

# Manual de programador

Por:

Martínez Ramírez Dannya Montserrat

Velázquez Flores Eduardo

## receta\_model.dart

*Descripción: En esta sección se define qué es lo que debe llevar cada receta existente y por crear de nuestra aplicación*

```
class Receta {  
  Receta(  
    Map map, {  
      this.id = '',  
      required this.nombre,  
      required this.dificultad,  
      required this.ingredientes,  
      required this.duracion,  
      required this.descripcion,  
      required this.imagen,  
      required this.preparacion,  
      required this.categoria,  
    })  
;  
  
  String id;  
  final String nombre;  
  final String dificultad;  
  final String ingredientes;  
  final String duracion;  
  final String descripcion;  
  final String imagen;  
  final String preparacion;  
  final String categoria;  
}
```

```
Map<String, dynamic> toJson() => {  
  'id':id,  
  'nombre':nombre,  
  'dificultad':dificultad,  
  'ingredientes':ingredientes,  
  'duracion' : duracion,  
  'descripcion' : descripcion,  
  'imagen': imagen,  
  'preparacion': preparacion,  
  'categoria': categoria,  
};  
  
static Receta fromJson(Map<String, dynamic> json) => Receta(  
  {},  
  nombre: json['nombre'],  
  dificultad: json['dificultad'],  
  ingredientes: json['ingredientes'],  
  duracion: json['duracion'],  
  descripcion: json['descripcion'],  
  imagen: json['imagen'],  
  preparacion: json['preparacion'],  
  categoria: json['categoria']);  
}
```

## agregar\_receta.dart

*Descripción: En esta sección se agregan los campos que el usuario debe llenar cada que quiera agregar una nueva receta a la aplicación*

```
return Scaffold(  
  appBar: AppBar(  
    title: const Text('Agregar receta'),  
    backgroundColor: Colors.orange,  
  ), // AppBar  
  body: ListView(  
    padding: const EdgeInsets.all(16), children: <Widget>[  
      TextField(  
        decoration: const InputDecoration(labelText: 'Nombre'),  
        controller: nombreController,  
      ), // TextField  
      TextField(  
        decoration: const InputDecoration(labelText: 'Descripcion'),  
        controller: descripcionController,  
      ), // TextField  
      TextField(  
        decoration: const InputDecoration(labelText: 'Ingredientes'),  
        controller: ingredientesController,  
      ), // TextField  
      TextField(  
        decoration: const InputDecoration(labelText: 'preparacion'),  
        controller: preparacionController,  
      ), // TextField  
      TextField(  
        decoration: const InputDecoration(labelText: 'duracion'),  
        controller: duracionController,  
      ), // TextField
```

```
      TextField(  
        decoration: const InputDecoration(labelText: 'Link de imagen'),  
        controller: imagenController,  
      ), // TextField  
      TextField(  
        decoration: const InputDecoration(labelText: 'categoria'),  
        controller: categoriaController,  
      ), Align( // TextField  
        alignment: Alignment.center,  
        child: ElevatedButton(  
          style: ButtonStyle(  
            backgroundColor:  
              MaterialStateProperty.all<Color>(Colors.white),  
          ), // ButtonStyle  
          onPressed: () {  
            if (nombreController.text.isNotEmpty &&  
                descripcionController.text.isNotEmpty &&  
                duracionController.text.isNotEmpty &&  
                ingredientesController.text.isNotEmpty &&  
                preparacionController.text.isNotEmpty &&  
                dificultadController.text.isNotEmpty &&  
                imagenController.text.isNotEmpty &&  
                categoriaController.text.isNotEmpty ) {  
              final nuevaReceta = Receta(  
                {},  
                nombre: nombreController.text,  
                descripcion: descripcionController.text,  
                duracion: duracionController.text,  
                ingredientes: ingredientesController.text,
```

```
                nombre: nombreController.text,  
                descripcion: descripcionController.text,  
                duracion: duracionController.text,  
                ingredientes: ingredientesController.text,  
                preparacion: preparacionController.text,  
                dificultad: dificultadController.text,  
                imagen: imagenController.text,  
                categoria: categoriaController.text  
            ); // Receta  
  
            crearReceta(nuevaReceta);  
  
            Navigator.pop(context);  
          } else {  
            final snackBar = SnackBar(  
              content: const Text('Alguno de los campos esta vacio'),  
              action:  
                SnackBarAction(label: 'Cerrar', onPressed: () {})); // SnackBar  
  
            ScaffoldMessenger.of(context).showSnackBar(snackBar);  
          }  
        },  
        child: const Text(  
          'Agregar receta',  
          style: TextStyle(color: Colors.black),  
        ), // Text  
      ), // ElevatedButton, Align
```

```
        crearReceta(nuevaReceta);  
  
        Navigator.pop(context);  
      } else {  
        final snackBar = SnackBar(  
          content: const Text('Alguno de los campos esta vacio'),  
          action:  
            SnackBarAction(label: 'Cerrar', onPressed: () {})); // SnackBar  
  
        ScaffoldMessenger.of(context).showSnackBar(snackBar);  
      }  
    },  
    child: const Text(  
      'Agregar receta',  
      style: TextStyle(color: Colors.black),  
    ), // Text  
  ), // ElevatedButton, Align  
], // <Widget>[]  
), // ListView  
); // Scaffold  
}
```

⚠ Kotlin code style  
Do you want to up  
Apply the code sty

## categoría\_page.dart

*Descripción: Esta sección se encarga de ordenar las recetas por categoría y realiza la misma función que home*

```
@override
Widget build(BuildContext context){

  final Map<String, dynamic> categoria = ModalRoute.of(context)!.settings.arguments as Map<String, dynamic>;

  return Scaffold(
    backgroundColor: Colors.white,
    body: CustomScrollView(
      slivers: <Widget>[
        appBarBack(categoria['nombre']),
        SliverList(
          delegate: SliverChildListDelegate(
            [
              Titles(categoria['nombre'], 34),
              FutureBuilder(
                future: recetasProvider.cargarRecetaCategorias(categoria['nombre']),
                initialData: [],
                builder: (BuildContext context, AsyncSnapshot snapshot){
                  return Column(
                    children: recetasListado(context, snapshot.data!), //??[]
                  ); // Column
                },
              ) // FutureBuilder
            ]
          ) // SliverChildListDelegate
        ) // SliverList
      ]
    )
  );
}
```

## editar\_receta.dart

*Descripción: En esta sección se agregan los campos que el usuario debe llenar cada que quiera editar una receta de la aplicación*



## home\_page.dart

*Descripción: Esta sección es la vista de nuestra aplicación donde se dividen imágenes por un título principal y se muestran en pantalla*

```
import 'package:flutter/material.dart';
import 'package:proyectofinal/providers/provider.dart';
import 'package:proyectofinal/widgets/app_bar.dart';
import 'package:proyectofinal/widgets/categorias.dart';
import 'package:proyectofinal/widgets/menu_lateral.dart';
import 'package:proyectofinal/widgets/populares.dart';
import 'package:proyectofinal/widgets/titles.dart';

import 'package:proyectofinal/viewModels/recetas_listado.dart';

class HomePage extends StatelessWidget{
  final GlobalKey<ScaffoldState> _scaffoldKey = GlobalKey();

  @override
  Widget build(BuildContext context){

    return Scaffold(
      drawer: menuLateral(context),
      key: _scaffoldKey,
      backgroundColor: Colors.white,
      body: CustomScrollView(
        slivers:<Widget> [
          appBar(context, _scaffoldKey),
          SliverList(
            delegate: SliverChildListDelegate(
              [
                Column(
```

```
                delegate: SliverChildListDelegate(
                  [
                    Column(
                      children:<Widget> [
                        Populares(),
                        Titles('Categorias', 24),
                        Categorias(),
                        Titles('Recetas populares', 24),
                        FutureBuilder(
                          future: recetasProvider.cargarRecetas(),
                          initData: [],
                          builder: ( BuildContext context, AsyncSnapshot<List<dynamic>> snapshot){
                            return Column(
                              children: recetasListado(context, snapshot.data!), //??[]
                            ); // Column
                          },
                        ) // FutureBuilder
                      ], // <Widget>[]
                    ), // Column
                  ]
                ) // SliverChildListDelegate
              ) // SliverList
            ], // <Widget>[]
          ) // CustomScrollView
        ); // Scaffold
      }
    }
```

## login.dart

*Descripción: Esta ventana es el clásico inicio de sesión a nuestra aplicación*

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutterfire_ui/auth.dart';
import 'package:flutter/material.dart';
import 'package:proyectofinal/pages/home_page.dart';

class Login extends StatefulWidget {
  const Login({Key? key}) : super(key: key);

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  @override
  Widget build(BuildContext context) {
    return StreamBuilder<User?>(
      stream: FirebaseAuth.instance.authStateChanges(),
      builder: (context, snapshot) {
        if (!snapshot.hasData) {
          return SignInScreen(
            subtitleBuilder: (context, action) {
              return const Padding(
                padding: EdgeInsets.only(bottom: 8),
                child: Text(
                  "Please sign in to continue.",
                ),
              );
            },
            providerConfigs: const [

```

```

        if (!snapshot.hasData) {
          return SignInScreen(
            subtitleBuilder: (context, action) {
              return const Padding(
                padding: EdgeInsets.only(bottom: 8),
                child: Text(
                  "Please sign in to continue.",
                ), // Text
              ); // Padding
            },
            providerConfigs: const [
              EmailProviderConfiguration(),
              GoogleProviderConfiguration(
                clientId:
                  "882169408481-srft6h3b9hi33ft59av3bq6fhvif888.apps.googleusercontent.com"),
            ]); // SignInScreen
          }
        return HomePage();
      },
    ); // StreamBuilder
  }
}

```

## receta\_completa.dart

*Descripción: Esta es la ventana a la que se mandará al usuario al dar click en una receta. Esta contiene todos los datos de la receta.*

```
import 'package:flutter/material.dart';
import 'package:proyectofinal/models/receta_model.dart';
import 'package:proyectofinal/widgets/app_bar_completa.dart';
import 'package:proyectofinal/widgets/titles.dart';
import 'package:proyectofinal/pages/editar_receta.dart';

class CompletaPage extends StatelessWidget {
  final String descripcion =
    'Mucho texto Mucho texto Mucho texto Mucho texto Mucho texto Mucho texto Mucho texto';
  final String ingredientes =
    'muchos ingredientes muchos ingredientes muchos ingredientes muchos ingredientes muchos ingredientes muchos ingredientes muchos ingredientes much';

  @override
  Widget build(BuildContext context) {

    final Map<String, dynamic> receta = ModalRoute.of(context)?.settings.arguments as Map<String, dynamic>;
    //ModalRoute.of(context)?.settings.arguments;

    return Scaffold(
      floatingActionButton: FloatingActionButton(
        backgroundColor: Colors.pink,
        onPressed: () async {
          Navigator.push(context, MaterialPageRoute(builder: (context) => EditarReceta(receta: receta)));
        },
        tooltip: 'Increment',
        child: const Icon(Icons.edit),
      ), // FloatingActionButton
    );
  }
}
```

```
body: CustomScrollView(
  slivers: <Widget>[
    appBarCompleta(receta['nombre'], receta['imagen']),
    SliverList(
      delegate: SliverChildListDelegate([
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: _cuerpoReceta(receta)
        ), // Padding
        Titles('Ingredientes', 28),
        SizedBox(
          height: 20.0,
        ), // SizedBox
        _textoIngredientes(receta['ingredientes']),
        Titles('Preparacion', 28),
        _textoDescripcion(receta['preparacion']),
      ]) // SliverChildListDelegate, SliverList
    ], // <Widget>[]
  ), // CustomScrollView
); // Scaffold
}
```

```
Widget _cuerpoReceta(Map<String, dynamic> receta){
  return Container(
    alignment: AlignmentDirectional.topStart,
    margin: EdgeInsets.only(left: 30.0, top: 0.0),
    child: Column(
```

```
Widget _cuerpoReceta(Map<String, dynamic> receta){
  return Container(
    alignment: AlignmentDirectional.topStart,
    margin: EdgeInsets.only(left: 30.0, top: 0.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children:<Widget> [
        Text(
          receta['nombre'],
          textAlign: TextAlign.left,
          style: TextStyle(
            fontFamily: 'Roboto',
            fontWeight: FontWeight.bold,
            color: Color.fromRGB0(15, 55, 91, 1.0),
            fontSize: 18.0
          ), // TextStyle
        ), // Text
        SizedBox(
          height: 10.0,
        ), // SizedBox
        Text(
          receta['descripcion'],
          textAlign: TextAlign.left,
          style: TextStyle(
            fontFamily: 'Roboto',
            fontWeight: FontWeight.bold,
            color: Color.fromRGB0(15, 55, 91, .8),
            fontSize: 14.0
```

```

Container(
  child: Row(
    children:<Widget> [
      Expanded(
        flex: 1,
        child: Row(
          children: <Widget>[
            Icon(Icons.alarm, color: Color.fromRGBO(243, 198, 36, 1)),
            Container(
              margin: EdgeInsets.only(left: 5.0),
              child: Text(receta['duracion'],
                style: TextStyle(
                  fontFamily: 'Roboto',
                  fontWeight: FontWeight.bold,
                  color: Color.fromRGBO(15, 55, 91, 1),
                  fontSize: 14.0,
                ), // TextStyle
              ), // Text
            ) // Container
          ], // <Widget>[]
        ) // Row
      ), // Expanded

```

```

      Expanded(
        flex: 2,
        child: Row(
          children: <Widget>[
            Icon(Icons.fastfood, color: Color.fromRGBO(243, 198, 36, 1)),
            Container(
              margin: EdgeInsets.only(left: 5.0),
              child: Text(receta['dificultad'],
                style: TextStyle(
                  fontFamily: 'Roboto',
                  fontWeight: FontWeight.bold,
                  color: Color.fromRGBO(15, 55, 91, 1),
                  fontSize: 14.0,
                ), // TextStyle
              ), // Text
            ) // Container
          ], // <Widget>[]
        ) // Row
      ) // Expanded
    ], // <Widget>[]
  ), // Row
) // Container
], // <Widget>[]
), // Column
); // Container
}

```



```

Widget _textoIngredientes(String texto) {
  return Container(
    margin: EdgeInsets.only(top: 5.0, left: 30.0, right: 30.0),
    child: Text(
      texto,
      style: TextStyle(
        fontFamily: 'Roboto',
        fontWeight: FontWeight.normal,
        color: Colors.black,
        fontSize: 13
      ), // TextStyle
    ), // Text
  ); // Container
}

```

```

Widget _textoDescripcion(String texto) {
  return Container(
    margin: EdgeInsets.only(top: 5.0, left: 30.0, right: 30.0),
    child: Text(
      texto,
      style: TextStyle(
        fontFamily: 'Roboto',
        fontWeight: FontWeight.normal,
        color: Colors.black,
        fontSize: 13
      ), // TextStyle
    ), // Text
  ); // Container
}

```