

Projeto Aplicativo ADS

Uma abordagem prática de aprendizado

Fausto Toloí

Visão Geral das Tecnologias

React Native: Um framework que nos permite escrever aplicativos móveis reais usando JavaScript.

Expo: Uma plataforma que fornece ferramentas e serviços para construir aplicativos React Native com mais facilidade e rapidez.

Node.js e NPM: Ambiente de execução para JavaScript no servidor e gerenciador de pacotes, respectivamente, usados para gerenciar as dependências do projeto.

Arquitetura do Projeto

Exploramos a organização fundamental do nosso aplicativo:

- **Diretórios:** Separando nossos arquivos para melhorar a clareza (`src` , `assets` , `components`).
- **Arquivos Principais:** `App.js` coordena a navegação e os estilos, enquanto `package.json` gerencia nossas dependências.

A estrutura clara ajuda na manutenção e compreensão do projeto.

Fluxo da Aplicação

Neste slide, vamos entender como o usuário navega pelo app:

- **Navegação:** Como as telas são organizadas e interconectadas.
 - **Componentes de Navegação:** Uso de `NavigationContainer` e diferentes 'Stacks'.
- Vamos discutir o papel da navegação no contexto de um aplicativo React Native.

Estrutura de Diretórios

Cada pasta tem seu propósito:

- **src:** Coração do nosso código.
- **assets:** Onde moram imagens e outros recursos.
- **components:** Peças reutilizáveis do nosso app.
- **screens:** As vistas individuais do nosso app.

Entender essa estrutura é crucial para trabalhar de forma eficiente no projeto.

Tecnologias Detalhadas

Aprofundamos nas ferramentas que tornam nosso app especial:

- **React Native:** Escrevemos um código, mas ele roda tanto em Android quanto em iOS.
 - **Expo:** Nos permite testar rapidamente e usar recursos avançados sem configuração complexa.
 - **React Navigation:** Gerencia o fluxo entre as telas do aplicativo.
- Estas tecnologias juntas proporcionam uma experiência de desenvolvimento poderosa e flexível.

Prática: Explorando o Código

Vamos mergulhar no código de uma funcionalidade específica:

- Localizem a lógica dentro dos arquivos relevantes.
- Observem como a estrutura é organizada.
- Discutiremos as decisões por trás do design do código.

Este é o momento de entender como aplicar o que aprendemos em um contexto real.

Tela Principal - HomeScreen

A **HomeScreen** é a primeira tela que os usuários veem. Destaca as principais funcionalidades do app e proporciona uma navegação intuitiva.

- Objetivo principal da tela.
- Elementos de UI utilizados.
- Fluxo de interação do usuário.

Código da HomeScreen

```
import React from 'react';
import { View, Text } from 'react-native';

export default function HomeScreen() {
  return (
    <View>
      <Text>Bem-vindo ao App ADS</Text>
    </View>
  );
}
```

Analise como usamos componentes básicos de React Native para construir a interface.

Busca por CEP - BuscaCepScreen

A `BuscaCepScreen` permite aos usuários inserir um CEP e visualizar informações do endereço.

- Descreve a funcionalidade de busca.
- Como a UI facilita a entrada de dados.
- Exibição dos resultados da busca.

Código da BuscaCepScreen

```
import React, { useState } from 'react';
import { View, TextInput, Button } from 'react-native';

export default function BuscaCepScreen() {
  const [cep, setCep] = useState('');

  return (
    <View>
      <TextInput
        placeholder="Digite o CEP"
        value={cep}
        onChangeText={setCep}
      />
      <Button title="Buscar" onPress={() => console.log(cep)} />
    </View>
  );
}
```

Exploramos o uso de `useState` para gerenciar o estado e `TextInput` para coletar entrada do usuário.

Sobre Nós - QuemSomosScreen

A `QuemSomosScreen` fornece informações sobre a equipe ou o propósito do aplicativo.

- Conteúdo típico desta tela.
- Estratégias para manter o conteúdo informativo e envolvente.

Código da QuemSomosScreen

```
import React from 'react';
import { ScrollView, Text } from 'react-native';

export default function QuemSomosScreen() {
  return (
    <ScrollView>
      <Text>Informações sobre a equipe...</Text>
    </ScrollView>
  );
}
```

Utilizamos `ScrollView` para permitir que o conteúdo extenso seja facilmente navegável.

Informações do Aplicativo - SobreScreen

A **SobreScreen** oferece detalhes sobre o aplicativo, como versão, recursos e direitos autorais.

- Importância de fornecer transparência e créditos no app.
- Elementos incluídos nesta tela.

Código da SobreScreen

```
import React from 'react';
import { View, Text } from 'react-native';

export default function SobreScreen() {
  return (
    <View>
      <Text>Versão do Aplicativo: 1.0.0</Text>
    </View>
  );
}
```

Demonstra uma forma simples de apresentar informações estáticas ao usuário.

Pacote: React Native

- **Utilização:** Desenvolvimento de interfaces móveis.
- **No Projeto:** Usamos para criar a UI do aplicativo.
- **Exemplo de Código:**
 - Componentes como `View`, `Text`, `Button`.

Pacote: Expo

- **Utilização:** Framework e plataforma para aplicações universais React.
- **No Projeto:** Facilita o desenvolvimento, teste e implantação.
- **Exemplo de Código:**
 - Configurações no `app.json`, uso de `expo start`.

Pacote: React Navigation

- **Utilização:** Navegação entre telas no aplicativo.
- **No Projeto:** Gerencia o fluxo entre as diferentes telas e abas.
- **Exemplo de Código:**
 - Implementação de `Stack.Navigator` e `BottomTabs.Navigator`.

Pacote: React Native Paper

- **Utilização:** Componentes de interface de usuário seguindo Material Design.
- **No Projeto:** Estilização e componentes como `Button`, `Card`.
- **Exemplo de Código:**
 - Uso de `PaperProvider` e componentes estilizados.

Pacote: Expo Image

- **Utilização:** Otimização de carregamento de imagens.
- **No Projeto:** Melhora o desempenho e a eficiência no carregamento de imagens.
- **Exemplo de Código:**
 - `<Image source={require('./path/to/image.png')} /> .`

Pacote: React Native Safe Area Context

- **Utilização:** Ajuste de elementos de UI para áreas seguras.
- **No Projeto:** Evita sobreposições de conteúdo em áreas críticas (como notches).
- **Exemplo de Código:**
 - Uso de `SafeAreaView` para envolver componentes de tela.

Pacote: React Native Screens

- **Utilização:** Otimização de memória e desempenho para telas.
- **No Projeto:** Melhora a performance ao alternar entre telas.
- **Exemplo de Código:**
 - Ativação via `enableScreens()` no início do app.

Pacote: React Native Vector Icons

- **Utilização:** Biblioteca de ícones para React Native.
- **No Projeto:** Fornece ícones para todo o app.
- **Exemplo de Código:**
 - `<Icon name="rocket" size={30} color="#900" /> .`

Desenvolvimento e Estilização

- **Foco:** Como o código contribui para a estética e funcionalidade.
- **Práticas:** Estratégias de estilização e modularização do código.
- **Exemplo de Código:**
 - Estilos com `StyleSheet.create()` e uso de temas.