# *DESENVOLVIMENTO QUOD APP*

Simulação de Fluxos de Autenticação e Antifraude

*Mariana Spinola Federico*

*Challenge FIAP | RM553476*

# Table of Contents

# 1. Telas do aplicativo e seus possíveis cenários

## Telas do Aplicativo:

### Tela de Boas-Vindas:

**Cenário 1:** O usuário abre o aplicativo e é apresentado a uma tela inicial com informações breves sobre o que o aplicativo faz e um botão de "Começar".

**Cenário 2:** Se o usuário já completou um fluxo, ele pode ser redirecionado para a tela de acompanhamento.

### Tela de Biometria Facial:

**Cenário 1:** O usuário é instruído a posicionar o rosto para a captura da imagem.

**Cenário 2:** Após a captura, o aplicativo simula a validação, exibindo se a biometria foi bem-sucedida ou não.

**Cenário 3:** Se a captura falhar (por exemplo, o rosto não está bem posicionado), uma mensagem de erro será exibida e o usuário pode tentar novamente.

### Tela de Biometria Digital:

**Cenário 1:** O usuário deve colocar o dedo no sensor para capturar a impressão digital.

**Cenário 2:** Após a captura, o sistema valida a digital e retorna uma mensagem de sucesso ou erro, dependendo do resultado.

### Tela de Análise de Documento (Documentoscopia):

**Cenário 1:** O usuário tira uma foto do documento de identidade e da sua face.

**Cenário 2:** O aplicativo simula a validação do documento e da foto, exibindo se são autênticos ou se há alguma inconsistência.

### Tela de SIM SWAP:

**Cenário 1:** O usuário insere seu número de telefone para verificar se houve uma troca recente de chip.

**Cenário 2:** O aplicativo simula a chamada a um endpoint de operadora e exibe se a troca de chip foi detectada ou não.

### Tela de Autenticação Cadastral:

**Cenário 1:** O usuário preenche um formulário com informações como nome, CPF, endereço e telefone.

**Cenário 2:** O aplicativo valida os dados inseridos e exibe mensagens de erro ou sucesso, dependendo da verificação.

### Tela de Score Antifraude:

**Cenário 1:** O usuário insere o CPF e o aplicativo exibe o score de fraude (1-1000).

**Cenário 2:** O score é mostrado juntamente com uma explicação sobre o risco de fraude com base no valor exibido.

# 2. Detalhes das bibliotecas utilizadas para desenvolvimento do aplicativo (Descritivo)

## Bibliotecas Utilizadas:

### SwiftUI:

- **Descrição:** SwiftUI é um framework da Apple utilizado para construir interfaces de usuário de maneira declarativa. Ele permite criar aplicativos de forma intuitiva e rápida, utilizando uma abordagem reativa para gerenciar o estado da interface.
- **Utilização no aplicativo:** SwiftUI é utilizado em todo o design de interfaces do aplicativo, desde telas de captura biométrica até formulários de autenticação e verificação.

### SwiftUI Previews (para testar o layout e a interface):

- **Descrição:** A ferramenta SwiftUI Previews permite visualizar e testar a interface de forma interativa enquanto o código é escrito.
- **Utilização no aplicativo:** Usada para testar diferentes estados das telas, como sucesso ou erro nas validações, de maneira rápida e eficiente.

## Bibliotecas a serem implementadas:

### PhoneNumberKit:

- **Descrição:** PhoneNumberKit é uma biblioteca de código aberto que facilita a validação, formatação e análise de números de telefone em diversos formatos internacionais.
- **Utilização no aplicativo:** Utilizada para validar e formatar números de telefone, especialmente no fluxo de SIM SWAP e autenticação cadastral. Ela permite garantir que os números de telefone estejam no formato correto e atendam às exigências do sistema.

### Vision Framework (para biometria facial):

- **Descrição:** O framework Vision da Apple permite realizar tarefas como detecção de rostos e reconhecimento facial, o que pode ser utilizado para a validação biométrica facial.
- **Utilização no aplicativo:** Para realizar a captura e validação de biometria facial, como a detecção de rosto para validação.

### CoreImage (para análise de documentos):

- **Descrição:** CoreImage é uma biblioteca poderosa para processar imagens e realizar filtros, que pode ser usada para validar documentos por meio de imagens capturadas.
- **Utilização no aplicativo:** Usada para realizar a análise das fotos de documentos e validar se são autênticas ou não.

### HealthKit (para biometria digital):

- **Descrição:** Embora o HealthKit não seja utilizado diretamente para captura de impressões digitais, ele pode ser uma opção para registrar dados biométricos de maneira segura, caso você use sensores biométricos disponíveis no dispositivo.
- **Utilização no aplicativo:** Potencialmente utilizado se o aplicativo necessitar de uma integração com dispositivos de leitura biométrica.

### URLSession (para comunicação com API de SIM SWAP):

- **Descrição:** URLSession é um framework da Apple para realizar requisições HTTP. Ele pode ser usado para chamar endpoints e obter dados de APIs externas.
- **Utilização no aplicativo:** Usado para fazer uma requisição ao endpoint da operadora para validar a troca de chip (SIM SWAP).

### Combine (para gerenciamento de estado e fluxo de dados):

- **Descrição:** Combine é uma framework reativa da Apple que facilita o gerenciamento de eventos e mudanças de estado em tempo real.
- **Utilização no aplicativo:** Pode ser usado para gerenciar os estados de validação de cada fluxo de autenticação e notificar a interface sobre mudanças de dados de forma reativa.