

Projeto em React Native - Desenvolvimento de Aplicativo

Introdução

O aplicativo foi desenvolvido com o objetivo principal de promover a conscientização sobre o peso corporal e a saúde do usuário, oferecendo uma plataforma sólida para o planejamento alimentar. Por meio dele, os usuários podem facilmente registrar seu peso e altura, acompanhando seu progresso ao longo do tempo. Construído com a tecnologia **React Native**, o aplicativo é totalmente compatível com dispositivos **iOS e Android**, proporcionando uma experiência de uso acessível em ambas as plataformas.

Público Alvo

O aplicativo foi idealizado para atender um público que compartilha o objetivo de melhorar a saúde e adotar hábitos alimentares equilibrados.

Descrição de cada funcionalidade

O aplicativo conta com funcionalidades dinâmicas e essenciais para o desenvolvimento de um aplicativo em React Native, como tela de cadastrado, tela do perfil do usuário, integrações APIS, etc.

Tela de Boas-Vindas

Este componente representa a tela principal do aplicativo, tem como principal objetivo desejar boas-vindas ao usuário.

A tela conta com dois botões, permite que o usuário escolha se cadastrar no aplicativo ou entre no aplicativo. Ao clicar uma das opções o usuário será levado para as respectivas tela de cadastro ou tela de login.

Tela de Cadastro do Usuário

A tela de cadastro é uma das principais telas do aplicativo, pois ela permite que novos usuários se registrem apenas inserindo as suas informações pessoais, como o nome, e-mail e senha. Após o usuário registrar os seus dados pessoais e enviar no formulário, os dados serão enviados para um servidor, criando um novo usuário no sistema.

Tela de Login do Usuário

Essa tela é a de login do sistema, ela permite que o usuário acesse a sua conta informando o e-mail e sua senha. Após o usuário informar os seus dados ele é imediatamente levado para a tela principal do aplicativo.

Se um dos dados pessoais estiverem incorretos, é exibido um alerta informando erro no login. Além disso, a tela oferece opção de navegar para a tela de **Cadastro** caso o usuário ainda não tenha uma conta, ou a opção de recuperar a senha caso tenha esquecido.

Tela de formulário para cálculo relacionados a Índice de Massa Corporal (IMC).

Essa tela é onde será realizado a lógica do cálculo do Índice de Massa Corporal (IMC). O usuário irá inserir o seu peso e sua altura, com os dados preenchidos imediatamente será calculado o Índice de Massa Corporal desse usuário. Assim que o cálculo for bem sucedido, uma mensagem de sucesso é exibida, e o usuário é redirecionado para a tela principal do aplicativo. Caso ocorra algum erro no envio ou nos dados, um alerta de erro é mostrado.

Tela Principal - Informações do Índice de Massa Corporal (IMC).

Essa tela desempenha um papel fundamental, pois é responsável por apresentar todas as informações relacionadas ao Índice de Massa Corporal (IMC) do usuário.

Na tela principal, há um gráfico circular em que cada cor representa uma classificação do estado nutricional do usuário, como peso normal, magreza extrema, sobrepeso ou obesidade. O gráfico exibe os resultados com base no cálculo do IMC do indivíduo.

Por exemplo, se uma pessoa tiver 1,70 m de altura e pesar 50 kg, o IMC será de 17,3. Nesse caso, ela será classificada como abaixo do peso.

Uma das principais funcionalidades deste sistema será a recomendação de dietas personalizadas com base no estado nutricional do usuário.

Por exemplo, se uma pessoa estiver classificada como abaixo do peso, o sistema sugerirá um plano alimentar diário que incluirá café da manhã, almoço, café da tarde e jantar. Para indivíduos abaixo do peso ideal, o cardápio será elaborado com foco em alimentos ricos em carboidratos e proteínas, atendendo à necessidade de ganho de peso de forma equilibrada e saudável.

É importante destacar que, independentemente da classificação nutricional do usuário, o sistema sempre incluirá a recomendação de atividades físicas como complemento essencial para alcançar melhores resultados e promover a saúde geral.

Tela de Receitas Exclusivas

A tela de Recomendações de Receitas exclusivas tem como principal objetivo oferecer sugestões personalizadas de dietas que atendam às diferentes necessidades e preferências dos usuários. Entre as opções disponíveis, incluem-se dietas low carb, veganas, hipercalóricas, entre outras.

O propósito dessa funcionalidade é permitir que o usuário reproduza essas receitas em casa de maneira prática e segura, incorporando-as ao seu plano alimentar diário. Todas as recomendações são elaboradas com foco em promover uma alimentação equilibrada e saudável, proporcionando ao usuário uma experiência que alia praticidade e bem-estar.

Além disso, a tela incentiva a autonomia alimentar, oferecendo opções que atendem a diferentes estilos de vida e objetivos nutricionais, sempre respeitando a individualidade e as necessidades específicas de cada usuário.

Tela do Perfil do Usuário

A tela de Perfil do Usuário oferece um espaço dedicado para que cada indivíduo personalize suas informações de forma prática e criativa. Nesta seção, o usuário poderá:

- Inserir ou atualizar seu **nome**, garantindo que o perfil reflita sua identidade.
- Adicionar ou alterar uma **foto de perfil**, tornando o ambiente mais personalizado e único.

Essa funcionalidade foi projetada para atender às preferências individuais, permitindo que cada usuário organize e configure seu perfil de acordo com suas necessidades. Com um design focado na simplicidade e na personalização, a tela promove uma experiência agradável e adaptada ao estilo de cada pessoa.

Hooks Utilizados

Os hooks são uma funcionalidade do React que permitem gerenciar estado e lidar com efeitos colaterais diretamente em componentes funcionais, eliminando a necessidade de usar componentes de classe. Eles tornam a lógica dos componentes mais simples, flexível e fácil de reutilizar. Com hooks, os desenvolvedores podem criar e organizar lógicas complexas de forma mais eficiente. Dois dos hooks mais populares no React e React Native são o `useState`, utilizado para gerenciar o estado local dos componentes, e o `useEffect`, que trata de operações como side effects e interações com API.

Em nosso projeto, utilizamos dois tipos de hooks principais: o `useState` e o `useEffect`.

- O `useState` é usado para armazenar e gerenciar estados, permitindo adicionar dados a uma variável e acessá-los posteriormente. Ele é essencial para a interação entre o front-end e o back-end, ajudando a atualizar a interface com base nas respostas do servidor ou ações do usuário.
- Já o `useEffect` é empregado para executar efeitos colaterais, como chamadas a funções ou APIs. Ele é frequentemente usado para realizar operações assíncronas, como buscar dados do servidor ou configurar ouvintes de eventos, garantindo que essas ações ocorram no momento apropriado durante o ciclo de vida do componente.

Exemplos de requisições de API

Para as requisições da API, implementamos um CRUD para gerenciar os dados do usuário. Além disso, criamos uma função no back-end para calcular o IMC (Índice de Massa Corporal). Com base no resultado, o sistema sugere a dieta ideal para o dia a dia do usuário, complementada por dicas práticas para uma alimentação equilibrada.

Estrutura de componentes

Nossa Estrutura de componentes segue uma organização modular que facilita a manutenção, escalabilidade e compreensão do código. Aqui está uma explicação geral sobre como está estruturada:

1. **Pasta components:** Contém componentes reutilizáveis que servem como blocos fundamentais da interface do usuário.
 - **cardHome.js, cardImc.js, cardMenu.js:** Representam componentes visuais, como cartões que exibem informações ou opções específicas na aplicação.
 - **Footer.js:** Pode ser o rodapé comum utilizado em várias telas para navegação ou informações adicionais.
2. Essa separação permite reutilizar esses elementos em diferentes telas, reduzindo redundâncias no código.
3. **Pasta navigations:**
 - **AppNavigator.js:** Centraliza a configuração da navegação do app, utilizando bibliotecas como React Navigation. Esse arquivo pode conter as rotas e o fluxo de telas, gerenciando como os usuários navegam entre diferentes partes da aplicação.
4. **Pasta screens:**

Contém as telas principais da aplicação, cada uma representando uma funcionalidade ou seção específica. Exemplos:

 - **FoodCoffeeScreen.js, FoodDinnerScreen.js, FoodLunchScreen.js:** Provavelmente organizam as refeições do usuário em categorias específicas (café, almoço, jantar), separando a lógica e os layouts por tipo de refeição.
 - **ImcFormScreen.js** A tela responsável por coletar dados para calcular o IMC do usuário.
 - **LoginScreen.js e SignUp.js:** Gerenciam as interfaces de autenticação, como login e cadastro.
 - **DashboardScreen.js e HomePage.js:** Servem como pontos centrais da aplicação, exibindo informações gerais e navegando para outras telas.

Essa organização separa claramente a lógica e a interface de cada funcionalidade, o que facilita futuras modificações.