

Guia Passo a Passo: Navegação Entre Telas no React Native Usando React Navigation

Objetivos:

- Configurar um projeto básico de React Native.
- Implementar navegação entre três telas usando o React Navigation.
- Adicionar estilização para centralizar botões e ajustar seu tamanho.

Pré-requisitos:

- Node.js instalado.
- Editor de código (como VSCode).

Passo 1: Criar o projeto React Native

Execute o seguinte comando para criar um projeto:

```
npx create-expo-app MeuAppDeNavegacao --template
```

Passo 2: Instalar as dependências do React Navigation

Instale o núcleo da biblioteca:

```
npm install @react-navigation/native
```

Depois, instale as dependências adicionais:

```
npm install react-native-screens react-native-safe-area-context
```

Passo 3: Instalar a biblioteca de navegação em pilha

Instale o pacote de navegação em pilha:

```
npm install @react-navigation/stack
```

Passo 4: Configurar o projeto

No arquivo App.js:

```
1  import * as React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3  import { createStackNavigator } from '@react-navigation/stack';
4  import HomeScreen from './src/screens/HomeScreen';
5  import DetailsScreen from './src/screens/DetailsScreen';
6  import ProfileScreen from './src/screens/ProfileScreen';
7
8  const Stack = createStackNavigator();
9
10 export default function App() {
11   return (
12     <NavigationContainer>
13       <Stack.Navigator initialRouteName="Home">
14         <Stack.Screen name="Home" component={HomeScreen} />
15         <Stack.Screen name="Details" component={DetailsScreen} />
16         <Stack.Screen name="Profile" component={ProfileScreen} />
17       </Stack.Navigator>
18     </NavigationContainer>
19   );
20 }
```

Passo 5: Criar as telas

Agora, crie os componentes das telas com a estilização para os botões:

HomeScreen.js

```
1 import React from 'react';
2 import { View, Text, Button, StyleSheet, Dimensions } from 'react-native';
3
4 const windowWidth = Dimensions.get('window').width;
5
6 export default function HomeScreen({ navigation }) {
7   return (
8     <View style={styles.container}>
9       <Text style={styles.title}>Home Screen</Text>
10      <View style={styles.buttonContainer}>
11        <Button
12          title="Go to Details"
13          onPress={() => navigation.navigate('Details')}
14        />
15      </View>
16      <View style={styles.buttonContainer}>
17        <Button
18          title="Go to Profile"
19          onPress={() => navigation.navigate('Profile')}
20        />
21      </View>
22    </View>
23  );
24 }
```

```
1 const styles = StyleSheet.create({
2   container: {
3     flex: 1,
4     justifyContent: 'center',
5     alignItems: 'center',
6     backgroundColor: '#f0f8ff', // Cor de fundo da tela
7   },
8   title: {
9     fontSize: 24,
10    marginBottom: 20,
11  },
12  buttonContainer: {
13    backgroundColor: '#add8e6', // Cor de fundo do container do botão
14    margin: 10,
15    width: windowWidth * 0.5, // 50% da largura da tela
16    borderRadius: 5,
17  },
18 });
```


DetailsScreen.js

```
1 import React from 'react';
2 import { View, Text, Button, StyleSheet, Dimensions } from 'react-native';
3
4 const windowWidth = Dimensions.get('window').width;
5
6 export default function DetailsScreen({ navigation }) {
7   return (
8     <View style={styles.container}>
9       <Text style={styles.title}>Details Screen</Text>
10      <View style={styles.buttonContainer}>
11        <Button
12          title="Go to Home"
13          onPress={() => navigation.navigate('Home')}
14        />
15      </View>
16      <View style={styles.buttonContainer}>
17        <Button
18          title="Go to Profile"
19          onPress={() => navigation.navigate('Profile')}
20        />
21      </View>
22    </View>
23  );
24 }
```

```
1 const styles = StyleSheet.create({
2   container: {
3     flex: 1,
4     justifyContent: 'center',
5     alignItems: 'center',
6     backgroundColor: '#faf0e6', // Cor de fundo da tela
7   },
8   title: {
9     fontSize: 24,
10    marginBottom: 20,
11  },
12  buttonContainer: {
13    backgroundColor: '#ffebcd', // Cor de fundo do container do botão
14    margin: 10,
15    width: windowWidth * 0.5, // 50% da largura da tela
16    borderRadius: 5,
17  },
18 });
```


ProfileScreen.js

```
1 import React from 'react';
2 import { View, Text, Button, StyleSheet, Dimensions } from 'react-native';
3
4 const windowWidth = Dimensions.get('window').width;
5
6 export default function ProfileScreen({ navigation }) {
7   return (
8     <View style={styles.container}>
9       <Text style={styles.title}>Profile Screen</Text>
10      <View style={styles.buttonContainer}>
11        <Button
12          title="Go to Home"
13          onPress={() => navigation.navigate('Home')}
14        />
15      </View>
16      <View style={styles.buttonContainer}>
17        <Button
18          title="Go to Details"
19          onPress={() => navigation.navigate('Details')}
20        />
21      </View>
22    </View>
23  );
24 }
```

```
1 const styles = StyleSheet.create({
2   container: {
3     flex: 1,
4     justifyContent: 'center',
5     alignItems: 'center',
6     backgroundColor: '#e6e6fa', // Cor de fundo da tela
7   },
8   title: {
9     fontSize: 24,
10    marginBottom: 20,
11  },
12  buttonContainer: {
13    backgroundColor: '#dda0dd', // Cor de fundo do container do botão
14    margin: 10,
15    width: windowWidth * 0.5, // 50% da largura da tela
16    borderRadius: 5,
17  },
18 });
```

Passo 6: Executar o projeto

Execute o projeto com:

```
npx expo start
```

Explicação:

- **React Navigation:** A biblioteca usada para navegar entre as telas.
- **Stack Navigator:** Gerencia a navegação entre as telas empilhando-as, permitindo "voltar" à tela anterior.
- **navigation.navigate:** Função usada para trocar de tela programaticamente.
- **Dimensions:** Usado para obter a largura da tela.
- **container:** Centraliza os elementos na tela.
- **buttonContainer:** Define o tamanho e a cor de fundo dos botões.

Com esses passos, você terá um aplicativo básico com navegação entre três telas usando o React Navigation.

O botão Voltar será renderizado automaticamente em um navegador de pilha sempre que for possível para o usuário retornar da tela atual — em outras palavras, o botão Voltar será renderizado sempre que houver mais de uma tela na pilha. Geralmente é isso que você deseja, mas em outras circunstâncias você desejará ter um controle maior na navegação para voltar. Para isso, aqui está a atualização do código da ProfileScreen com a remoção da seta de voltar e a adição de um terceiro botão que utiliza o `goBack` para voltar à tela anterior:

```
1 <View style={styles.buttonContainer}>
2   <Button
3     title="Go to Details"
4     onPress={() => navigation.navigate('Details')}
5   />
6 </View>
7 <View style={styles.buttonContainer}>
8   <Button
9     title="Go Back"
10    onPress={() => navigation.goBack()}
11  />
12 </View>
```

Alterações:

- **Remoção da seta de voltar:** Isso pode ser feito adicionando a opção `headerShown: false` no `Stack.Screen` que define a `ProfileScreen` no `App.js`:

```
<Stack.Screen name="Profile" component={ProfileScreen} options={{ headerShown: false }} />
```

- **Botão "Go Back":** Agora existe um terceiro botão que chama `navigation.goBack()`, retornando à tela anterior.

Com essa alteração, o botão "Go Back" substitui a seta de voltar, mantendo a navegação totalmente controlada pelos botões na tela.