

# Guião de Apresentação do Projeto "Mediador de Seguros"

---

Este guião tem como objetivo mostrar, de forma estruturada e didática, como apresentar todas as funcionalidades e componentes do projeto "Mediador de Seguros", de acordo com as diretrizes solicitadas pelo professor.

---

## 1. Introdução Geral

---

1. **Objetivo**
    - Apresentar uma **aplicação móvel em Flutter** que facilite a gestão de seguros pelo mediador (CRUD de seguradoras, orçamentos, apólices) e a subscrição/gestão de apólice pelo cliente (pedido de orçamentos, visualização e escolha de seguradora, gestão de apólices).
  2. **Estrutura**
    - Mostragem do fluxo de ecrãs (Splash, Login, Home, Modo Mediador, Modo Cliente) e integrações adicionais (Firebase, APIs externas, lógica de descontos, etc.).
- 

## 2. Roteiro de Apresentação

---

### 2.1. Splash Screen

1. **Mostrar a Splash Screen**
    - Explicar que, ao abrir a app, a Splash Screen verifica no Firebase Auth se há um utilizador autenticado.
    - Se sim, redireciona para a HomeScreen; se não, para a LoginScreen.
- 

### 2.2. Login & Autenticação

1. **Apresentar Login Screen**
    - **Login** via email/password (Firebase Auth).
    - Explicar que, caso o utilizador não tenha conta, pode registar-se como **Cliente** ou **Mediador**:
      - **Register Client**
      - **Register Mediador**
    - Referir que o **código** utiliza FirebaseAuth para criar e gerir os utilizadores.
    - **Sessão**: o utilizador, após login, permanece autenticado até efetuar logout.
- 

### 2.3. Home Screen

1. **Papel (Role) do Utilizador**
    - Dependendo do papel (Cliente ou Mediador), a HomeScreen exibe menus e opções diferentes:
      - Se mediador: ícone, texto de "Bem-vindo Mediador!".
        - Acesso a gerir seguradoras, pedidos de orçamento, apólices ativas.
      - Se cliente: ícone, texto de "Bem-vindo Cliente!".
        - Acesso a pedir orçamentos, ver orçamentos respondidos, gerir apólices ativas.
  2. **Demonstração das Rotas**
    - Explicar a navegação interna (através dos botões ou cards mostrados na HomeScreen) para cada funcionalidade.
- 

### 2.4. Funcionalidades do Modo Mediador

1. **Adicionar, Editar e Remover Seguradoras**
    - Mostrar o ecrã "Lista de Seguradoras" (CRUD):
      - **Adicionar**: apresenta um formulário para inserir **nome** e **tipos** de seguro.
      - **Editar**: formulário para alterar os mesmos campos.
      - **Remover**: exibe confirmação e apaga do Firestore.
  2. **Visualizar Pedidos de Orçamento Pendentes e Responder**
    - Mostrar a tela de "Pedidos de Orçamento Pendentes":
      - Lista de pedidos com status "pendente".
      - Responder colocando um valor para cada seguradora.
      - Aplicação de **descontos** conforme a quantidade de apólices ativas do cliente (5% ou 10%).
      - Ao "responder", o status do orçamento passa a "respondido" e o cliente vê os valores.
  3. **Visualizar Apólices Ativas**
    - Mostrar a tela onde o mediador vê todas as apólices com status "ativa", podendo consultar dados do cliente, da seguradora, e datas de início/fim.
-

## 2.5. Funcionalidades do Modo Cliente

### 1. Escolher Tipo de Seguro e Pedir Orçamento

- Explicar ecrã para **"Pedir Orçamento"**, mostrando diferentes campos de entrada consoante o tipo de seguro (Vida, Automóvel, Trabalho, Habitação).
- **Integração com API** (ex. Car API) para preencher marca, modelo, etc. no seguro automóvel.
- Ao enviar o formulário, cria um documento no Firestore com status "pendente".

### 2. Visualizar Orçamentos e Escolher Seguradora

- Mostrar a lista de orçamentos com status "respondido" (o mediador já atribuiu valores).
- O cliente pode:
  - Escolher uma seguradora (e isto cria uma apólice com status "ativa").
  - Ou recusar o orçamento (status "recusado").

### 3. Visualizar Apólices Ativas

- Mostrar a tela das apólices ativas do cliente, com possibilidade de:
  - **Cancelar** a apólice imediatamente (status "cancelada").
  - **Antecipar a data de término** (status "terminada").

---

## 3. Principais Pontos de Avaliação (Checklist)

### 3.1. Routings & Screens (4 valores)

- **Screens:**
  - Splash, Login, Home (mediador + cliente), Pedir Orçamento (cliente), Responder Orçamentos (mediador), Ver Orçamentos (cliente), Apólices Ativas (mediador + cliente).
  - Demonstração de como cada rota está configurada no `main.dart`.
- **Menu:**
  - Apresentar o **MainDrawer** (caso aplicável) ou outro mecanismo de navegação.

---

### 3.2. Forms & Inputs + Firebase (7 valores)

- **CRUD para seguradoras (mediador)**
  - Inclusão, edição e remoção na Firestore.
- **Listar pedidos de orçamentos e apólices ativas (mediador)**
  - Pedidos pendentes -> Responder.
  - Apólices -> Visualização.
- **Pedir orçamentos, visualizar, escolher seguradora ou recusar (cliente)**
  - Mostrar implementação no Firestore e a lógica de input de dados.
- **Gestão de apólices ativas (cliente)**
  - Cancelar, antecipar fim.

---

### 3.3. Login e Autenticação (3 valores)

- **Registo de utilizadores modo Cliente**
- **Registo de utilizadores modo Mediador**
- **Login e autenticação** (FirebaseAuth)
- **Sessão** (ficar logado até logout).

---

### 3.4. Custom Widgets (1.5 valores)

- Uso de **MainDrawer**, ListTile reutilizáveis, Cards, etc.

---

### 3.5. API (3 valores)

- **Configuração e integração** da Car API (ex.: NHTSA) e OpenWeather.
  - **Chamadas assíncronas com http**.
  - **Listar dados da API** na UI (marcas, modelos de carro).
  - **Validação e tratamento** dos dados recebidos.
-

### 3.6. Release da App (1.5 valores)

- Apresentar o **ficheiro APK ou IPA** gerado.
  - **Ícone da aplicação** (flutter\_launcher\_icons).
  - **Splash screen** personalizada (imagem + background).
- 

### 3.7. Organização, Esforço e Investigação (X)

- Explicar brevemente como o código está organizado, comentado e modularizado.

#### 3.7.1. Código (1.5 valores)

- Estrutura e organização.
- Comentários.

#### 3.7.2. Estética e Intuitividade (0.5 valores)

- Mostrar que a interface é clara, intuitiva e fácil de navegar.

#### 3.7.3. Valorização Adicional (2 valores)

- Uso de mais que uma API (Car API + OpenWeather).
  - Firebase Authentication.
  - Lógica de descontos.
  - Outras otimizações (ex.: modo escuro, splash screen customizada, etc.).
- 

## 4. Conclusão e Fecho

---

#### 1. Resumo

- Mostrar como o projeto cumpre todos os requisitos: routings, formulários, Firebase, APIs, etc.
- Destacar a lógica de descontos e a modularização do código (custom widgets).

#### 2. Demonstração Final

- Exibir brevemente um fluxo completo: Login (Cliente) -> Pedir Orçamento -> Logout -> Login (Mediador) -> Responder Orçamento -> Logout -> Login (Cliente) -> Escolher Seguradora -> Ver Apólices Ativas.

#### 3. Q&A

- Abrir para eventuais questões do professor sobre a implementação, organização do projeto, decisões de design, etc.
- 

#### Trabalho realizado por:

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