**Documentação Técnica do Projeto Nutri Ai**

**1. Visão Geral do Projeto**

O NutriX Pro Ai é um aplicativo de apoio nutricional projetado para auxiliar os usuários a registrar suas dietas, receber orientações de nutrição e obter sugestões de refeições e receitas personalizadas com o auxílio de inteligência artificial (GPT API). O projeto, desenvolvido como Trabalho de Conclusão de Curso (TCC), visa democratizar o acesso a informações nutricionais, oferecendo uma ferramenta acessível e personalizada para pessoas com rotinas corridas ou que buscam uma alimentação mais saudável.

**2. Tecnologias Utilizadas**

O projeto utiliza uma stack de desenvolvimento moderna para a criação de aplicativos móveis multiplataforma:

| Tecnologia | | Função |
| --- | --- |
| **React Native** | | Framework para desenvolvimento mobile multiplataforma. |
| **Expo** | | Ambiente simplificado para desenvolvimento com React Native. |
| **TypeScript** | | Superset de JavaScript com tipagem estática para garantir a qualidade e escalabilidade do código. |
| **Node.js** | | Ambiente de execução para o Expo. |
| **Firebase** | | Utilizado para autenticação de usuários e como banco de dados (Firestore) para armazenar informações como perfis, refeições e receitas. |
| **OpenAI** | **(GPT API)** | | Integrado para fornecer um chat inteligente, sugerir receitas e estimar calorias. |
| **Git/GitHub** | | Para versionamento de código, com o repositório principal localizado em: https://github.com/lucca7r/NutriAi.git. |

Exportar para as Planilhas

**3. Estrutura de Pastas**

A estrutura do projeto é organizada de forma modular para facilitar a manutenção e o desenvolvimento:

NutriAi/

│

├── App.tsx # Ponto de entrada da aplicação

├── package.json # Dependências e informações do projeto

├── tsconfig.json # Configurações do TypeScript

├── .gitignore # Arquivos ignorados pelo Git

│

└── src/ # Código-fonte principal

├── screens/ # Telas da aplicação (HomeScreen, LoginScreen, etc.)

├── components/ # Componentes reutilizáveis (Logo, WeightChart, etc.)

├── services/ # Serviços de integração (Firebase, OpenAI)

├── context/ # Contextos globais (Autenticação, Tema)

├── navigation/ # Configuração da navegação (BottomTabs, AppNavigator)

├── styles/ # Estilos e temas da aplicação

└── @types/ # Definições de tipos TypeScript

**4. Funcionalidades Principais**

O aplicativo oferece um conjunto de funcionalidades para auxiliar o usuário em sua jornada de alimentação saudável:

* **Autenticação de Usuário:** Sistema de cadastro e login com e-mail e senha, utilizando o Firebase Authentication.
* **Formulário Inicial:** Coleta de informações do usuário para personalizar a experiência, incluindo dados pessoais, objetivos, estilo de vida e restrições alimentares.
* **Dashboard (Tela Inicial):** Apresenta um resumo do consumo diário de calorias e a evolução do peso do usuário em gráficos.
* **Registro de Refeições:** Permite ao usuário registrar suas refeições, com a opção de estimar as calorias com o auxílio da IA.
* **Chat com IA:** Um chat inteligente que utiliza a API do GPT para responder a dúvidas sobre nutrição, sugerir receitas e adaptações na dieta.
* **Gerenciador de Receitas Favoritas:** Os usuários podem salvar e organizar suas receitas favoritas em pastas personalizadas.
* **Dicas Personalizadas:** Geração de dicas de nutrição e bem-estar com base no perfil e objetivos do usuário.

**5. Configuração do Ambiente e Execução**

Para executar o projeto, siga os passos abaixo:

1. **Instalação de Dependências:** Certifique-se de ter o Node.js instalado. Em seguida, clone o repositório e instale as dependências necessárias executando o seguinte comando no terminal:

Bash

npm install

1. **Configuração de Variáveis de Ambiente:** O projeto utiliza um arquivo .env para armazenar as chaves de API do Firebase e da OpenAI. Certifique-se de criar este arquivo na raiz do projeto e adicionar as seguintes variáveis:
   * OPENAI\_API\_KEY
   * FIREBASE\_API\_KEY
   * FIREBASE\_AUTH\_DOMAIN
   * FIREBASE\_PROJECT\_ID
   * FIREBASE\_STORAGE\_BUCKET
   * FIREBASE\_MESSAGING\_SENDER\_ID
   * FIREBASE\_APP\_ID
2. **Execução do Projeto:** Para iniciar o aplicativo, utilize o Expo. Execute o comando abaixo no terminal e, em seguida, escaneie o QR code com o aplicativo Expo Go no seu celular ou utilize um emulador:

Bash

npx expo start

**6. Observações Adicionais**

* O projeto utiliza a biblioteca react-native-progress-steps para criar o formulário inicial de forma progressiva.
* A navegação do aplicativo é gerenciada pelo React Navigation, com uma estrutura de navegação em abas (BottomTabs) para as telas principais e uma navegação em pilha (Stack.Navigator) para as demais telas, como login, cadastro e o formulário.
* Os estilos são criados de forma dinâmica com base no tema (claro ou escuro) selecionado pelo usuário, utilizando a Context API do React.
* A integração com a API da OpenAI é realizada através do serviço openaiService.ts, que contém funções para estimar calorias, gerar receitas e interagir com o chat.