**Documentação do App**

**1. Introdução**

**1.1 Objetivo**

Este documento descreve as diretrizes e a estrutura para o desenvolvimento do aplicativo em Flutter, incluindo suas funcionalidades, arquitetura e integração com o backend.

**1.2 Escopo**

A documentação se aplica a todos os desenvolvedores e stakeholders envolvidos no desenvolvimento do aplicativo, que utiliza Flutter como framework e Nest.js para o backend.

**2. Descrição do App**

**2.1 Tecnologias Utilizadas**

* **Frontend**: Flutter
* **Backend**: Nest.js
* **Autenticação**: Token JWT
* **Arquitetura**: Hexagonal (no backend)
* **Containerização**: Docker
* **Documentação da API**: Swagger

**2.2 Funcionalidades Principais**

* **Tela de Login e Registro**: O aplicativo contará com uma interface para login e registro de usuários.
* **Autenticação**: A autenticação será realizada utilizando token JWT, garantindo a segurança e integridade das sessões dos usuários.
* **Notificações Inteligentes**: A tela inicial do aplicativo exibirá notificações geradas por uma Inteligência Artificial em casos de tentativas de invasão.
* **Simulação de E-mails**:
  + A tela inicial incluirá simulações de recebimento de e-mails que conterão URLs.
  + As URLs serão enviadas para uma API para verificação de suspeita de phishing.
  + A API indicará se a URL é suspeita; se confirmada, o e-mail terá uma borda vermelha, indicando perigo. Caso contrário, a borda será verde, indicando que o site é seguro.

**3. CRUD de Usuário e E-mail**

**3.1 CRUD de Usuário**

* **Create**: Registro de novos usuários.
* **Read**: Recuperação de dados do usuário.
* **Update**: Atualização das informações do usuário.
* **Delete**: Remoção de usuários do sistema.

**3.2 CRUD de E-mail**

* **Create**: Simulação de novos e-mails recebidos.
* **Read**: Visualização dos e-mails simulados.
* **Update**: Modificação das informações dos e-mails.
* **Delete**: Exclusão de e-mails simulados.

**4. Conclusão**

Este documento serve como um guia para o desenvolvimento do aplicativo Flutter, assegurando que todas as funcionalidades e requisitos sejam atendidos de maneira organizada e eficiente. A implementação das tecnologias e funcionalidades descritas, em conjunto com o backend em Nest.js, visa garantir a segurança e a experiência do usuário no aplicativo.