Explicação do Projeto Mediador de Seguros

O projeto **Mediador de Seguros** é uma aplicação **Flutter** cujo objetivo é permitir a interação entre clientes e um mediador de seguros, com funcionalidades específicas para cada tipo de utilizador. Seguem-se abaixo as principais informações sobre como o projeto está organizado, as funcionalidades e o que é necessário para o executar.

---

1. Estrutura Geral do Projeto

1. **lib/screens**
   - **cliente**: Telas específicas para o utilizador com papel "cliente" (Pedir Orçamento, Ver Orçamentos, Apólices ativas, etc.).
   - **mediador**: Telas específicas para o utilizador com papel "mediador" (Lista de Seguradoras, Responder Orçamento, Apólices ativas, etc.).
   - **settings**: Telas de configurações (alterar tema, informações pessoais, etc.).
   - **login**: Tela de login.
   - **home**: Tela principal que redireciona para funcionalidades dependendo do papel do utilizador.
   - **splash**: Tela de Splash (inicial) para verificar se há utilizador autenticado e encaminhar para Home ou Login.
2. **lib/models**
   - Contém classes de modelo (por exemplo, `Seguradora`, `PigeonUserDetails`) que representam dados manipulados no projeto.
3. **lib/services**
   - Responsáveis pela comunicação com APIs (OpenWeather, Car API) e com o Firebase (AuthService).
4. **lib/widgets**
   - Widgets reutilizáveis, como o `MainDrawer`, que apresenta o menu lateral de configurações.
5. **assets**
   - Pasta onde ficam as imagens utilizadas pelo projeto (ícone da aplicação, logos, etc.).
6. **pubspec.yaml**
   - Ficheiro de configuração onde se definem dependências (Firebase, http, shared_preferences, etc.), assets e outras configurações do projeto.
7. **firebase_options.dart**
   - Ficheiro gerado pela FlutterFire CLI, com as configurações e chaves necessárias para ligar ao Firebase em cada plataforma (Android/iOS).
8. **main.dart**
   - Ponto de entrada da aplicação. Inicializa o Firebase, carrega as preferências de tema e configura as rotas iniciais (Splash, Login, Home, etc.).

---

2. Funcionalidades Principais

1. **Login e Registo**
   - Tela de login (`LoginScreen`) para o utilizador se autenticar (email/password).
   - Registo para dois papéis (Cliente e Mediador) através de `RegisterClientScreen` e `RegisterMediadorScreen`.
2. **Gestão de Temas**
   - O utilizador pode alternar entre tema claro e escuro (guardado em `SharedPreferences`).
3. **Splash Screen**
   - Verifica se existe um utilizador autenticado e redireciona para `HomeScreen` ou `LoginScreen`.
4. **Home Screen**
   - Mostra opções diferentes dependendo do papel do utilizador:
     - **Mediador**: Acesso a listas de seguradoras, pedidos de orçamento e apólices ativas.
     - **Cliente**: Pedir orçamentos, ver orçamentos respondidos e gerir apólices ativas.
5. **Funcionalidades de Cliente**
   - **Pedir Orçamento**: Preenche dados para Habitação, Vida, Automóvel ou Trabalho. Cria documento no Firestore com status "pendente".
   - **Ver Orçamentos**: Mostra orçamentos com status "respondido".
   - **Apólices Ativas**: Consulta as apólices (pode cancelar ou antecipar término).
6. **Funcionalidades de Mediador**
   - **Lista de Seguradoras**: Criar, editar ou apagar seguradoras no Firestore.
   - **Pedidos de Orçamento Pendentes**: Ver todos os pedidos "pendentes" e responder com valores para cada seguradora.
   - **Apólices Ativas**: Ver informações de todas as apólices ativas.
7. **Integrações Externas**
   - **Car API (NHTSA)**: Obter lista de marcas/modelos para o seguro automóvel.
   - **OpenWeather**: Obter condições climáticas (para seguros de habitação).

- **Firebase**:
  - **Firebase Auth** e **Firestore** para autenticação e base de dados de orçamentos, apólices, seguradoras e dados de utilizador.

8. **Configurações (Settings)**
   - Permite mudar de tema (claro/escuro).
   - Acesso a tela de Informações Pessoais, etc.

---

3. O que é Necessário para Rodar o Projeto

1. **Flutter e Dart**
   - Ter o Flutter instalado (>=3.6.0) e configurado.
   - Verificar se o SDK Dart está de acordo com o `pubspec.yaml`.
2. **Firebase**
   - Ter uma conta no Firebase e o projeto configurado (já refletido em `firebase_options.dart`).
   - Configurar `google-services.json` (Android) e `GoogleService-Info.plist` (iOS), se aplicável.
3. **Dependências**
   - Executar `flutter pub get` para instalar:
     - **firebase_core**, **firebase_auth**, **cloud_firestore**
     - **shared_preferences**
     - **http**
     - **provider** (opcional)
     - **flutter_launcher_icons** (opcional para ícones)
4. **Configurações de Ambiente**
   - Para Android: ter Android Studio ou SDK instalado.
   - Para iOS: precisar do Xcode e ambiente Mac.
5. **Execução**
   - Garantir que há um emulador ou dispositivo físico disponível.
   - No terminal, executar:

     ```
     flutter pub get
     flutter run
     ```

   - Isto instala as dependências, compila o projeto e executa no dispositivo selecionado.

---

4. Conclusão

O **Mediador de Seguros** combina:

- Capacidades de **Cliente**: solicitar orçamentos e gerir apólices.
- Capacidades de **Mediador**: responder aos orçamentos, gerir seguradoras.

Usa **Firebase** (Auth/Firestore), integrações externas (Car API e OpenWeather) e suporte a tema escuro/claro em **Flutter**.

Para executar:

1. Ter Flutter configurado.
2. Correr `flutter pub get`.
3. Correr `flutter run`.

As chaves do Firebase estão em `firebase_options.dart`. Ajuste conforme o seu ambiente, se necessário.

Trabalho realizado por:

- **Miguel Magalhães**
- Nº: 2021103166
- Unidade Curricular de Computação Móvel
- Licenciatura em Engenharia Informática
- ISPGAYA