

Ficha 7

Main.dart:

```
import 'package:flutter/material.dart';

import 'home_page.dart';

import 'descarregar_produtos.dart';

import 'listar_produtos.dart';


void main() {

  runApp(const MyApp());

}


class MyApp extends StatelessWidget {

  const MyApp({Key? key}) : super(key: key);


  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Gestão de Produtos',

      theme: ThemeData(

        primarySwatch: Colors.blue,

        visualDensity: VisualDensity.adaptivePlatformDensity,

      ),

      initialRoute: '/',

      routes: {

        '/': (context) => const HomePage(),

        '/download': (context) => const DownloadProductsPage(),
```

```
      '/list': (context) => const ListProductsPage(),  
    }, ); }}
```

Home_page.dart:

```
import 'package:flutter/material.dart';
```

```
import 'app_drawer.dart';
```

```
class HomePage extends StatelessWidget {  
  const HomePage({Key? key}) : super(key: key);
```

```
  @override
```

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

```
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Página Principal'),  
      ),  
      drawer: AppDrawer(),  
      body: Center(  
        child: const Text(  
          'Gestão de produtos',  
          textAlign: TextAlign.center,  
          style: TextStyle(fontSize: 18),  
        ),      ), ); }}
```

Descarregar_produtos.dart:

```
import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

import 'dart:convert';

import 'package:sqflite/sqflite.dart';

import 'database_helper.dart';

import 'app_drawer.dart';

class DownloadProductsPage extends StatefulWidget {

  const DownloadProductsPage({Key? key}) : super(key: key);

  @override

  _DownloadProductsPageState createState() => _DownloadProductsPageState();

}

class _DownloadProductsPageState extends State<DownloadProductsPage> {

  bool _isLoading = false;

  String _message = "";

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text('Descarregar Produtos'),

      ),

      drawer: AppDrawer(),

      body: Center(

        child: Column(
```

```
mainAxisAlignment: MainAxisAlignment.center,
children: [
  if (_isLoading)
    const CircularProgressIndicator()
  else
    ElevatedButton(
      onPressed: _downloadProducts,
      child: const Text('Descarregar Produtos'),
    ),
  const SizedBox(height: 20),
  Text(
    _message,
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: 16,
      color: _message.contains('Erro') ? Colors.red : Colors.green,
    ),
  ),
],
),
); }
```

```
Future<void> _downloadProducts() async {  
  setState(() {  
    _isLoading = true;  
    _message = 'A descarregar produtos...';  
  });  
  try {  
    // Fazer a requisição à API  
    final response = await http.get(Uri.parse('https://dummyjson.com/products'));  
  
    if (response.statusCode != 200) {  
      throw Exception('Falha ao carregar produtos. Status code:  
${response.statusCode}');  
    }  
  
    final data = json.decode(response.body);  
    final List<dynamic> products = data['products'];  
  
    if (products.isEmpty) {  
      throw Exception('Nenhum produto encontrado na API');  
    }  
  
    final firstTenProducts = products.take(10).toList();  
  }  
}
```

```
await _saveProductsToDatabase(firstTenProducts);
```

```
setState(() {  
  _message = 'Produtos descarregados com sucesso!';  
});  
} catch (e) {  
  setState(() {  
    _message = 'Erro: $e';  
  });  
} finally {  
  setState(() {  
    _isLoading = false;  
  }); } }
```

```
Future<void> _saveProductsToDatabase(List<dynamic> products) async {  
    // Obter uma referência ao banco de dados  
    final database = await DatabaseHelper.instance.database;  
  
    // Limpar a tabela antes de inserir novos produtos  
    await database.delete('products');  
  
    // Inserir os produtos na base de dados  
    for (var product in products) {  
        await database.insert(  
            'products',  
            {  
                'id': product['id'],  
                'title': product['title'],  
                'description': product['description'],  
                'price': product['price'],  
            },  
            conflictAlgorithm: ConflictAlgorithm.replace,  
        );  
    }  
}
```

Listar_produtos.dart:

```
import 'package:flutter/material.dart';
```

```
import 'database_helper.dart';
```

```
import 'app_drawer.dart';
```

```
class ListProductsPage extends StatefulWidget {
```

```
  const ListProductsPage({Key? key}) : super(key: key);
```

```
  @override
```

```
  _ListProductsPageState createState() => _ListProductsPageState();
```

```
}
```

```
class _ListProductsPageState extends State<ListProductsPage> {
```

```
  List<Map<String, dynamic>> _products = [];
```

```
  bool _isLoading = true;
```

```
  @override
```

```
  void initState() {
```

```
    super.initState();
```

```
    _loadProducts();
```

```
}
```

```
Future<void> _loadProducts() async {
```

```
  setState(() {
```

```
    _isLoading = true;
```

```
  });
```



```
try {  
  
  final database = await DatabaseHelper.instance.database;  
  
  final products = await database.query('products');  
  
  
  setState(() {  
  
    _products = products;  
  
  });  
} catch (e) {  
  
  if (mounted) {  
  
    ScaffoldMessenger.of(context).showSnackBar(  
  
      SnackBar(content: Text('Erro ao carregar produtos: $e')),  
  
    );  
  
  }  
} finally {  
  
  setState(() {  
  
    _isLoading = false;  
  
  });  
  
}  
}
```

```
@override
```

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

```
  return Scaffold(
```

```
    appBar: AppBar(
```

```
      title: const Text('Lista de Produtos'),
```

```
    ),
```

```
    drawer: AppDrawer(),
```

```
    body: _isLoading
```

```
      ? const Center(child: CircularProgressIndicator())
```

```
      : _products.isEmpty
```

```
        ? const Center(
```

```
          child: Text(
```

```
            'Nenhum produto encontrado.\nPor favor, descarregue os produtos  
primeiro.',
```

```
            textAlign: TextAlign.center,
```

```
          ),
```

```
        )
```

```
      : Padding(
```

```
        padding: const EdgeInsets.all(16.0),
```

```
        child: Column(
```

```
          crossAxisAlignment: CrossAxisAlignment.start,
```

```
          children: [
```

```
            const Text(
```

```
              'Produtos na base de dados:',
```

```
              style: TextStyle(
```

```
                fontSize: 18,
```

```
                fontWeight: FontWeight.bold,
```

```
            ),
```

```
    ),  
    const SizedBox(height: 16),  
    Expanded(  
      child: ListView.builder(  
        itemCount: _products.length,  
        itemBuilder: (context, index) {  
          final product = _products[index];  
          return Card(  
            margin: const EdgeInsets.only(bottom: 8),  
            child: ListTile(  
              title: Text(product['title']),  
              subtitle: Text('Preço: ${product['price']}€'),  
            ),  
          );  
        },  
      ),  
    ),  
  ],  
),  
);  
}  
}
```

App_drawer.dart:

```
import 'package:flutter/material.dart';

class AppDrawer extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return Drawer(

      child: ListView(

        padding: EdgeInsets.zero,

        children: [

          const DrawerHeader(

            decoration: BoxDecoration(

              color: Colors.blue,

            ),

            child: Text(

              'Menu de Navegação',

              style: TextStyle(

                color: Colors.white,

                fontSize: 24,

              ),

            ),

          ),

          ListTile(

            leading: const Icon(Icons.home),

            title: const Text('Página Principal'),

            onTap: () {

              Navigator.pop(context); // Fecha o drawer

              Navigator.pushReplacementNamed(context, '/');
```

```
    },  
  ),  
  ListTile(  
    leading: const Icon(Icons.download),  
    title: const Text('Descarregar Produtos'),  
    onTap: () {  
      Navigator.pop(context); // Fecha o drawer  
      Navigator.pushReplacementNamed(context, '/download');  
    },  
  ),  
  ListTile(  
    leading: const Icon(Icons.list),  
    title: const Text('Listar Produtos'),  
    onTap: () {  
      Navigator.pop(context); // Fecha o drawer  
      Navigator.pushReplacementNamed(context, '/list');  
    },  
  ),  
],  
),  
);  
}  
}
```

Database_helper.dart:

```
import 'package:sqflite/sqflite.dart';

import 'package:path/path.dart';

// Classe para gerenciar a base de dados

class DatabaseHelper {

  static final DatabaseHelper instance = DatabaseHelper._init();

  static Database? _database;

  DatabaseHelper._init();

  Future<Database> get database async {

    if (_database != null) return _database!;

    _database = await _initDB('products.db');

    return _database!;

  }

  Future<Database> _initDB(String filePath) async {

    final dbPath = await getDatabasesPath();

    final path = join(dbPath, filePath);

    return await openDatabase(

      path,

      version: 1,

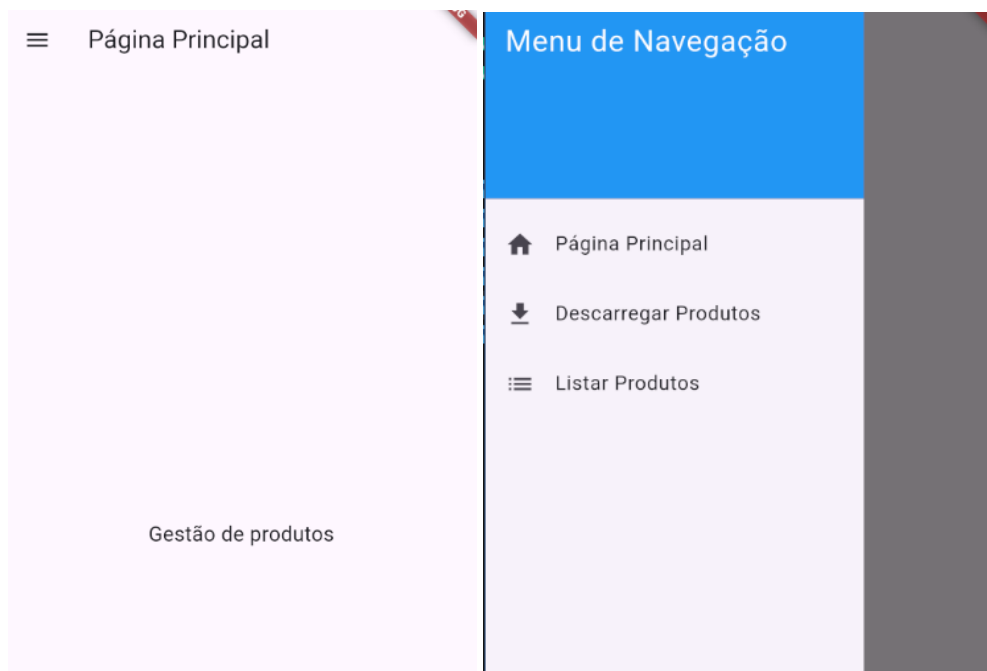
      onCreate: _createDB,

    );

  }

}
```

```
Future<void> _createDB(Database db, int version) async {  
  await db.execute("  
    CREATE TABLE products(  
      id INTEGER PRIMARY KEY,  
      title TEXT NOT NULL,  
      description TEXT,  
      price REAL  
    )  
  ");  
}  
}
```





Descarregar Produtos

Descarregar Produtos

Produtos descarregados com sucesso!



Lista de Produtos

Produtos na base de dados:

Essence Mascara Lash Princess
Preço: 9.99€

Eyeshadow Palette with Mirror
Preço: 19.99€

Powder Canister
Preço: 14.99€

Red Lipstick
Preço: 12.99€

Red Nail Polish
Preço: 8.99€

Calvin Klein CK One
Preço: 49.99€

Chanel Coco Noir Eau De
Preço: 129.99€