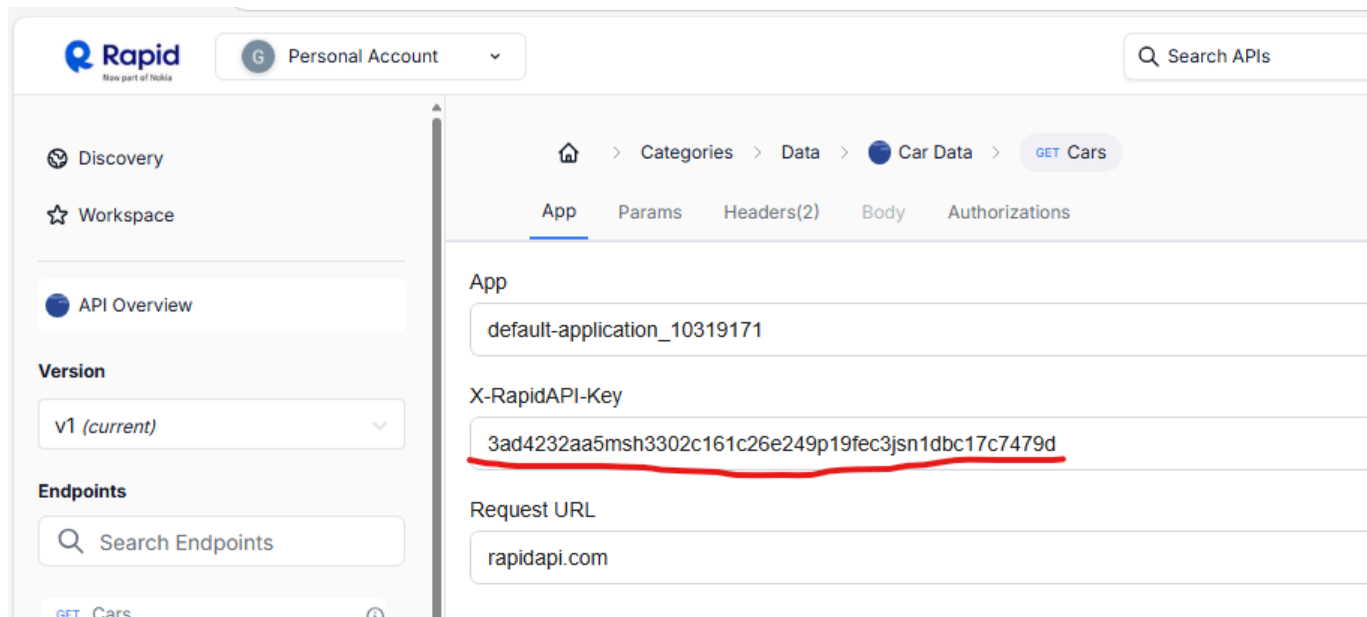
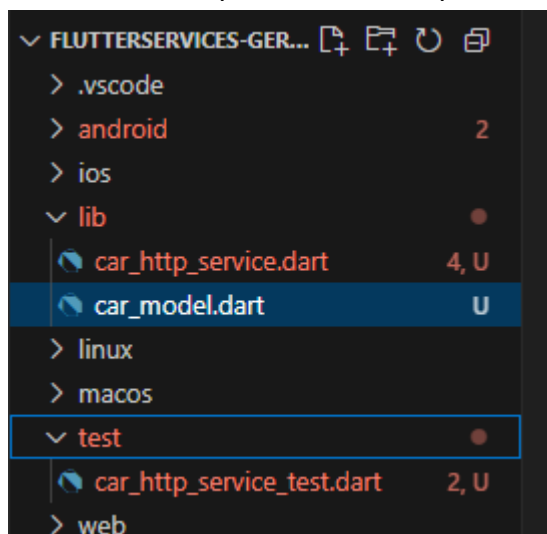


1. Exercici 1 Accés a un servei (pas a pas)

Ens hem registrat a rapiapi generant una clau que més endavant l'utilitzarem:



Hem creat la carpeta "test" i hem posat els seus respectius arxius:



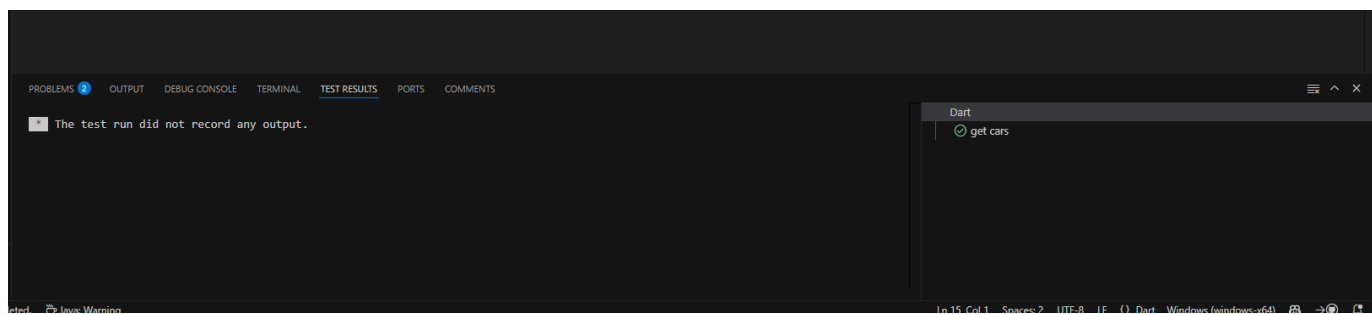
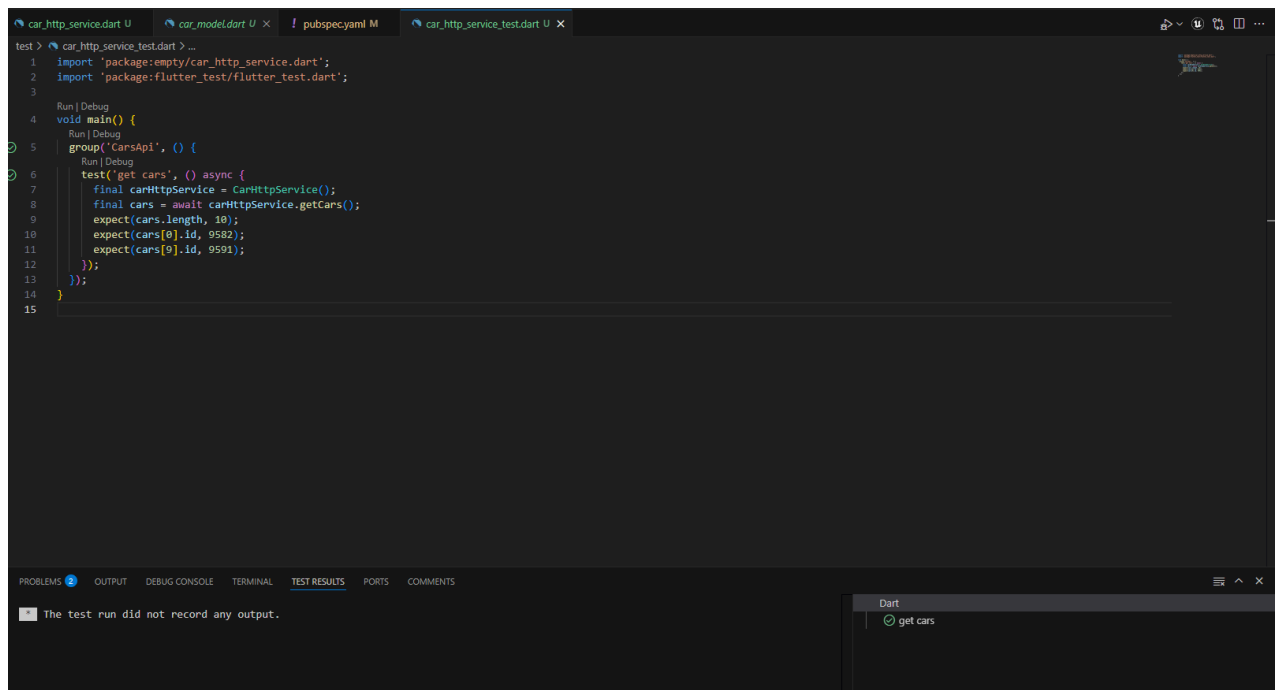
Hem canviat la clau a l'arxiu pubspec.yaml per a la de l'aplicació:

```
lib > car_http_service.dart > CarHttpService
1 import 'package:empty/car_model.dart';
2 import 'package:http/http.dart' as http;
3 // import 'package:rentcar/model/brand_model.dart';
4 // import 'package:rentcar/model/car_model.dart';
5 // import 'package:rentcar/utils/const_app.dart';
6
7 class CarHttpService {
8   final String _serverUrl = "https://car-data.p.rapidapi.com";
9   final String _headerKey =
10     "3ad4232aa5msh3302c161c26e249p19fec3jsn1dbc17c7479d";
11   final String _headerHost = "car-data.p.rapidapi.com";
12
13   /*
14   Obtenir la llista de cotxes
15   */
16   Future<List<CarsModel>> getCars() async {
17     //URL de l'endpoint: És la URL del servidor, més la URL de l'endpoint
18     var uri = Uri.parse(
19       "$_serverUrl/cars"); //Aquí hem de canviar el text de l'exemple per /cars
20
```

I també als imports de cada arxiu:

```
car_http_service.dart U • car_model.dart U ! pubspec.yaml M car_http_service_test.dart U
lib > car_http_service.dart > CarHttpService
1 import 'package:empty/car_model.dart';
2 import 'package:http/http.dart' as http;
3 // import 'package:rentcar/model/brand_model.dart';
4 // import 'package:rentcar/model/car_model.dart';
5 // import 'package:rentcar/utils/const_app.dart';
6
7 class CarHttpService {
```

Hem executat per fer un test unitari i aquest és el seu resultat:



2. Exercici 2 Provider integrar vista i model

1. Què és el que aprendrem a aquest vídeo? En este vídeo aprenderemos a manejar Un estado global de la aplicación en este paquete poderosísimo que se llama Provider

En este vídeo aprenderemos a manejar Un estado global de la aplicación en este paquete poderosísimo que se llama Provider

2. Quina comanda es fa servir al vídeo per a crear una nova aplicació flutter?

`flutter create nom_aplicació`

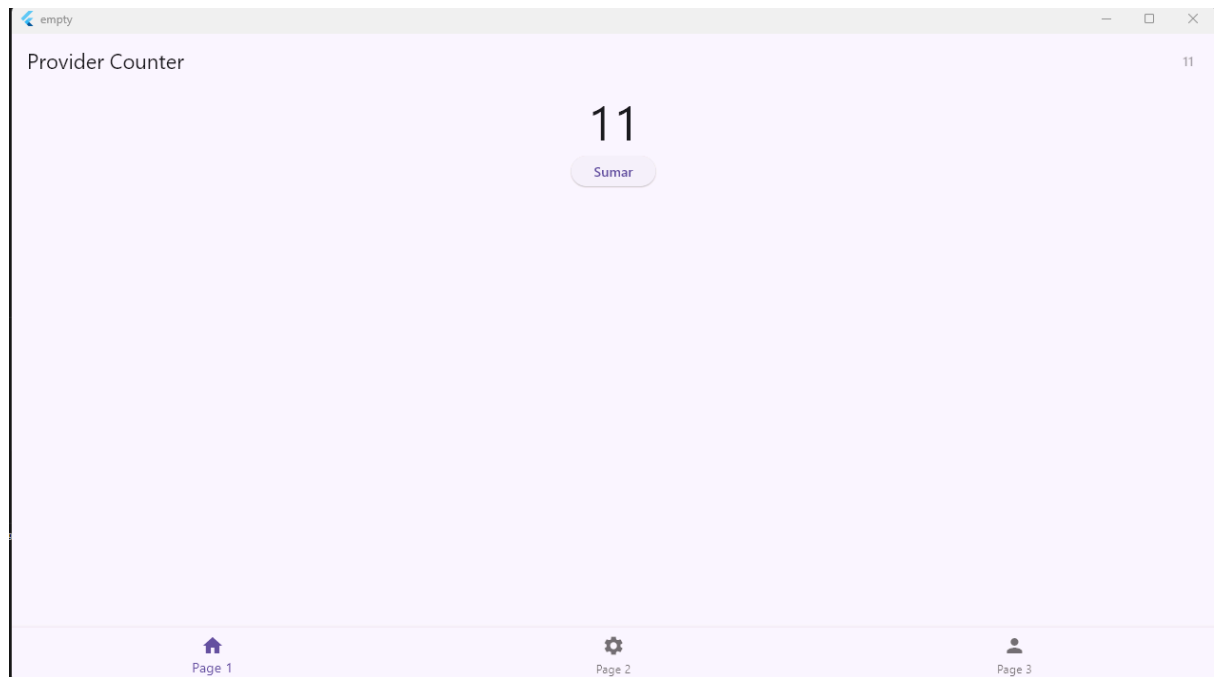
3. Com es treu el debugger banner? Afegeix la línia de codi que fan servir al vídeo per treure'l

`debugShowCheckedModeBanner: false,`

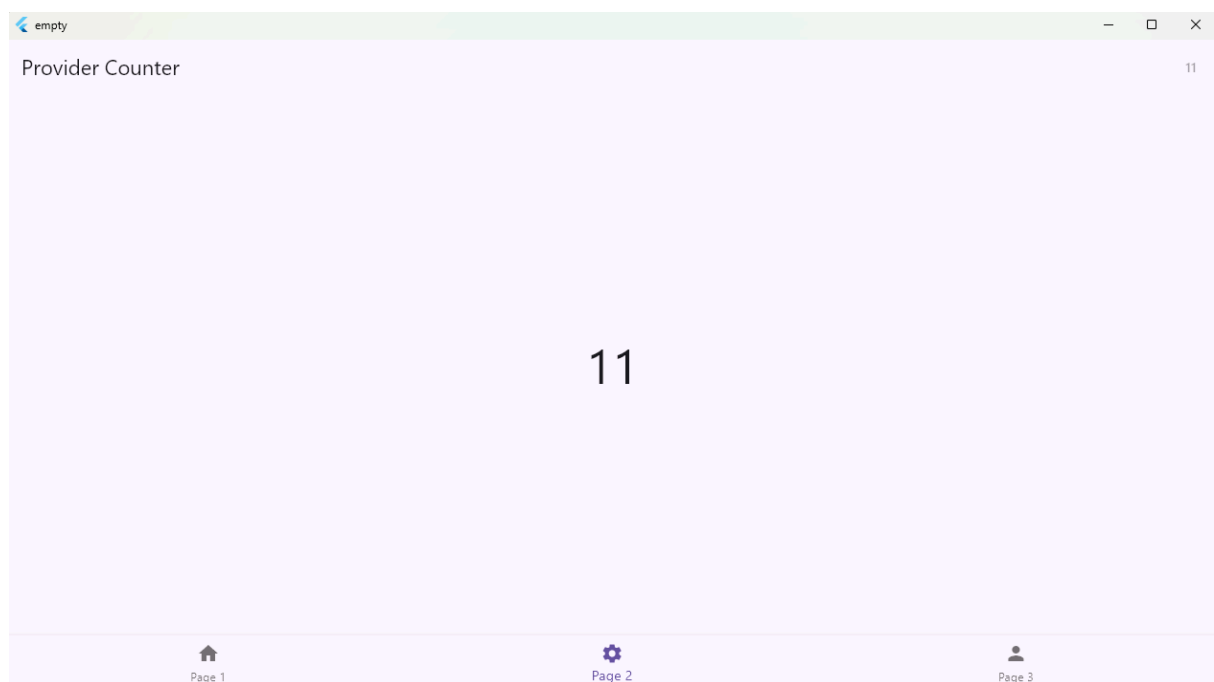
S'instal·la el provider:

```
History restored
PS C:\Users\gerar\Downloads\flutter_camera_app\flutterservices-geroo22> flutter pub add provider[]
ed. Java: Warning
```

Es puja el contador



I a la segona pàgina té el mateix nombre que el provider



page1.dart:

Es fan els imports necessaris i es fa el codi de la primera pagina, el provider una de les coses importants son aquestes:

- `context.watch<CounterProvider>()` s'utilitza per obtenir el valor actual del counter i escoltar-ne els canvis, de manera que es redibuixi el Text quan el valor canviï.
- `context.read<CounterProvider>()` s'utilitza per cridar la funció `increment()`, però no redibuixa automàticament el widget.

```
car_http_service.dart  pubspec.yaml  car_http_service_test.dart  page1.dart  acudit_controlador.dart  acudit.dart  acudit_controlador.dart  lib > Exercici2 > pages > page1.dart > ...
1  import 'package:empty/Exercici2/providers/counter_provider.dart';
2  import 'package:flutter/material.dart';
3  import 'package:provider/provider.dart';
4
5
6  class Page1 extends StatefulWidget {
7    const Page1({Key? key}) : super(key: key);  Parameter 'key' could be a super parameter. @Trying converting 'key' to a super parameter.
8
9    @override
10    State<Page1> createState() => _Page1State();
11  }
12
13  class _Page1State extends State<Page1> {
14
15
16    @override
17    Widget build(BuildContext context) {
18      return Center(
19        child: Column(
20          children: [
21            Text(context.watch<CounterProvider>().counter.toString(), style: const TextStyle(fontSize: 50)),
22            ElevatedButton(
23              onPressed: () {
24                setState(() {
25                  context.read<CounterProvider>().increment();
26                });  Unnecessary empty statement. @Try removing the empty statement or restructuring the code.
27              },
28            ),
29            child: const Text('Sumar'),
30          ],  // ElevatedButton
31        ),  // Column
32      );  // Center
33    }
34  }
35
36 }
```

page2.dart:

context.watch<CounterProvider>().counter assegura que Page2 es redibuixi automàticament quan el valor del comptador canviï.

Només mostra informació, sense modificar estat

```
page2.dart U X
lib > Exercici2 > pages > page2.dart > ...
1 import 'package:empty/Exercici2/providers/counter_provider.dart';
2 import 'package:flutter/material.dart';
3 import 'package:provider/provider.dart';
4
5 class Page2 extends StatefulWidget {
6   const Page2({Key? key}) : super(key: key); Parameter 'key' could be a super parameter.ðTrying converting 'key' to a super parameter.
7
8   @override
9   State<Page2> createState() => _Page2State();
10 }
11
12 class _Page2State extends State<Page2> {
13   @override
14   Widget build(BuildContext context) {
15     return Center(
16       child: Text(context.watch<CounterProvider>().counter.toString(), style: const TextStyle(fontSize: 50)),
17     ); // Center
18   }
19 }
```

main.dart:

ChangeNotifierProvider és qui proporciona CounterProvider a tota l'aplicació, assegurant que qualsevol widget pugui accedir a l'estat del comptador.

MyApp com a StatelessWidget: No necessita ser StatefulWidget perquè no gestiona estat intern. Simplement configura i retorna l'MaterialApp.

HomeScreen() és la primera pantalla de l'aplicació, i s'encarrega de navegar entre Page1 i Page2. S'ha de revisar el seu codi per veure com s'integren aquestes pàgines.

```
lib > Exercici2 > main.dart > MyApp > build
1 import 'package:flutter/material.dart';
2 import 'package:provider/provider.dart';
3 import 'package:empty/Exercici2/screens/home_screen.dart';
4 import 'package:empty/Exercici2/providers/counter_provider.dart';
5
6 Run | Debug | Profile
7 void main() {
8   runApp(
9     ChangeNotifierProvider(
10       create: (_) => CounterProvider(),
11       child: MyApp(), // Use 'const' with the constructor to improve performance. Try adding the 'const' keyword to the constructor invocation.
12     ); // ChangeNotifierProvider
13   }
14
15 class MyApp extends StatelessWidget {
16   const MyApp({super.key});
17   @override
18   Widget build(BuildContext context) {
19     return MaterialApp(
20       debugShowCheckedModeBanner: false,
21       title: 'Provider Counter',
22       theme: ThemeData(
23         primarySwatch: Colors.purple,
24       ), // ThemeData
25       home: const HomeScreen(),
26     ); // MaterialApp
27   }
28 }
29
```

3. Exercici 3

4. Exercici 4 (acudits)

Es fa el fitxer del model (acudit.dart) representa un acudit amb setup i punchline

```

1 // lib/models/acudit.dart
2 class Joke {
3   final String setup;
4   final String punchline;
5
6   Joke({required this.setup, required this.punchline});
7
8   factory Joke.fromJson(Map<String, dynamic> json) {
9     return Joke(
10       setup: json['setup'] ?? '',
11       punchline: json['punchline'] ?? '',
12     );
13   }
14 }

```

Es fa el controlador (acudit_controlador.dart). Fa una crida HTTP a l'API i retorna un acudit.:

```

lib > acudits > controllers > acudit_controlador.dart > JokeController
1 // lib/controllers/acudit_controlador.dart
2 import 'dart:convert';
3 import 'package:http/http.dart' as http; Target of URI doesn't exist: 'package:ht
4 import '../models/acudit.dart'; Target of URI doesn't exist: '../models/acu
5
6 class JokeController {
7   static const String apiUrl = "https://api.sampleapis.com/jokes/goodJokes";
8
9   Future<Joke> fetchJoke() async { The name 'Joke' isn't a type, so it can't be us
10     final response = await http.get(Uri.parse(apiUrl));
11
12     if (response.statusCode == 200) {
13       List<dynamic> jokes = jsonDecode(response.body);
14       if (jokes.isNotEmpty) {
15         final randomJoke = jokes[0]; // Pots canviar això per un acudit aleatori
16         return Joke.fromJson(randomJoke); Undefined name 'Joke'. Try correcting t
17       }
18     }
19     throw Exception("No s'ha pogut carregar l'acudit");
20   }
21 }

```


I també és fa la vista (acudit_screen.dart). Mostra l'acudit i permet carregar-ne un altre amb un botó. :

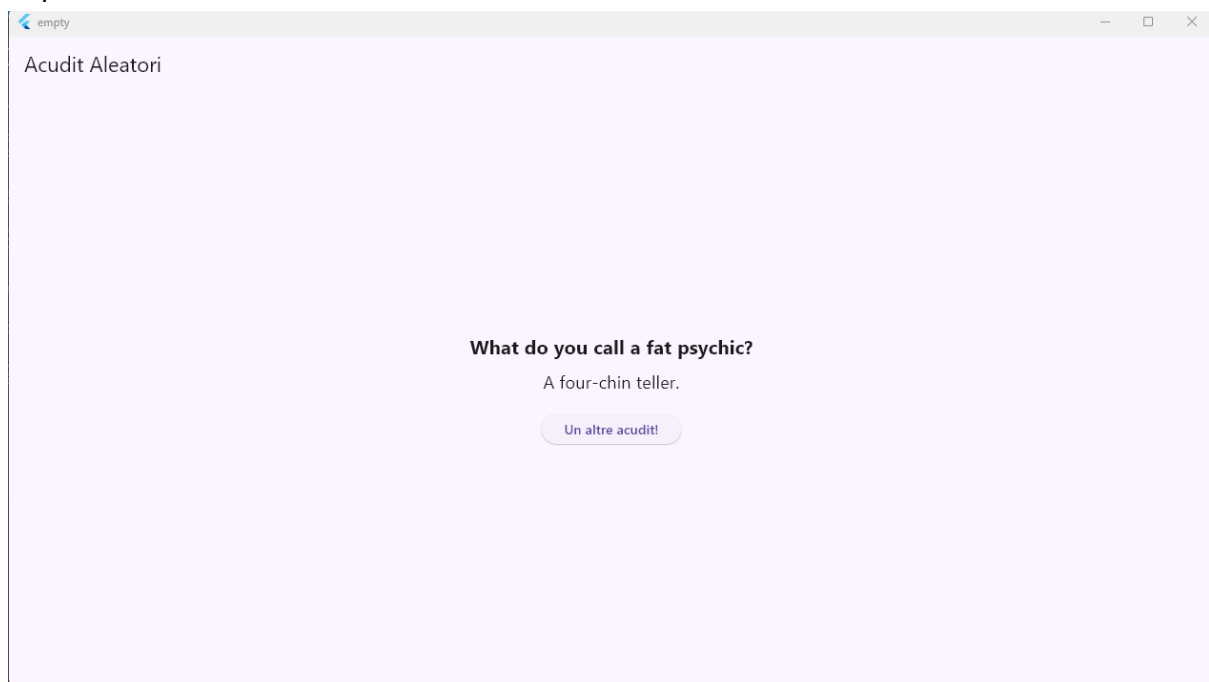
```
lib > acudits > views > acudit_screen.dart > ...
1 // lib/views/joke_screen.dart
2 import 'package:flutter/material.dart'; Target of URI doesn't exist: 'package:flutter/material.dart'.@Try creating the file referenced by the URI, or try using
3 import '../controllers/acudit_controlador.dart';
4 import '../models/acudit.dart';
5
6 class JokeScreen extends StatefulWidget { Classes can only extend other classes.@Try specifying a different superclass, or removing the extends clause.
7   @override
8   _JokeScreenState createState() => _JokeScreenState(); The method doesn't override an inherited method.@Try updating this class to match the superclass, or re
9 }
10
11 class _JokeScreenState extends State<JokeScreen> { Classes can only extend other classes.@Try specifying a different superclass, or removing the extends clause
12   final JokeController _controller = JokeController();
13   Joke? _currentJoke;
14   bool _isLoading = false;
15
16   void _getNewJoke() async {
17     setState(() { The method 'setState' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or defining a
18       _isLoading = true;
19     });
20
21     try {
22       Joke joke = await _controller.fetchJoke();
23       setState(() { The method 'setState' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or defining
24         _currentJoke = joke;
25       });
26     } catch (e) {
27       setState(() { The method 'setState' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or defining
28         _currentJoke = Joke(setup: "Error carregant l'acudit", punchline: "");
29       });
30     }
31
32     setState(() { The method 'setState' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or defining a
33       _isLoading = false;
34     });
35   }
36
37   @override
38   void initState() { The method doesn't override an inherited method.@Try updating this class to match the superclass, or removing the override annotation.
39     super.initState(); The method 'initState' isn't defined in a superclass of '_JokeScreenState'.@Try correcting the name to the name of an existing method, o
40     _getNewJoke();
41   }
42
43   @override
44   Widget build(BuildContext context) { Undefined class 'BuildContext'.@Try changing the name to the name of an existing class, or creating a class with the nam
45     return Scaffold( The method 'Scaffold' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or definin
46       appBar: AppBar(title: Text("Acudit Aleatori")), The method 'Text' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of a
47       body: Center( The method 'Center' isn't defined for the type '_JokeScreenState'.@Try correcting the name to the name of an existing method, or defining a
```

Amb el model, vista i controlador, podem configura l'aplicació amb el main per a fer funcionar la aplicació (main.dart):

```
lib > acudits > main.dart > MyApp
1 // lib/main.dart
2 import 'package:flutter/material.dart'; Target of URI doesn't exist:
3 import 'views/acudit_screen.dart';
4
Run | Debug | Profile
5 void main() {
6   runApp(MyApp()); The function 'runApp' isn't defined. Try importin
7 }
8
9 class MyApp extends StatelessWidget { Classes can only extend other
10   @override
11   Widget build(BuildContext context) { Undefined class 'BuildContext'
12     return MaterialApp( The method 'MaterialApp' isn't defined for t
13       debugShowCheckedModeBanner: false,
14       title: 'Acudits Flutter',
15       theme: ThemeData(primarySwatch: Colors.blue), Undefined name 'C
16       home: JokeScreen(),
17     );
18   }
19 }
```

Resultat:

Captura 1



Captura 2

