

Taller3: Integración con Firebase

Autor

Daniela Erazo Marin

Docente

David Steven Duran Vallejo

Unidad Central Del Valle

Facultad De Ingeniería

Ingeniería De Sistemas

Tuluá – Valle Del Cauca

2025

Repositorio: https://github.com/daniela-erazo-marin/curso_flutter

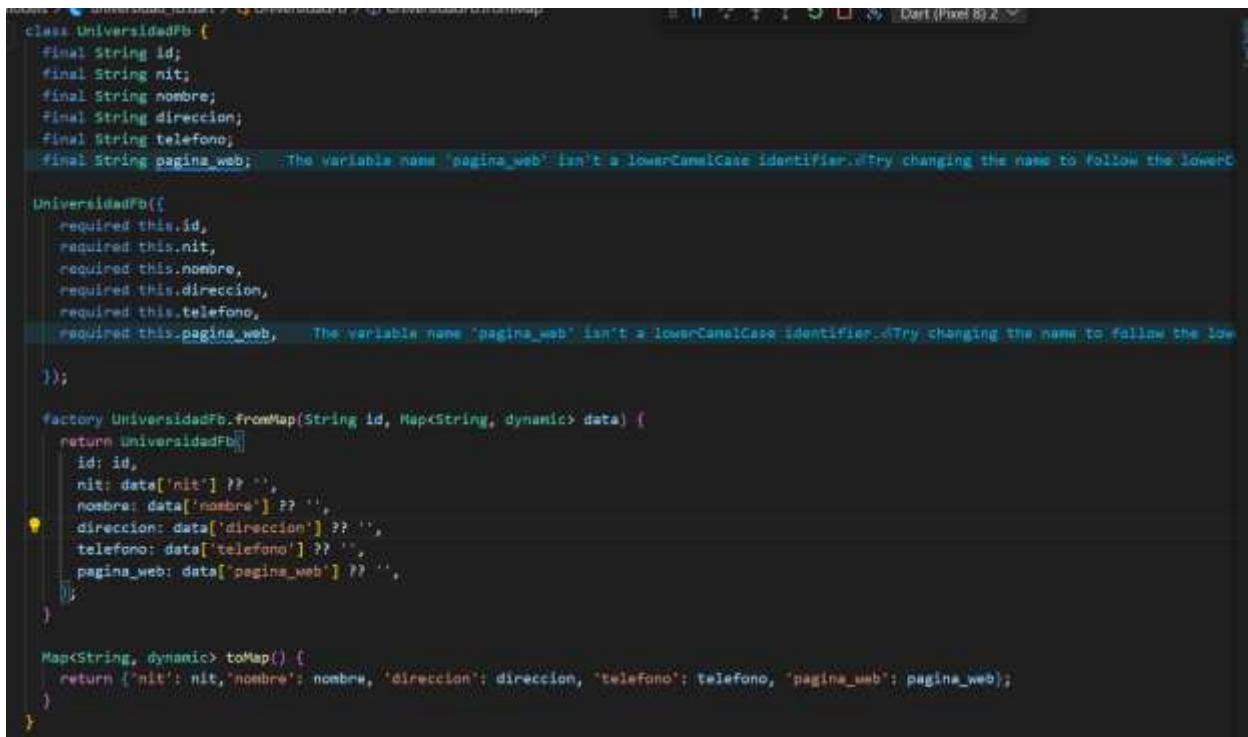
Objetivo

Desarrollar un módulo en Flutter que integre Firebase y cree/gestione la colección universidades con los campos:

- nit (string)
- nombre (string)
- direccion (string)
- telefono (string)
- pagina_web (string, URL)

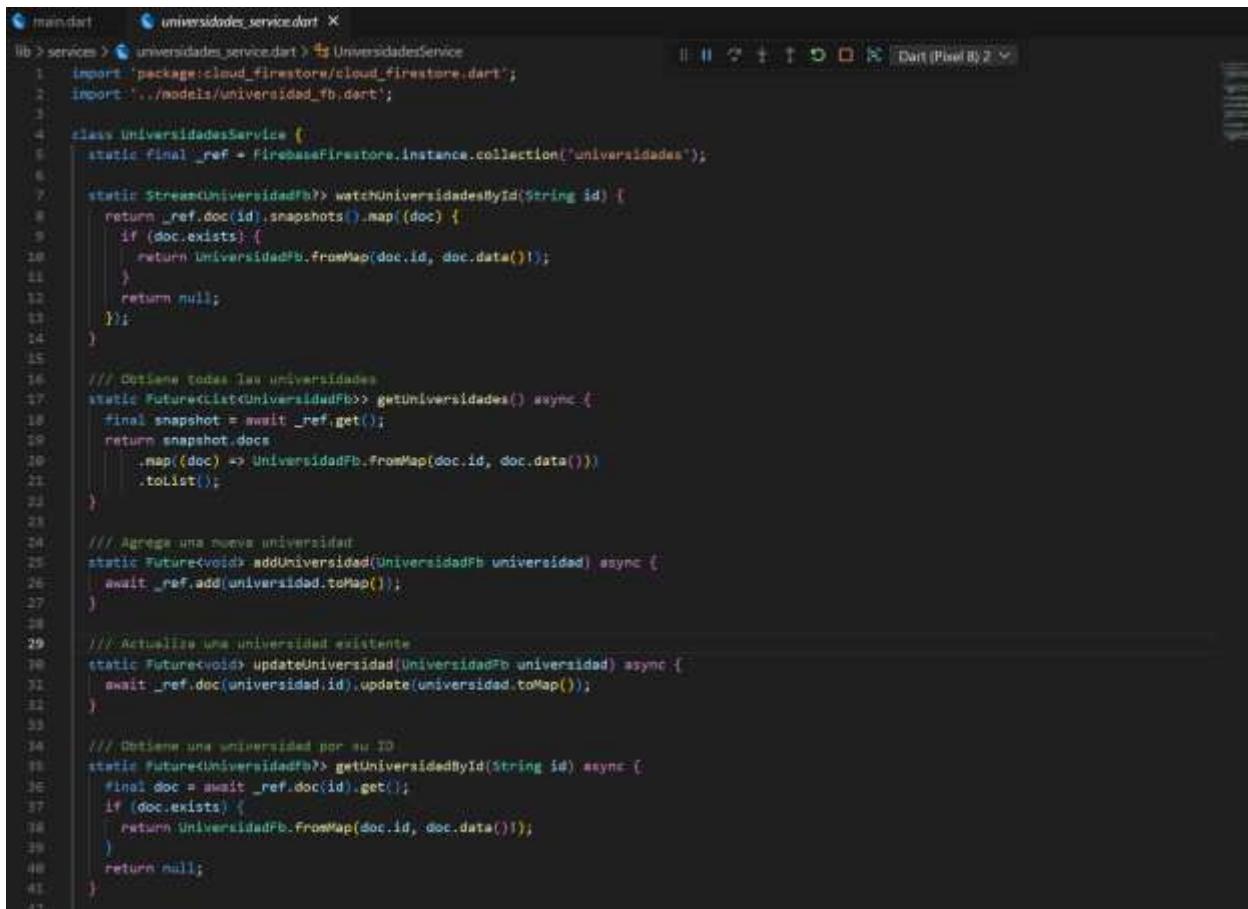
El módulo debe **conectarse a Firebase**, realizar **operaciones CRUD** y mostrar una **vista de evidencia** con datos sincronizados (listar).

Se desarrolló el modelo universidad:



```
class UniversidadFB {  
  final String id;  
  final String nit;  
  final String nombre;  
  final String direccion;  
  final String telefono;  
  final String pagina_web; // The variable name 'pagina_web' isn't a lowerCamelCase identifier. Try changing the name to follow the lowerCamelCase convention.  
  
  UniversidadFB({  
    required this.id,  
    required this.nit,  
    required this.nombre,  
    required this.direccion,  
    required this.telefono,  
    required this.pagina_web, // The variable name 'pagina_web' isn't a lowerCamelCase identifier. Try changing the name to follow the lowerCamelCase convention.  
  });  
  
  factory UniversidadFB.fromMap(String id, Map<String, dynamic> data) {  
    return UniversidadFB(  
      id: id,  
      nit: data['nit'] ?? '',  
      nombre: data['nombre'] ?? '',  
      direccion: data['direccion'] ?? '',  
      telefono: data['telefono'] ?? '',  
      pagina_web: data['pagina_web'] ?? '',  
    );  
  }  
  
  Map<String, dynamic> toMap() {  
    return {'nit': nit, 'nombre': nombre, 'direccion': direccion, 'telefono': telefono, 'pagina_web': pagina_web};  
  }  
}
```

Se desarrolló el servicio universidad:

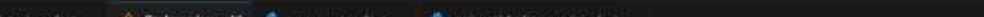


The screenshot shows a code editor window with the file 'universidades_service.dart' open. The code is written in Dart and interacts with Firestore. It includes methods for getting universities by ID and a stream, adding a new university, updating an existing one, and getting a university by its ID.

```
main.dart  universidades_service.dart ×
lib> services > universidades_service.dart > UniversidadesService
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import '../models/universidad_fb.dart';
3
4 class UniversidadesService {
5   static final _ref = FirebaseFirestore.instance.collection('universidades');
6
7   static Stream<UniversidadFB> watchUniversidadesById(String id) {
8     return _ref.doc(id).snapshots().map((doc) {
9       if (doc.exists) {
10         return UniversidadFB.FromMap(doc.id, doc.data());
11       }
12       return null;
13     });
14   }
15
16   /// Obtiene todas las universidades
17   static Future<List<UniversidadFB>> getUniversidades() async {
18     final snapshot = await _ref.get();
19     return snapshot.docs
20       .map<doc> => UniversidadFB.FromMap(doc.id, doc.data())
21       .toList();
22   }
23
24   /// Agrega una nueva universidad
25   static Future<void> addUniversidad(UniversidadFB universidad) async {
26     await _ref.add(universidad.toMap());
27   }
28
29   /// Actualiza una universidad existente
30   static Future<void> updateUniversidad(UniversidadFB universidad) async {
31     await _ref.doc(universidad.id).update(universidad.toMap());
32   }
33
34   /// Obtiene una universidad por su ID
35   static Future<UniversidadFB> getUniversidadById(String id) async {
36     final doc = await _ref.doc(id).get();
37     if (doc.exists) {
38       return UniversidadFB.FromMap(doc.id, doc.data());
39     }
40     return null;
41   }
42 }
```

Alcance funcional

1. Conexión con Firebase.



```
universidades_fb_list_view.dart  firebase.json X app_router.dart  universidades_service.dart  
! firebase.json  
1  {"flutter":{"platforms":{"android":{"default":{"projectId":"universidades-5cc96","appId":"1:188483447759:android:d2882cce#8231a85994ba","m...}}
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "CURSO_FLUTTER". It includes folders for themes, auth, categoria_fb, cdt, ciclo_vida, future, home, isolate, meal, paso_parametros, paso_parametros2, pokemons, timer, universidad_fb, widgets_demo, and widgets. Specific files like categoria_fb_form_view.dart, categoria_fb_list_view.dart, cdt_list_view.dart, and firebase_options.dart are also listed.
- Code Editor:** The main editor area displays the content of the firebase_options.dart file. The code defines two static const variables: macos and windows, each representing a FirebaseOptions object with specific configuration values for apiKey, appId, messagingSenderId, projectId, storageBucket, and iosBundleId.

```
lib > firebase_options.dart > ...  
68  
69 static const FirebaseOptions macos = FirebaseOptions(  
70   apiKey: 'AIzaSyCAP8i4LM5iNEDWp9rI8gu5Zq3RbcpbLRU',  
71   appId: '1:188483447759:ios:c838ad83F73d256f5994ba',  
72   messagingSenderId: '188483447759',  
73   projectId: 'universidades-5cc96',  
74   storageBucket: 'universidades-5cc96.firebaseio.storage.app',  
75   iosBundleId: 'com.example.cursoFlutter',  
76 );  
77  
78 static const FirebaseOptions windows = FirebaseOptions(  
79   apiKey: 'AIzaSyA8JgMzrZU17QSFUVY79860AJ77eYHbjg',  
80   appId: '1:188483447759:web:34b158c8659383a15994ba',  
81   messagingSenderId: '188483447759',  
82   projectId: 'universidades-5cc96',  
83   authDomain: 'universidades-5cc96.firebaseioapp.com',  
84   storageBucket: 'universidades-5cc96.firebaseio.storage.app',  
85 );  
86  
87
```

2. Colección universidades en Firestore.

Agregar un documento

Ruta superior
/universidades

ID de documento ⓘ
8oUdnfFEEosYoEKbsZDA

| Campo | Tipo |
|-----------|---------------|
| nit | = string |
| String | 890.123.456-7 |
| nombre | = string |
| String | UCEVA |
| dirección | = string |

Cerrar Guardar

universidades > Cloud Firestore > Base de datos

Protege tus recursos de Cloud Firestore contra los abusos, como fraudes de facturación o phishing. Configurar la Verificación de aplicaciones

Vista del panel Compositor de consultas

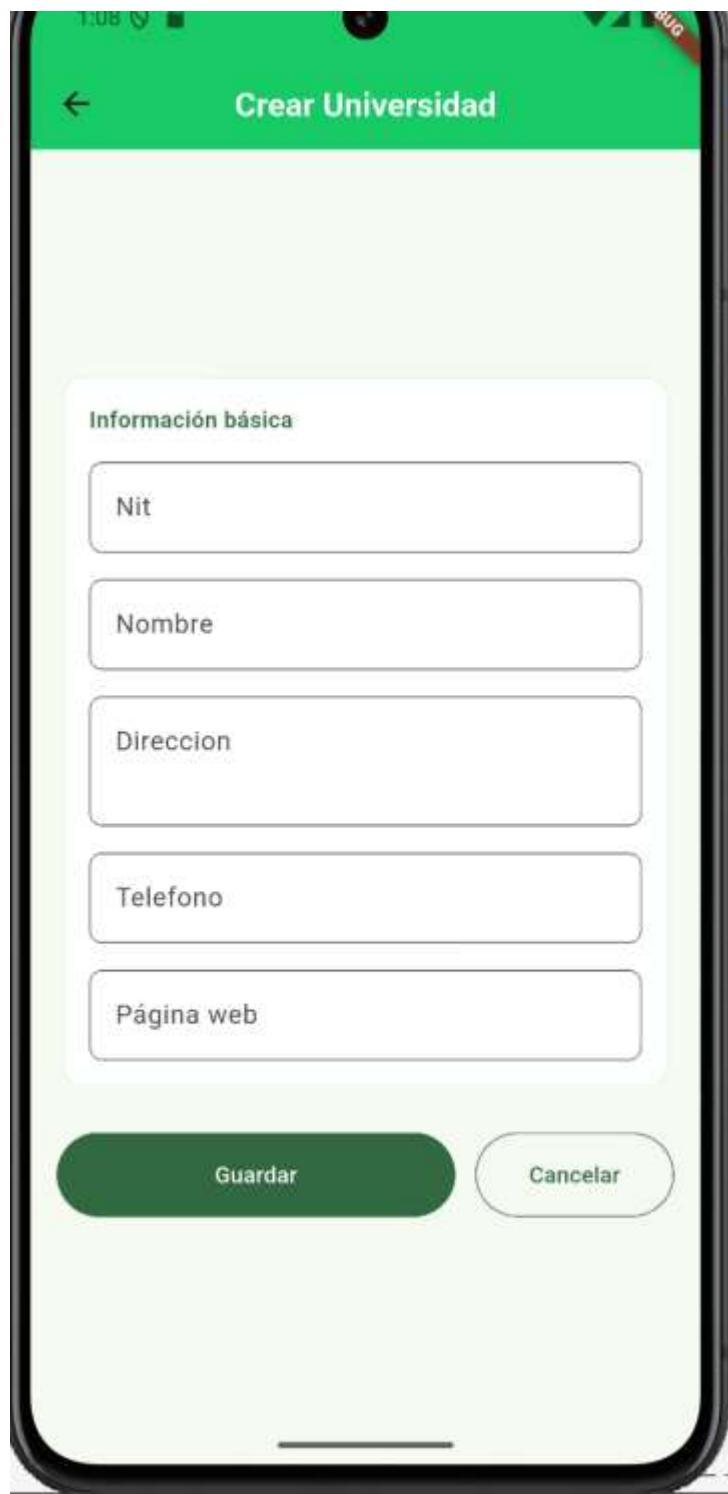
Más funciones en Google Cloud

(default) > universidades > 8oUdnfFEEosYoEKbsZDA

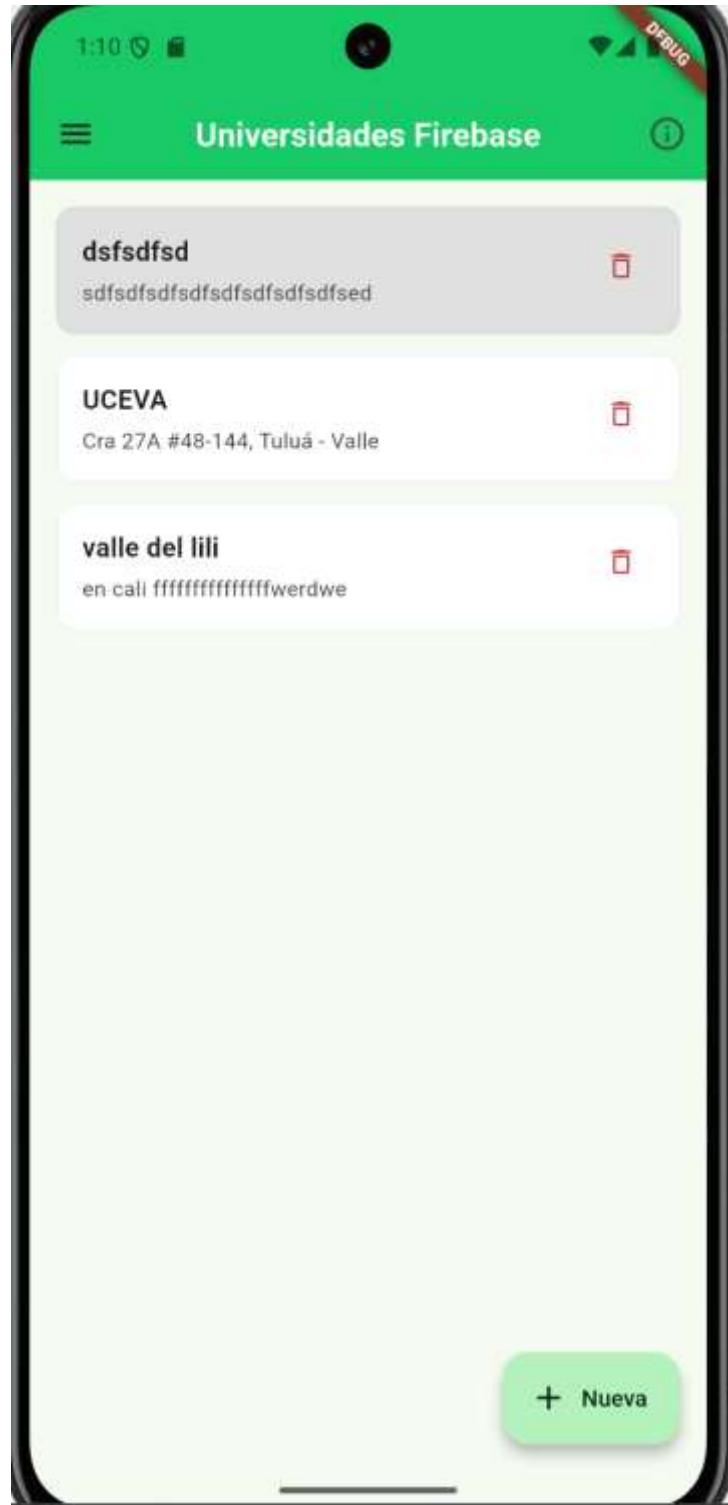
| Universidad | Dirección | Página web | Teléfono |
|-------------|--------------------------------|--------------------------|--------------------|
| UCEVA | Dia 27A #46-146, Tuluá - Valle | https://www.uceva.edu.co | 3157 652 2342/2027 |

3. **Crear** (formulario) y **listar** universidades en tiempo real.

Crear:



Listar:



4. Validación básica de campos (no vacíos y URL válida en pagina_web).

```
16 > views > universidad_fb > universidad_fb_form.dart > _UniversidadFBFormViewState.dart
17
18 class _UniversidadFBFormViewState extends State<UniversidadFBFormView> {
19   Widget _buildFormulario(BuildContext context, {required String id}) {
20
21     textCapitalization: TextCapitalization.words,
22     validator: (value) {
23       if (value == null || value.trim().isEmpty) {
24         return 'El teléfono es requerido';
25       }
26
27       if (value.trim().length < 13) {
28         return 'El teléfono debe tener al menos 13 caracteres';
29       }
30
31       return null;
32     },
33   }, // TextFormField
34   const SizedBox(height: 35),
35   TextFormField(
36     controller: _pagina_webController,
37     decoration: InputDecoration(
38       labelText: 'Página web',
39       hintText: 'Ingrésa la página web de la universidad',
40       border: OutlineInputBorder(
41         borderRadius: BorderRadius.circular(8),
42       ), // OutlineInputBorder
43       inputDecoration: InputDecoration(
44         keyboardType: TextInputType.url,
45         textCapitalization: TextCapitalization.none,
46         validator: (value) {
47           if (value == null || value.trim().isEmpty) {
48             return 'La página web es requerida';
49           }
50
51           final url = value.trim();
52
53           // Verifica si es una URL válida
54           final uri = Uri.tryParse(url);
55
56           if (uri == null || !uri.hasScheme || !uri.hasAuthority) {
57             return 'Ingrésa una URL válida (ejemplo: https://www.universidad.edu.co)';
58           }
59
60           return null;
61         },
62       ), // InputDecoration
63     ),
64   ), // TextFormField
```

Validaciones:



Validación de correo:



5. Vista de evidencia mostrando datos leídos desde la BD en vivo.

The image consists of two side-by-side screenshots. The left screenshot shows the Google Cloud Firestore console interface. A document in the 'universidades' collection is being edited. The document contains fields: 'id' (890.123.456-7), 'Nombre' (UCEVA), 'Direccion' (Cra 27A #48-144, Tuluá - Valle), 'Telefono' (+57 602 2242202), and 'pagina_web' (https://www.univalle.edu.co). The right screenshot shows a mobile application titled 'Crear Universidad' (Create University) with a form mirroring these fields. The bottom right of the mobile screen has a green 'Guardar' (Save) button. The right screenshot also shows a list of universities in the background, with one item selected: 'UCEVA' (Cra 27A #48-144, Tuluá - Valle). The bottom right of the mobile screen has a green '+ Nuevo' (New) button.

Universidades - Base de datos

Configure la Verificación de aplicaciones

Información básica

id: 890.123.456-7

Nombre: UCEVA

Dirección: Cra 27A #48-144, Tuluá - Valle

Teléfono: +57 602 2242202

Página web: https://www.univalle.edu.co

Guardar Cancelar

Universidades - Base de datos

Configure la Verificación de aplicaciones

Universidades Firebase

defdfdsf

UCEVA

valle del III

UCEVA

+ Nuevo

Eliminación:



The image shows a split-screen view. On the left is the Firebase Cloud Firestore console interface, and on the right is a mobile application interface.

Firebase Cloud Firestore Console:

- Project:** universidades - Base de datos
- Collection:** universidades
- Document:** 7WWM6L0ChLn...
- Fields:**
 - name: "Universidad Nacional de Colombia"
 - address: "Cra 45 #25-89, Sede Bogotá"
 - city: "Bogotá D.C."
 - country: "Colombia"
 - email: "un@unal.edu.co"
 - fax: "571-3165000000"
 - phone: "571-3165000000"
 - pagina_web: "https://unal.edu.co"

Mobile Application (Universidades Firebase):

- Universidad Nacional de Colombia**
Cra 45 #25-89, Sede Bogotá
- UCEVA**
Cra 27A #48-144, Tuluá - Valle
- Universidad del Valle**
Calle 13 #100-00, Sede Medellín
- Valle del Lili**
en cas: 111111111111111111
- UCEVA**
Cra 27A #48-144, Tuluá - Valle

A green button at the bottom right of the mobile app screen says "+ Nueva".

Descripción técnica del taller

Durante el desarrollo del taller se construyó una aplicación en Flutter orientada a la gestión de universidades, implementando una arquitectura modular basada en la separación por capas: modelo, servicios e interfaces gráficas.

1. Modelo de datos (Entidad Universidad)

Se diseñó la clase Universidad como modelo base, la cual contiene los atributos principales:

- `nit` (string)
- `nombre` (string)
- `direccion` (string)
- `telefono` (string)
- `pagina_web` (string, URL)

Esta estructura fue implementada en Dart y métodos `fromMap()` y `toMap()` para permitir su integración con Firebase.

2. Servicio de datos (UniversidadService)

Se creó un servicio encargado de la comunicación con la base de datos cloud **Firebase**, gestionando las operaciones **CRUD**:

- `getUniversidades()` → obtiene la lista de universidades.
- `addUniversidad()` → agrega un nuevo registro.
- `updateUniversidad()` → actualiza los datos existentes.
- `deleteUniversidad()` → elimina un registro.

El servicio utiliza la librería `cloud_firestore` para manejar la persistencia en la nube de manera asíncrona, garantizando integridad y respuesta en tiempo real.

3. Configuración de Firebase

El proyecto se integró con **Firebase** mediante la consola de desarrollador, registrando la aplicación y descargando los archivos de configuración `firebase_options.dart` y `firebase.json`.

Se habilitó **Firestore Database** en modo de prueba y se crearon las colecciones necesarias (universidades).

La estructura de cada documento contiene los campos mencionados en el modelo, con sus tipos de datos definidos (String).

4. Interfaz gráfica (UI)

Se desarrolló una interfaz visual responsiva utilizando componentes de **Material Design**:

- Formulario de registro con `TextField` y validadores personalizados (por ejemplo, validación de formato para correos y URLs).
- Listado tipo grid implementado con `GridView` y `_buildGridContent()`, mostrando los datos de cada universidad con un diseño limpio, botones de acción (`IconButton`) y manejo de eventos (`onPressed`, `showDialog`, etc.).
- Uso de `ColorScheme` para adaptar los colores a los temas claro/oscuro del sistema.

5. Validaciones técnicas

Se implementaron validaciones en los campos del formulario para garantizar la integridad de los datos:

- El nombre y dirección no pueden estar vacíos.
- La página web se valida usando Uri.tryParse() para asegurar un formato correcto de URL.

6. Pruebas con datos ficticios

Finalmente, se cargaron registros de prueba en la colección de Firebase con información simulada de universidades colombianas (nit, nombre, dirección, teléfono y página web) para verificar el correcto funcionamiento del flujo completo de registro, lectura, actualización y eliminación.