Tarefa 12 – PV26465

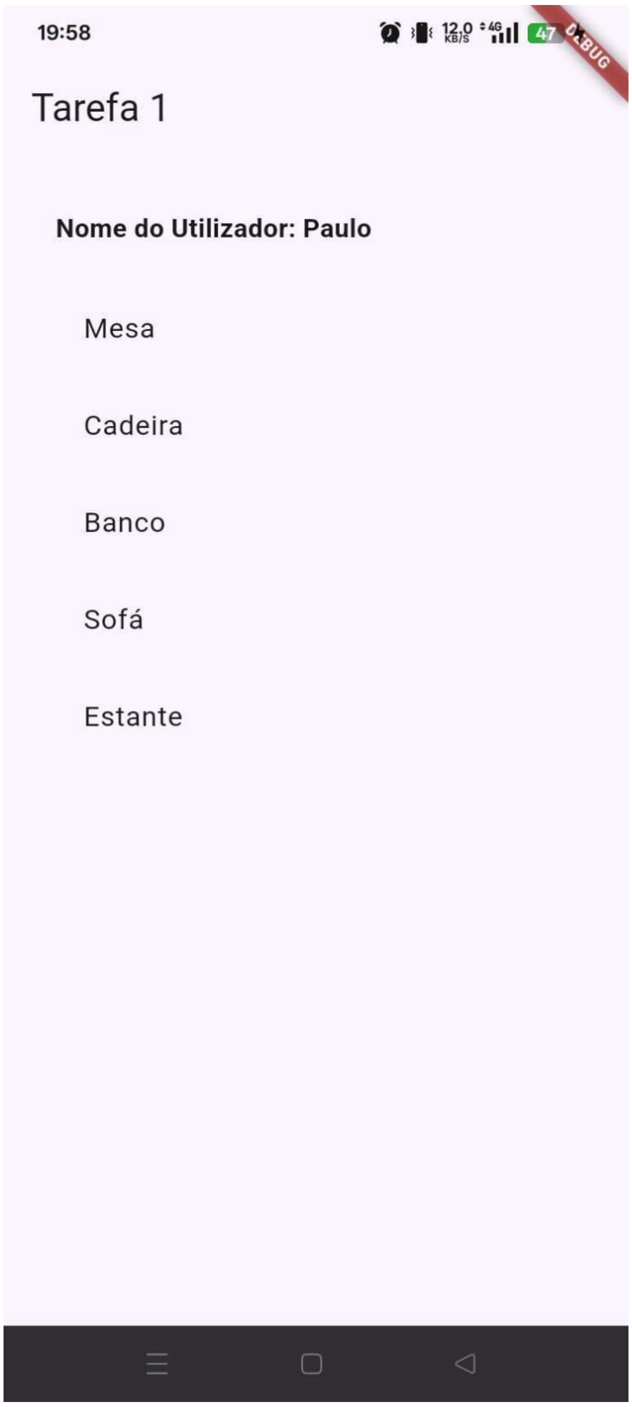
TAREFA 1

Main.dart

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

class Utilizador with ChangeNotifier {
 String nome = '';

 List<Map<String, String>> usuarios = [
  {'username': 'Paulo', 'password': '1234'},
  {'username': 'Maria', 'password': '5678'},
  {'username': 'João', 'password': '9012'},
 ];

 Future<void> valida() async {
  for (var usuario in usuarios)
  {
   await Sharpref().registaUt(usuario['username']!, usuario['password']!);
  }

  nome = await Sharpref().lerUt(
   //Se quiser testar com outro utilizador, basta alterar aqui o index.
   usuarios[0]['username']!,
   usuarios[0]['password']!,
  );

  notifyListeners();
  }
 Future<void> criar() async {
  await valida();
  notifyListeners();
 }
}

class Produto with ChangeNotifier {
 List<String> produtos = [];
 Future<void> lista() async {
  produtos = ['Mesa', 'Cadeira', 'Banco', 'Sofá',
  'Estante']; notifyListeners();
 }
}

void main() {
 runApp(
 MultiProvider(
   providers: [
    ChangeNotifierProvider(create: (_) =>
    Utilizador()), ChangeNotifierProvider(create: (_)
    => Produto()),
   ],
   child: MyApp(),
  ),
 );
}

class MyApp extends StatelessWidget
 { const MyApp({super.key});

 @override
 Widget build(BuildContext context) {
  return MaterialApp(home: Ecrpr());
 }
}

class Ecrpr extends StatefulWidget {
 const Ecrpr({super.key});

 @override
 _Ecrpr createState() => _Ecrpr();
}

class _Ecrpr extends State<Ecrpr>
 { @override
 void initState() {
  super.initState();
  Future.microtask(() {
   Provider.of<Utilizador>(context, listen:
   false).criar(); Provider.of<Produto>(context,
   listen: false).lista();
  });
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(title: Text('Tarefa
   1')), body: Container(
    padding:
    EdgeInsets.all(30), child:
    Column(
     children: [
      Consumer<Utilizador>
      (
       builder: (context, utili, child) {
        return Text(
         // ignore: unnecessary_string_interpolations
         'Nome do Utilizador: ${utili.nome}',
         style: TextStyle(fontSize: 15, fontWeight: FontWeight.bold),
        );
       },
      ),
      SizedBox(height:
      20), ListView(
      shrinkWrap: true,
      children: [
        Consumer<Produto>(
         builder: (context, produtos,
          child) { return Column(
           children:
             produtos.produtos
               .map((produto) => ListTile(title: Text(produto)))
               .toList(),
          );
         },
        ),
       ],
      ),
     ],
    ),
   ),
  );
 }
}
```

Shared.dart

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


class Sharpref {
 Future<void> registaUt(lo, pa) async

  { final SharedPreferences prefs =

  await

SharedPreferences.getInstance();

  await prefs.setString('log_$lo', lo);

  await prefs.setString('pass_$lo',

  pa);

 }


 Future<String> lerUt(lo, pa) async {
  final SharedPreferences prefs =
  await
SharedPreferences.getInstance();

  String? log =

  prefs.getString('log_$lo');

  String? pas =

  prefs.getString('pass_$lo'); if (log == lo

  CC pas == pa) {

   return log!;

  } else {

   return ('Utilizador nao autenticado');

  }

 }

}
```



Tarefa 1

Nome do Utilizador: Paulo

Mesa

Cadeira

Banco

Sofá

Estante

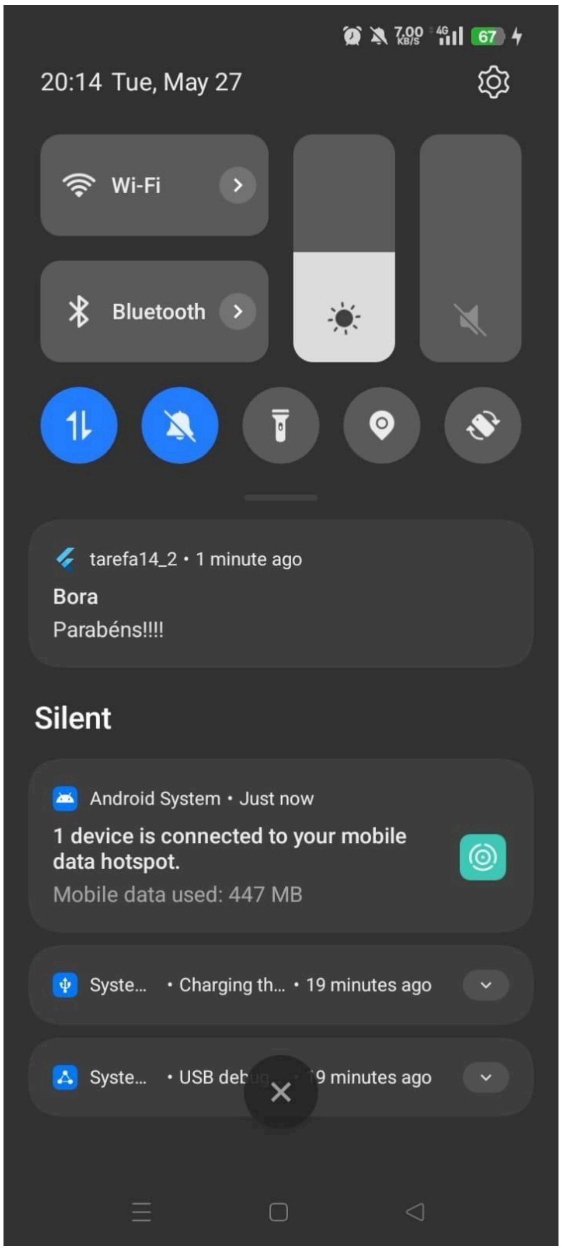
TAREFA 2

Main.dart

```
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase.dart';

void main() async {
 WidgetsFlutterBinding.ensureInitialized(
 ); await Firebase.initializeApp();
 await
 FirebaseAPI().initNotifications();
 runApp(const MyApp());
}

class MyApp extends StatelessWidget
 { const MyApp({super.key});

 // This widget is the root of your
 application. @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter Demo',
   theme: ThemeData(
    // This is the theme of your application.
    //
    // TRY THIS: Try running your application with "flutter run". You'll see
    // the application has a purple toolbar. Then, without quitting the app,
    // try changing the seedColor in the colorScheme below to Colors.green
    // and then invoke "hot reload" (save your changes or press the "hot
    // reload" button in a Flutter-supported IDE, or press "r" if you used
    // the command line to start the app).
```

```
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    backgroundColor:
Theme.of(context).colorScheme.inversePrima
ry,
    title: Text(widget.title),
   ),
   body: Center(
    child:
    Column(
     mainAxisAlignment:
     MainAxisAlignment.center, children:
     <Widget>[
      ElevatedButton(
       onPressed: () async
       {
        final token = await
FirebaseMessaging.instance.getToken(
);
        print('Token manual: $token');
       },
       child: Text('Obter Token'),
      ),
     ],
    ),
   ),
  );
 }
}
```


20:14 Tue, May 27

tarefa14_2 • 1 minute ago
Bora
Parabéns!!!!

Silent

Android System • Just now
1 device is connected to your mobile data hotspot.
Mobile data used: 447 MB

Syste... • Charging th... • 19 minutes ago

Syste... • USB deb... 9 minutes ago

TAREFA 3

Main.dart

```
import 'package:flutter/material.dart';
import 'package:open_route_service/open_route_service.dart';
import 'package:geolocator/geolocator.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';

void main() {
 runApp(const MyApp());
}

class MyApp extends StatelessWidget
 { const MyApp({super.key});
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter Demo',
   theme: ThemeData(
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
   ),
   home: const MyHomePage(title: 'Flutter Demo Home Page'),
  );
 }
}

class MyHomePage extends StatefulWidget {
 const MyHomePage({super.key, required
 this.title}); final String title;
 @override
 State<MyHomePage> createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
 final Localizacao local = Localizacao();
 final Rota rota = Rota();
 Set<Polyline> directs = {};

 double lat = 40.657845;
 double long = -7.913486;
 double latc = 40.661; // Latitude do ponto central
 double longc = -7.912; // Longitude do ponto
 central

 Future<void> _carregarRota() async {
  try {
   Polyline conj_pontos = await rota.daRota(
    40.6574, // ESTGV
    -7.9121,
    40.657845, // Rossio
    -7.913486,
   );
   setState(() {
    directs.add(conj_pontos);
   });
  } catch (e) {
   print('Erro ao carregar rota: $e');
  }
 }

 @override
 void initState() {
  super.initState();
  _carregarRota();
```

```
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   backgroundColor: Theme.of(context).colorScheme.inversePrimary,
   title: Text(widget.title),
  ),
  body:
   SizedBox(
   width: 300,
   height: 300,
   child:
   Column(
    mainAxisAlignment:
    MainAxisAlignment.center, children:
    <Widget>[
     GoogleMap(
      initialCameraPosition: CameraPosition(
       target: LatLng(latc, longc),
       zoom: 15.0,
      ),
      markers:
       {
       Marker(
        markerId: MarkerId('Viseu'),
        position: LatLng(lat, long),
       ),
      },
      polylines: directs,
     ),
    ],
   ),
```

),

```
  );
 }
}

class Rota {
 final OpenRouteService client = OpenRouteService(
  apiKey: '5b3ce3597851110001cf62485f612f5cfefb40f4bc6994335d0d91e9',
 );
 Future<Polyline> daRota(iniclat, iniclon, fimlat, fimlong)
  async { final List<ORSCoordinate> routeCoordinates = await
  client
    .directionsRouteCoordsGet(
     startCoordinate: ORSCoordinate(latitude: iniclat, longitude:
     iniclon), endCoordinate: ORSCoordinate(latitude: fimlat,
     longitude: fimlong),
    );
  final List<LatLng> routePoints =
    routeCoordinates
      .map(
       (coordinate) => LatLng(coordinate.latitude, coordinate.longitude),
      )
      .toList();
  final Polyline routePolyline =
   Polyline( polylineId:
   PolylineId('Percurso'), visible: true,
   points: routePoints,
   color: Colors.red,
   width: 4,
  );
  return (routePolyline);
 }
}
```

```
class Localizacao {
 Future<Position> determinaposicao() async {
  bool serviceEnabled;
  LocationPermission permission;
  serviceEnabled = await
  Geolocator.isLocationServiceEnabled(); if (!serviceEnabled) {
   return Future.error('Serviços indisponíveis.');
  }
  permission = await
  Geolocator.checkPermission(); if (permission ==
  LocationPermission.denied) {
   permission = await
   Geolocator.requestPermission(); if (permission ==
   LocationPermission.denied) {
    return Future.error('Não tem permissões de localização');
   }
  }
  if (permission == LocationPermission.deniedForever) {
   return Future.error('Permissões definitivamente
   negadas.');
  }
  var ret = await Geolocator.getCurrentPosition();
  return ret;
 }
}
```

TAREFA 4

Main.dart

```
// ignore_for_file: unnecessary_new

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

void main() {
 runApp(const MyApp());
}

class MyApp extends StatelessWidget
 { const MyApp({super.key});

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter Demo',
   theme: ThemeData(
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
   ),
   home: const MyHomePage(title: 'Flutter Demo Home Page'),
  );
 }
}

class MyHomePage extends StatefulWidget {
 const MyHomePage({super.key, required this.title});
 final String title;
 @override
 State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
 @override
 Widget build(BuildContext context) {
  return new Scaffold(
   appBar: new AppBar(title: new Text("Offline
   Demo")), body: OfflineBuilder(
    connectivityBuilder: (
     BuildContext context,
     List<ConnectivityResult>
     connectivity, Widget child,
    ) {
     final bool connected =
       !connectivity.contains(ConnectivityResult.non
     e); return new Stack(
      fit:
      StackFit.expand,
      children: [
      Positioned(
        height: 24.0,
        left: 0.0,
        right: 0.0,
        child: Container(
         color: connected ? Color(0xFF00EE44) : Color(0xFFEE4400),
         child: Center(
          child: Text(":{connected ? 'ONLINE' : 'OFFLINE'}"),
         ),
        ),
       ),
      ],
     );
    },
    child: Column(
     mainAxisAlignment: MainAxisAlignment.center,
     children: <Widget>[
      new Text('There are no bottons to push
      :)'), new Text('Just turn off your
      internet.'),
     ],
    ),
   ),
  );
 }
}
```