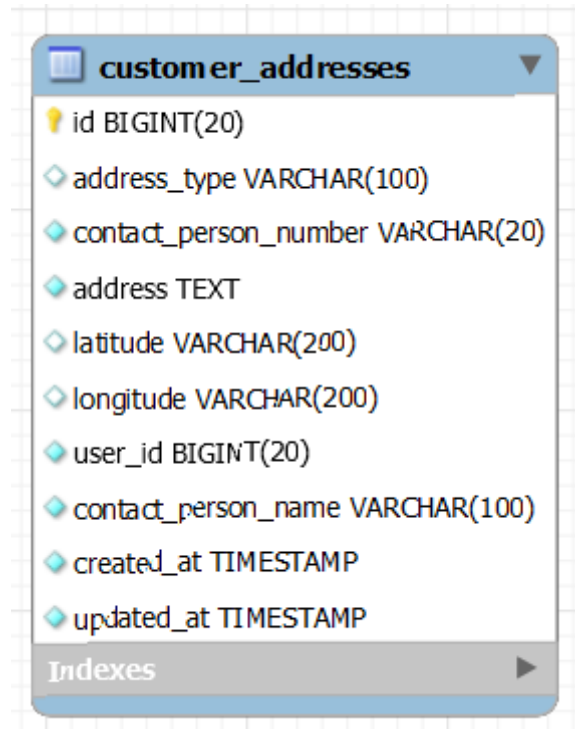
USUARIOS



users	
id	BIGINT(20)
f_name	VARCHAR(255)
phone	VARCHAR(20)
email	VARCHAR(255)
status	TINYINT(4)
email_verified_at	TIMESTAMP
password	VARCHAR(255)
remember_token	VARCHAR(100)
created_at	TIMESTAMP
updated_at	TIMESTAMP
order_count	INT(11)
Indexes	

- En esta tabla se almacenan los datos de los usuarios, por lo tanto, tendremos el nombre del usuario, su teléfono, correo electrónico, fecha de verificación del correo electrónico, contraseña encriptada, token del usuario, fecha de creación del usuario, y fecha de actualización, también guardo el numero de pedidos que ha hecho el usuario.

DIRECCIONES DE CLIENTES



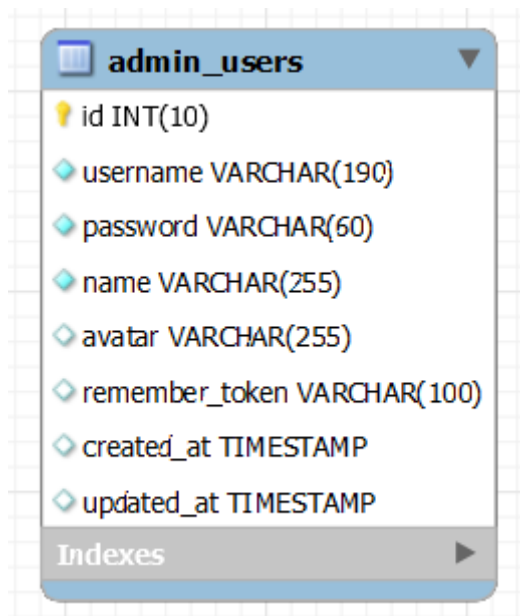
The image shows a screenshot of a database table definition for 'customer_addresses'. The table has the following columns:

Column Name	Data Type
id	BIGINT(20)
address_type	VARCHAR(100)
contact_person_number	VARCHAR(20)
address	TEXT
latitude	VARCHAR(200)
longitude	VARCHAR(200)
user_id	BIGINT(20)
contact_person_name	VARCHAR(100)
created_at	TIMESTAMP
updated_at	TIMESTAMP

Below the columns, there is a section labeled 'Indexes' with a right-pointing arrow.

- En esta tabla se almacenan los datos de los clientes, respecto a las direcciones de dichos clientes, por lo tanto almacenamos el tipo de dirección, que puede ser, casa, trabajo u otros, el numero de teléfono de dicho cliente, dirección, la latitud de su dirección, y altitud, nombre del cliente y fechas de creación y actualización.

USUARIOS ADMINISTRADORES



- Esta tabla esta reservada para el backend, ya que en esta tabla es donde inserto los usuarios administradores de mi cms, por lo tanto en esta tabla guardo el nombre de usuario, contraseña de administrador, nombre, y la imagen que es un avatar, token de dicho usuario, fecha de creación y fecha de actualización.

Backend



Para el backend de la aplicación he utilizado un framework llamado Laravel:

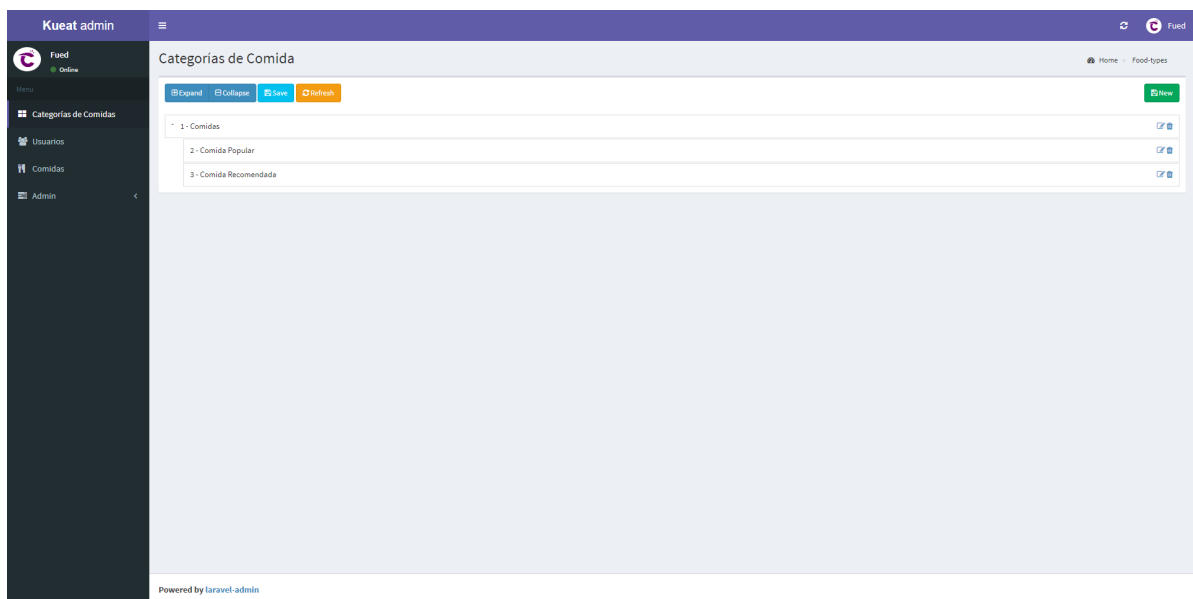
- **LARAVEL**

Laravel es un framework de código abierto que sirve para desarrollar aplicaciones y servicios web usando el lenguaje PHP, Laravel te permite desarrollar código en PHP de forma elegante y simple, Laravel esta hecho con la arquitectura MVC, gracias a Laravel, se aprovechan al máximo las últimas versiones de PHP, Laravel cuenta con muchísima documentación a través de internet, que te permite desarrollar cualquier tipo de funcionalidad de forma bastante rápida.

Wireframe CMS



Para administrar los datos de la aplicación he creado una web(CMS), para manejar todos los datos, esta web consta de tres apartador principales, el apartado para crear categorías de comidas, para crear usuarios, y para crear comidas. En la imagen que se encuentra encima de este párrafo, tenemos la apariencia principal de la web del CMS. Consta de un menú lateral donde tengo todas las opciones que se pueden seleccionar. Una imagen del usuario.



Si pinchamos en el apartado de Categorías de Comidas, vamos a entrar al apartado de categorías de comidas, como en mi aplicación solo tengo dos categorías de comidas, solo he creado estas dos. Luego también hay creada una categoría general, que engloba todas las categorías en una. Todas las comidas. Si quiero crear una nueva categoría simplemente podemos pulsar en New, y se nos abrirá el siguiente formulario:

Categorías de Comida
Create

Create

Categoría Main
ROOT

Nombre Categoría
Input Nombre Categoría

Descripción
Input Descripción

Pedido
- 1 +

Reset
View Continue creating Continue editing Submit

Dentro de este formulario podremos crear una nueva categoría de comida, por lo tanto le pondremos un nombre a la categoría una descripción y por ultimo el numero de pedidos limite que tendrá dicha categoría.

Kueat admin
Fused
Online

Menu
Categorías de Comidas
Usuarios
Comidas
Admin

Usuarios
List

+ New Export

	ID	Nombre	Email	Fecha de Verificación	Contraseña	Tokken	Fecha de Creación	Fecha de Actualización	Action
<input type="checkbox"/>	4		ff3@gmail.com				2022-05-26T16:41:41.000000Z	2022-05-26T16:41:41.000000Z	i
<input type="checkbox"/>	14		fued@gmail.com				2022-05-26T16:48:17.000000Z	2022-05-26T16:48:17.000000Z	i
<input type="checkbox"/>	24		fusedfed@gmail.com				2022-05-26T17:07:21.000000Z	2022-05-26T17:07:21.000000Z	i
<input type="checkbox"/>	34		fusedfssed@gmail.com				2022-05-26T17:18:27.000000Z	2022-05-26T17:18:27.000000Z	i
<input type="checkbox"/>	44		fuedsss@gmail.com				2022-05-26T17:52:44.000000Z	2022-05-26T17:52:44.000000Z	i
<input type="checkbox"/>	54		habiba@gmail.com				2022-05-26T17:57:52.000000Z	2022-05-26T17:57:52.000000Z	i
<input type="checkbox"/>	64		rudaina@gmail.com				2022-05-30T01:41:44.000000Z	2022-05-30T01:41:44.000000Z	i
<input type="checkbox"/>	74		fuedhorri123@gmail.com				2022-05-30T14:17:04.000000Z	2022-05-30T14:17:04.000000Z	i
<input type="checkbox"/>	84		admin@gmail.com				2022-05-31T02:00:49.000000Z	2022-05-31T02:00:49.000000Z	i
<input type="checkbox"/>	94		rubio123@gmail.com				2022-06-03T22:08:32.000000Z	2022-06-03T22:08:32.000000Z	i
<input type="checkbox"/>	104		felix@gmail.com				2022-06-06T10:24:29.000000Z	2022-06-06T10:24:29.000000Z	i
<input type="checkbox"/>	114		sami@gmail.com				2022-06-09T23:35:53.000000Z	2022-06-09T23:35:53.000000Z	i
<input type="checkbox"/>	124		fdefued@gmail.com				2022-06-11T01:32:11.000000Z	2022-06-11T01:32:11.000000Z	i
<input type="checkbox"/>	134		Mario@gmail.com				2022-06-16T18:37:04.000000Z	2022-06-16T18:37:04.000000Z	i

Showing 1 to 14 of 14 entries
Show 20 entries

Powered by laravel-admin

Si ahora en el menú seleccionamos la opción de Usuarios, nos vamos a dirigir al apartado de usuarios, donde vamos a poder encontrar todos los usuarios registrados en la app, si queremos podemos crear nuevos usuarios, pulsando en el botón de New.

Usuarios
Create

Create

Nombre
Input Nombre

Email
Input Email

Fecha de Verificación
2022-06-17 11:31:26

Contraseña
Input Contraseña

Tokken
Input Tokken

Reset
View Continue creating Continue editing Submit

Para el usuario, se requiere de un nombre, un correo electrónico, y una contraseña, el token lo podemos dejar vacío, ya que se generara automáticamente. Finalmente podemos pulsar en el botón de Submit para guardar los cambios.

Kueat admin

Fued

- Categorias de Comidas
- Usuarios
- Comidas
- Admin

Comidas List

ID	Nombre Comida	Categoria	precio	Número de Estrellas	Url de Imagen	Descripcion	Action
24	Albondigas	Comida Popular	3	3	https://p4-wallpaperbetter.com/wallpaper/853/312/975/meatballs-peppers-sauce-plate-walpaper-preview.jpg		[icon]
34	Allitas de Pollo	Comida Popular	4	5	https://media.istockphoto.com/photos/honeyoy-chicken-wings-picture-id1348674992?b=1&w=1066m=1348674992&p=1T086Ta&w=06H=hEeZvCfRlS4ScMu2UgJoAPDHNTLPq4-E5JHgHuCuRE=		[icon]
44	Helado Flocky	Comida Popular	6	5	https://images7.alphacoders.com/421/421534.jpg		[icon]
54	Bain Milan	Comida Popular	11	4	https://loving-london.com/es/wp-content/uploads/2017/09/restaurante-soho-londres-160912101639040-640x640.jpg		[icon]
64	Cocido	Comida Popular	13	5	https://costa.es/wp-content/uploads/2015/03/CostaCo_Cocado-de-garbanzos.jpg		[icon]
74	Croquetas de Garbanzo	Comida Popular	2	4	https://www.goya.com/media/6758/chickpea-croquettes.jpg?width=470		[icon]
84	Cus Cus	Comida Popular	7	5	https://canalcocina.es/medias/_cache/zoom-7635debaef19785ced801d166bf18f93-920-518.jpg		[icon]
94	Ensalada Vegetal	Comida Popular	6	3	https://www.isabel.net/wp-content/uploads/2019/06/ensalada-vegetal1920x1080.jpg		[icon]
104	Fideuá de Mariscos	Comida Popular	22	4	https://img.freepik.com/foto-gratis/fidea-mariscos-sobre-fondo-madera-vista-superior_75891-747.jpg		[icon]
114	Lubina al Horno	Comida Recomendada	14	4	https://blog.dia.es/wp-content/uploads/2017/02/LUBINA-AL-HORNO.jpg		[icon]
124	Sushi	Comida Recomendada	9	2	https://s2.best-wallpaper.net/Wallpaper/2560x1440/2002/Sushi-food-sticks-Japanese-food-2560x1440.jpg		[icon]
134	Galletas de la Fortuna	Comida Recomendada	2	1	https://astelus.com/wp-content/viajes/Galletas-de-la-fortuna.jpg	Estar galletas traen la bue	[icon]
144	Gazpacho	Comida Recomendada	6	5	https://images2.alphacoders.com/108/thumb-1920-1082886.jpg	El gazpacho es lo mas sencí	[icon]
154	Sardinas	Comida	28	3	https://static2.abc.es/media/bienestar/2020/10/01/sardinass-KWVG-620cx49(abc).jpg	Estar ricas sardinas	[icon]

En el menú si ahora pulsamos sobre el apartado de Comidas, vamos a poder acceder a la sección de comidas, donde podemos encontrar todas las comidas que e muestran dentro de la app, para poder crear una nueva comida podemos pulsar sobre el botón de New, y se nos mostrara el siguiente formulario:

Create

Nombre Comida

Input Nombre Comida

Categoria

ROOT

x ▾

Precio

-

Input Precio

+

Procedencia de Comida

Input Procedencia de Comida

Numero de Estrellas

-

Input Numer

+

Minutos

-

Input Minuto

+

Reviews

-

Input Review

+

Imagen

e

Input Imagen

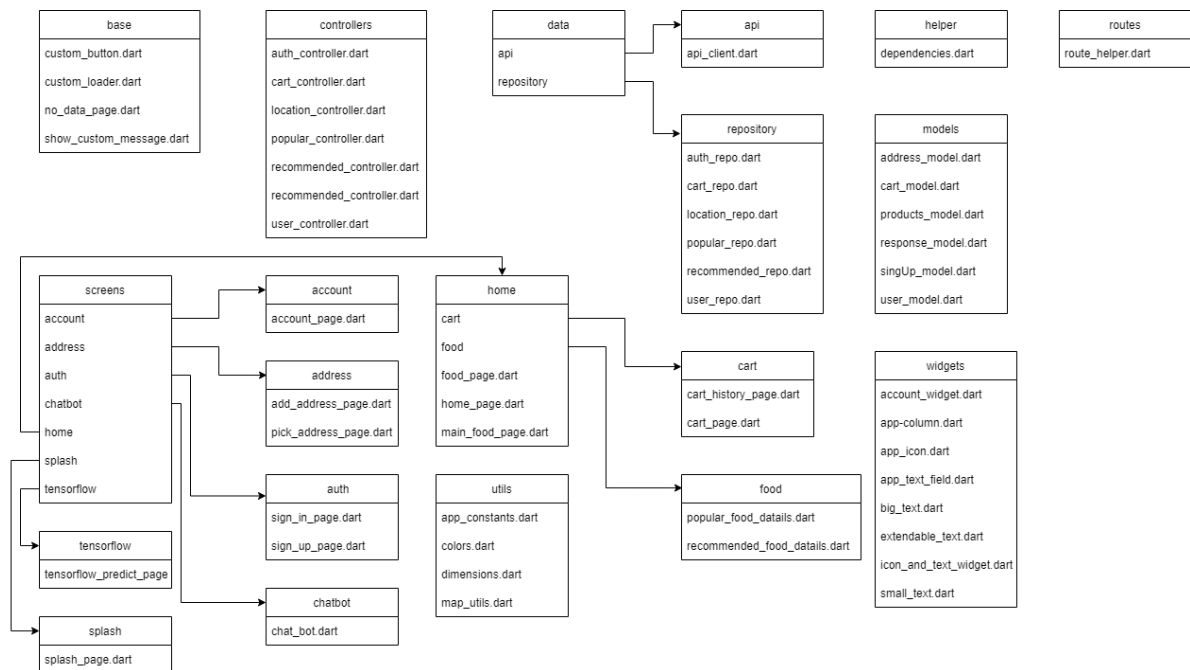
Descripcion

html

Para la comida, vamos a necesitar introducir, el nombre de la comida, debemos de seleccionar a que categoría pertenece, el precio de esta comida, el lugar de procedencia de dicha comida, el numero de estrellas que tiene dicha comida, el tiempo que normalmente se tarda en hacer esta comida, numero de reviews que tiene esta comida, también guardamos la URL de la imagen, y una descripción de dicha comida. Finalmente podremos pulsar en Submit, para guardar la comida.

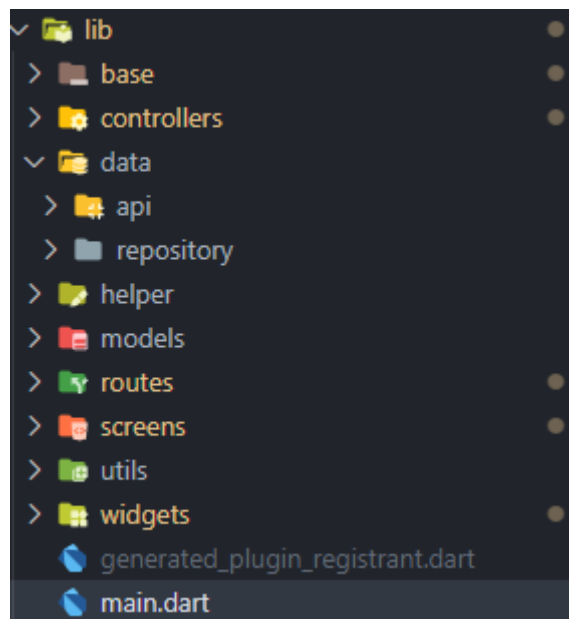
Diseño Técnico

Diagrama de paquetes y clases



Sistema de ficheros

- Explicaré el sistema de archivos de mi proyecto, lo que contiene cada carpeta:

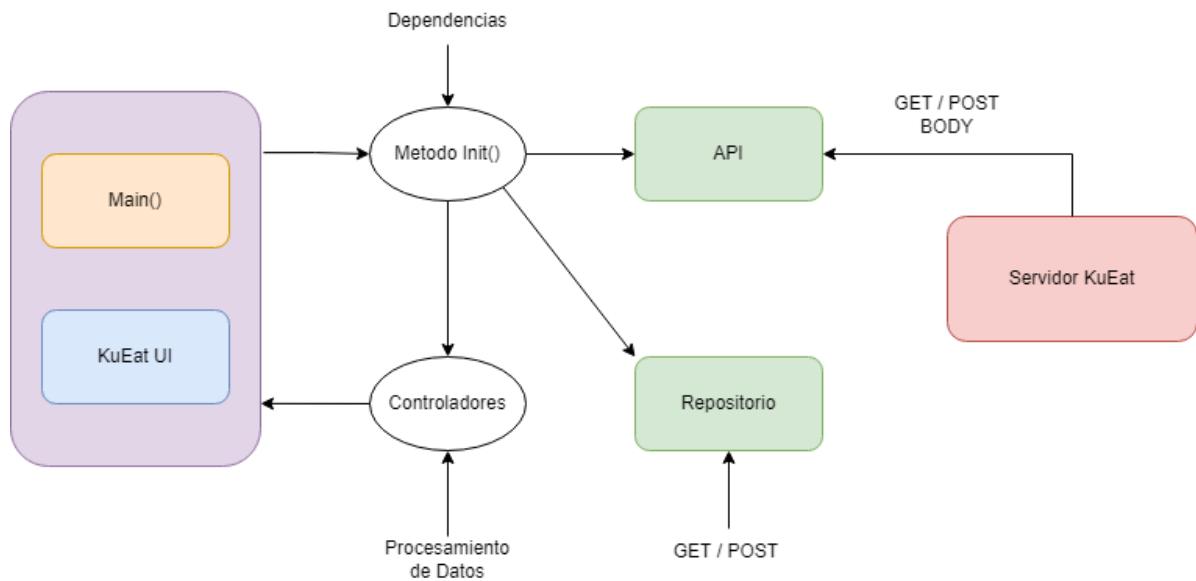


- Mi proyecto se compone de una carpeta principal, llamada **lib**, dentro de esta carpeta se encuentra todo el código fuente que he utilizado dentro de la app, dentro de la carpeta **base** podemos encontrar el código fuente de los distintos botones reutilizables dentro de la app, botones que utilizo en varios momentos dentro de la app, para que no se haga repetitivo el código, y sea reutilizable.

- Dentro de la carpeta **controllers**, encontramos cada uno de los controladores que se utilizan para manejar los datos que viene del servidor, por ejemplo, del servidor obtenemos las comidas populares y recomendadas, por lo tanto habrá un controlador de comidas, en el que tenemos aquellos métodos y atributos necesarios para el manejo de estos datos.
- En la carpeta **data** encontramos dos subcarpetas, en la carpeta de api, tengo los dos métodos que utilizo para conectarme al servidor, por método POST/GET. En la carpeta de repository, tengo cada uno de los repositorios de cada uno de los modelos, por lo tanto para comunicarme con el servidor tengo que instanciar un objeto de la clase repository, que este tiene la llamada al servidor, para obtener los datos necesarios, ya sea por método GET o POST.
- Dentro de la carpeta de **helper**, cuento con la inyección de datos, gracias a la librería de GetX, por lo tanto esta clase me permite el acceso de todos los métodos desde cualquier otra clase sin necesidad de instanciar un objeto de dicha clase.
- En la carpeta de **models** se encuentran todos los modelos utilizados en la app, por lo tanto por modelo tendremos un **repository** y un **controller**. Siguiendo el estándar de MVC.
- En la carpeta de **routes** tengo todos los cambios de pantalla que hago dentro de la app, por lo tanto para ir de una pantalla a otra, simplemente llamado a esta clase estática y llamo al respectivo método que me lleva a la pantalla que necesito. Dentro de esta clase también controlo las transacciones entre estas pantallas.
- Dentro de la carpeta de **screen**, tengo todas las pantallas de la app, el diseño de estas pantallas, creo una subcarpeta por pantalla para que así, si necesito utilizar otra clase dentro de esta pantalla, puedo crearla dentro de la propia carpeta de la pantalla.
- En la carpeta de **utils**, tengo aquellas clases estáticas que utilizo en toda la app, como por ejemplo los colores dentro de la app, las constantes de tipo string, o por ejemplo algunas redirecciones de google maps dentro de la app, esta clase me da acceso rápido a todas aquellas constantes.
- Finalmente en la carpeta de **widgets**, tengo aquellos widgets completos, o diseños reutilizables que uso a lo largo de la app, distintos diseños que comparten varias pantallas, las tengo incluidas en esta carpeta.

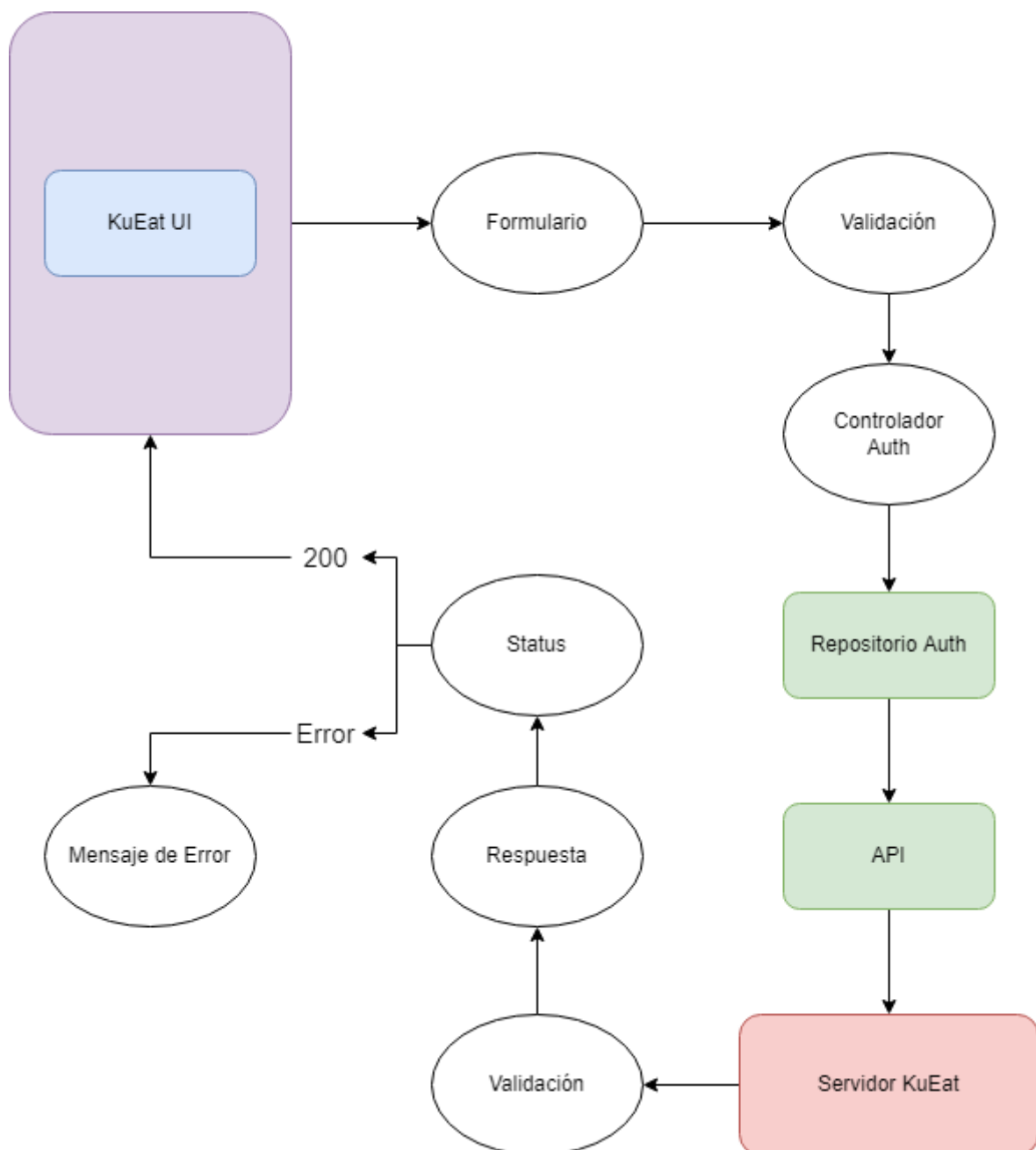
Arquitectura del sistema

Arquitectura del Sistema



- En cuanto a la arquitectura del sistema, es bastante sencilla, contamos con el inicio de la app, UI, esta UI se inicia con el método Init(), este método Init(), estará cargado con una serie de dependencias y métodos para poder comunicarse con el servidor.
- En la parte del backend, nuestro Servidor se encarga de insertar las comidas necesarias para así luego poder consumirlas a través de métodos GET desde la app móvil, por lo tanto el backend cuenta con un pequeño entorno para poder crear comidas y gestionar el registro de usuarios.
- Dentro de nuestra app, contamos con un repositorio que es el que se encarga de obtener esas respuestas get de la API, una vez que accedes estas peticiones get/post, las procesamos con una serie de controladores, para poder convertir esa respuesta en JSON en listas de objetos dentro de la app.
- Al igual que las peticiones POST, enviaremos datos al servidor en formato JSON para que así se puedan procesar en el backend y guardar en la base de datos.
- Una vez que se hayan procesado estos datos en el backend y se haya enviado la respuesta a la app, los controladores se encargaran de mostrarlo en la UI.

Diagrama de Autenticación



- Como dentro de la aplicación contamos con registro e inicio de sesión de usuarios, necesitamos de una arquitectura de autenticación del usuario.
- En el apartado de **Registro** el usuario, rellena el formulario que se muestra dentro de la app, rellinando todos los datos, de numero de teléfono, nombre, correo electrónico y contraseña, este formulario primero se valida dentro de la app, para ver si es correcto el formato, si el número de teléfono es correcto, la contraseña es de mas de 6 caracteres, y el correo electrónico también es válido.
- Seguidamente estos datos se envían al controlador de autenticación, para convertir dichos datos a formato JSON, y seguidamente se lo pasamos a nuestro repositorio dentro de la aplicación.
- Este repositorio se comunica con la API REST, se validan los datos en nuestro servidor, para comprobar que dicho usuario no existe en la base de datos, o que dicho numero de teléfono o correo electrónico es único, si todo es correcto, se manda una respuesta a nuestra app de nuevo, si esta respuesta es 200, podemos verificar que todo el proceso de registro a sido correcto y dejaremos que el usuario tenga acceso completo a la aplicación, cabe destacar

que a dicho usuario se le asigna un token único, para que luego pueda volver a iniciar sesión correctamente.

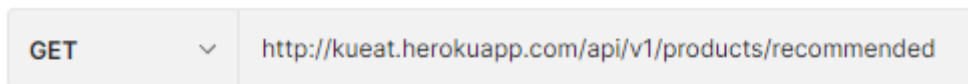
- En el caso de **Inicio de Sesión**, el usuario debe de introducir su número de teléfono y su contraseña para iniciar sesión, de nuevo se validará si los datos son correctos, si dichos datos existen en la base de datos, se recupera el token de dicho usuario, para que pueda acceder correctamente a la aplicación con sus datos.

Api Rest

- En cuanto a la comunicación entre la app móvil y el servidor, contamos con una api, a continuación muestra las llamadas a la api y los resultados que esta devuelve en formato JSON.

- **PETICIONES GET**

- Petición para obtener las comidas recomendadas.



- Respuesta de comidas recomendadas. Se obtienen todos los productos recomendados, cada uno con su respectivo campo, se devuelve el tamaño del número de productos existentes, al igual que en los productos populares.

```
"total_size": 7,
"type_id": 3,
"offset": 0,
"products": [
  {
    "id": 114,
    "name": "Lubina al Horno",
    "description": "Lubina al horno fácil y rápida.",
    "price": 14,
    "stars": 4,
    "people": 9,
    "selected_people": 3,
    "img": "https://blog.dia.es/wp-content/uploads/2017/02/LUBINA-AL-HORNO.jpg",
    "location": "Francia",
    "created_at": "2022-06-09 00:51:37",
    "updated_at": "2022-06-09 01:14:21",
    "type_id": 3
  },
  {
    "id": 124,
    "name": "Sushi",
    "description": "Estad atentos entonces porque aprenderemos a hacer sushi casero.",
    "price": 9,
    "stars": 2,
    "people": 12,
    "selected_people": 3,
    "img": "https://s2.best-wallpaper.net/wallpaper/2560x1440/2002/Sushi-food-sticks-Japanese-food_2560x1440.jpg",
    "location": "China",
    "created_at": "2022-06-09 00:53:59",
    "updated_at": "2022-06-09 00:56:28",
    "type_id": 3
  }
],
```

- Petición para obtener las comidas populares

GET

http://kueat.herokuapp.com/api/v1/products/popular

- Respuesta de comidas populares

```
"total_size": 9,  
"type_id": 2,  
"offset": 0,  
"products": [  
  {  
    "id": 24,  
    "name": "Albondigas",  
    "description": "Las albóndigas caseras de carne en salsa siempre me han recordado a mis tiempos de niño, a receta de  
    también. Esta receta de albóndigas a la cazuela o en salsa es la que me enseñó mi abuela Lucrecia, que está tal  
    verduras, de las cosillas que daba el huerto y un poco de vino blanco de casa, pero en el que podéis emplear un  
    cantidad de verduras y hortalizas que llevan. Su sabor es intenso, mezcla de todos los recuerdos de como hacía m  
    de patatas fritas. Además siempre es un buen entretenimiento que requiere de algo de maña pero comerlas es más di  
    probado, las albóndigas con tomate y estas albóndigas en salsa. Para estas albóndigas de carne siempre me gusta  
    y o bien sólo cerdo. Os dejo con esta comida casera sin mucha complicación, una forma de comer carne y verdura de  
    éstas verduras no te gustan podrás cambiarlas por las de tu elección, espero que os animéis con ellas.",  
    "price": 3,  
    "stars": 3,  
    "people": 6,  
    "selected_people": 17,  
    "img": "https://p4.wallpaperbetter.com/wallpaper/853/312/975/meatballs-peppers-sauce-plate-wallpaper-preview.jpg",  
    "location": "España",  
    "created_at": "2022-06-08 06:16:46",  
    "updated_at": "2022-06-09 00:43:50",  
    "type_id": 2  
  },  
]
```

Entorno de desarrollo, librerías y servicios

En este punto se explicarán las diferentes tecnologías utilizadas para la realización del proyecto, así como los elementos más importantes que permitan entender el funcionamiento del sistema.

- **ENTORNO DE DESAROLLO**



- A la hora de montar todo el entorno para desarrollar tanto el backend como la app móvil, he utilizado Visual Studio Code, ya que por excelencia es el entorno mas sencillo de utilizar y mas potente, ya que cuenta con todas las extensiones necesarias para poder desarrollar aplicaciones móvil, al igual que emuladores de todo tipo para poder ejecutar la aplicación y realizar pruebas.

- **TECNOLOGIA**



- He decidido utilizar **Flutter** para el desarrollo de mi app, Flutter es un SDK desarrollado por Google para crear aplicaciones móviles tanto para Android como para iOS (Apple). Fue desarrollado como un software para uso interno dentro de la compañía pero vieron el potencial que tenía y decidieron lanzarlo como proyecto de código libre. Actualmente es uno de los proyectos de desarrollo de aplicaciones móviles que más está creciendo. Además, desde la última versión estable, también es posible realizar aplicaciones Web y de escritorio para Windows y Mac. Aunque en estas dos plataformas aún está en fase experimental (beta).
- Estas son las tres principales ventajas que ofrece Flutter respecto a otras frameworks de desarrollo de aplicaciones multiplataforma:
 1. **Compila en nativo**, tanto en Android como en iOS.
 2. La **creación interfaces gráficas** es muy flexible, puedes combinar diferentes Widgets (elementos gráficos) para crear las vistas.
 3. El **desarrollo es muy rápido**, permite ver el resultado de forma instantánea mientras se escribe el código.
- Flutter me ha permitido desarrollar la UI muy rápidamente, ya que con los widgets se hace muy rápido el hecho de mostrar elementos en la pantalla, además te permite tener el control absoluto sobre cada elemento en la pantalla. En Flutter, se programa en su lenguaje llamado Dart, es un lenguaje muy parecido a Java, tiene una sintaxis muy parecida pero con algunos algoritmos propios de Dart, pero es bastante sencillo programar en este lenguaje, si se ha programado en Java, no se hace muy complicado utilizarlo.

- **LIBRERIAS UTILIZADAS**

```
dependencies:
  flutter:
    sdk: flutter
  get: ^4.1.4
  dots_indicator: ^2.1.0
  persistent_bottom_nav_bar: ^4.0.2
  shared_preferences: ^2.0.11
  image_picker: ^0.8.5+3
  intl: ^0.17.0
  tflite: ^1.1.1
  meta: ^1.1.1
  flutter_speed_dial: ^4.3.0
  google_maps_flutter: ^2.1.2
  geocoding: ^2.0.4
  geolocator: ^8.2.0
  url_launcher: ^5.4.0
  bubble:
```

- Entre las librerías a destacar que he utilizado en mi proyecto, cabe destacar las siguientes:

- **GetX**

GetX es una solución extra ligera y potente para Flutter. Combina gestión de estados de alto rendimiento, inyección de dependencia inteligente y gestión de rutas de forma rápida y práctica.

GetX tiene 3 principios básicos: esto significa que esta es la prioridad para todos los recursos de la biblioteca.

- **Rendimiento:** GetX se centra en el rendimiento y el consumo mínimo de recursos.
- **Productividad:** GetX utiliza una sintaxis fácil y agradable para agilizar el desarrollo.
- **Organización:** GetX permite el desacoplamiento total de la vista de la lógica de negocio.

Con GetX se gana versatilidad en la implementación y gestión de componentes, así como el manejo de la lógica de negocio de una forma más sintáctica y cómoda. También provee métodos para el manejo de objetos y propiedades observables que se actualizan en tiempo real

- **Dots Indicator**

Es nada mas que una librería para poder mostrar dentro de nuestro UI, una serie de puntos para luego unirlos por ejemplo a un Slider View, donde se muestran imágenes. Dots Indicator es muy sencillo de implementar, ya que solo tenemos que introducir la dependencia dentro de nuestro archivo pubspec.yaml, una vez introducida la dependencia vamos a poder, hacer el import de dicha dependencia y utilizarla donde nosotros deseemos dentro de nuestra UI.

- **Persistent Bottom Nav Bar**

Al igual que dots indicator, esta librería nos permite crear una barra de navegación dentro de nuestra app, dicha barra de navegación tiene una apariencia bastante llamativa, e incluye muchos tipos de barras de navegación, podemos modificarla a nuestro gusto, para usar la librería simplemente hay que incluir la dependencia dentro

de nuestro archivo pubspec.yaml, y posteriormente realizar el import de dicha clase donde queramos inicializar dicha barra de navegación.

- **Shared Preferences**

Las Shared Preferences son una buena forma de guardar datos con Flutter. Esta práctica se deriva de la funcionalidad del mismo nombre, disponible en Android y de UserDefaults en iOS.

Con este sistema, los datos se pueden conservar en el dispositivo del usuario de forma asincrónica.

Para poder utilizar dicha librería, solo tenemos que añadir dicha dependencia a nuestro archivo pubspec.yaml, después haciendo el import de la clase, vamos a poder utilizar la librería correctamente.

- **Image Picker**

Un complemento de Flutter para iOS y Android para seleccionar imágenes de la biblioteca de imágenes y tomar nuevas fotos con la cámara.

- **Intl**

La biblioteca más importante es intl. Define la clase Intl, con la configuración regional predeterminada y los métodos para acceder a la mayoría de los mecanismos de internacionalización. Esta biblioteca también define las clases DateFormat, NumberFormat y BidiFormatter.

- **Tflotter**

Un complemento de Flutter para acceder a la API de TensorFlow Lite. Admite clasificación de imágenes, detección de objetos (SSD y YOLO), Pix2Pix y Deeplab y PoseNet en iOS y Android.

- **Meta**

Este paquete define las anotaciones que pueden usar las herramientas que se envían con el SDK de Dart.

- **Flutter Speed Dial**

Este widget es un widget especial para poder introducir subelementos dentro de un mismo botón, así como se muestra en la siguiente imagen:

- **Google Maps Flutter**

Un complemento de Flutter que proporciona un widget de Google Maps.

- **Google Maps Geocoding**

Un complemento de geocodificación de Flutter que proporciona funciones de geocodificación sencilla y geocodificación inversa.

- **DialogFlow**

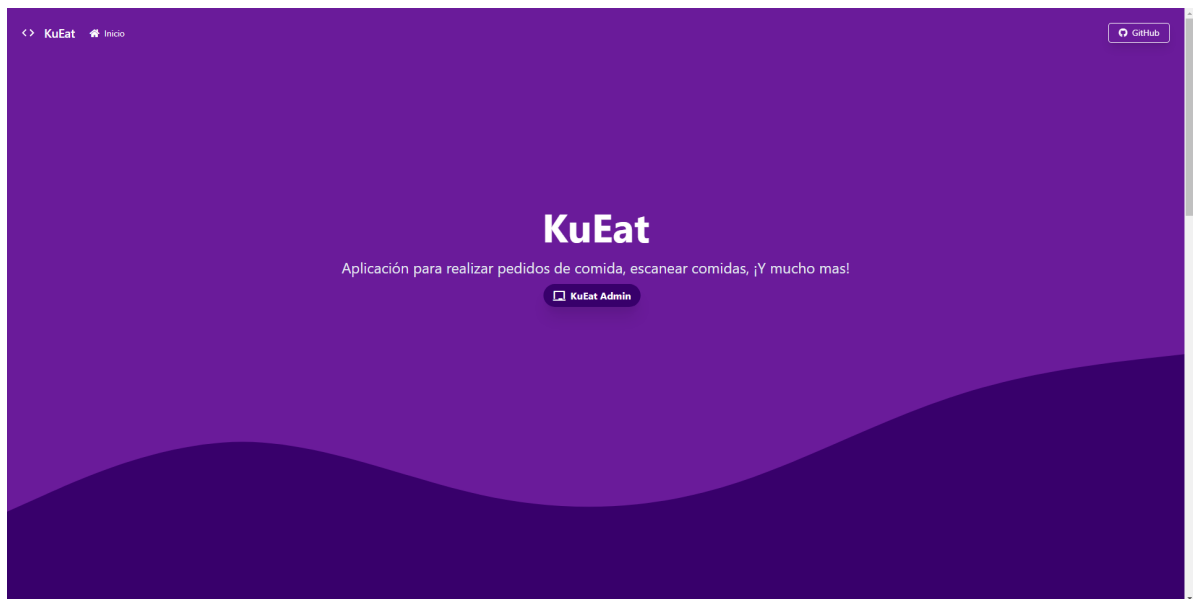
Este complemento te permite integrar inteligencia artificial de chatbots, esta librería te permite comunicarte con la API de DialogFlow para poder implementar el chatbot que he creado dentro de mi app Flutter.

- **Geolocator**

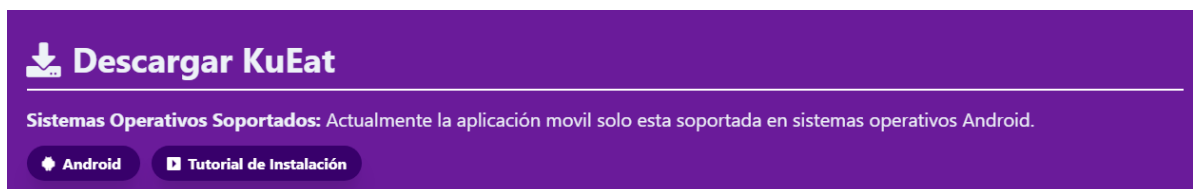
Un complemento de geolocalización de Flutter que brinda fácil acceso a los servicios de ubicación específicos de la plataforma (FusedLocationProviderClient o, si no está disponible, LocationManager en Android y CLLocationManager en iOS).

Instrucciones para la compilación, ejecución y despliegue de la aplicación

Sencillamente, para instalar la aplicación os dirigís a la pagina web de KuEat:



<https://kueatweb.herokuapp.com/>



- Esta es mi pagina web oficial donde podéis dirigiros a ella pinchando en el enlace de arriba, una vez pinchado arriba, vamos a deslizar un poco hacia abajo dentro de la pagina web, y pulsaremos en el botón de Android. Este botón nos dirigirá, a un enlace de Google Drive donde vamos a poder descargar el Apk de la app.
- También podremos ver un tutorial para como poder instalar un Apk.

Conclusiones

Conocimientos adquiridos

Gracias a las FCTS, los lenguajes que he estado usando durante estos meses me ha ayudado a enfocar de una manera mucho mas productiva el proyecto de fin de grado, con ello he estado usando bastante inteligencia artificial, por ello decidí incluirlo dentro de mi aplicación, también tuve la suerte que estuve en un proyecto de Backend donde se utilizaba PHP, por ello decidí realizar mi Backend en PHP, con ayuda de Laravel y un paquete para implementar todo el CRUD de usuarios y comidas.

Antes de hacer la aplicación ya tenía pensado hacerla en Flutter, ya que he hecho antes otras aplicaciones en dicho Framework, por lo tanto para aumentar mis conocimientos en este lenguaje, decidí comprar un curso para poder mejorar mi forma de programar en Dart, el lenguaje que usa Flutter, al cabo del tiempo todo esto me llevo a mejorar en cuanto a mi forma de programar.

Entre otros conocimientos a destacar, podríamos hablar de Docker, que también he aprendido a utilizarlo, aunque a la hora de implementarlo con mi Backend se complico por la base de datos relacional que tengo.

Mejoras futuras

Una de las principales mejoras sin duda alguna, es poder llegar a implementar pagos reales dentro de la app, ya que he estado mirando como poder integrar pagos con PayPal y con tarjeta bancaria desde, con ayuda del SDK de PayPal, pero debido al tiempo no he podido incluirlo.

Otra de las mejoras que seria interesante incluir dentro de la app, es poder tener categorías diferentes y poder elegir los ingredientes de cada una de las comidas, o por ejemplo poder construir un menú de comida a nuestro gusto.

También estaría bien la posibilidad de permitir al usuario que recupere su contraseña, si se da el caso de que el usuario ha olvidado su contraseña.

Dentro del apartado de usuario también estaría bien permitir que el usuario pudiera registrarse usando Google.

También me gustaría separar la funcionalidad de la IA en otro backend diferente, en Python, ya que es mucho mas rápido escanear imágenes y procesarlas en Python, por lo tanto podría, realizar la foto o subir la imagen, enviándosela al servidor y que este la procese y en la respuesta que venga la imagen ya procesada con sus respectivos datos.

Sin lugar a duda me gustaría integra en la aplicación la posibilidad de crear un sistema de pedidos, en el cual el usuario tenga generada una orden de compra y esta se cumpla en función de las ordenes pendientes, integrando el sistema de pedidos tradicional que se usa en casa todas las empresas de reparto.

Por ultimo, me hubiera gustado mejorar los mapas integrados dentro de mi app, pudiendo crear rutas dinámicas, mostrando donde se encuentra el pedido en cada momento del reparto, básicamente obteniendo la posición del repartidor a cada momento.