

React Native Navigations

Link da documentação: <https://reactnavigation.org/>

Instalação

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

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

```
expo install @react-navigation/native-stack
```

Caso esteja utilizando o React Native CLI em vez do Expo, basta substituir o `expo install` por `npm install`

Iniciando

Para começar precisamos importar dois caras, o `NavigationContainer` e o `createNativeStackNavigator` importamos também as páginas da nossa aplicação.

Todas essas coisas são feitas no nosso componente App, pois é ele que será o nosso componente principal, então é ele que fica responsável pela navegação das páginas.

Após importar tudo, nós vamos abrir em volta de tudo um `NavigationContainer` dentro dele abriremos um `StackNavigator`, e dentro dele adicionamos o `Stack.Screen` e passamos para ele `name=""` e o `component={}` que vai ser o renderizado.

```
const stack = createNativeStackNavigator

export default function App() {
  return(
    <NavigationContainer>

      <StackNavigator>
        <Stack.Screen name="Home" component={ Home } />
      </StackNavigator>
    </NavigationContainer>
  )
}
```

```

        <Stack.Screen name="Sobre" component={ Sobre } />
      </StackNavigator>

    </NavigationContainer>
  )
}

```

Configurando navegação

Vamos dizer que você está no seu componente Home e quer criar a navegação para a página de Sobre, com as rotas configuradas, vamos agora fazer a navegação no seu componente Home, para isso importamos o `useNavigation` e dentro dele passamos um `navigate` apontando para a Screen que definimos no name anteriormente.

```

export default function Home() {
  const navigation = useNavigation()

  return(
    <Button title="Ir para Sobre" onPress={() => navigation.navigate('Sobre')} />
  )
}

```

Dessa forma direcionamos o user para a página de Sobre.

Personalizações

Dentro da `Stack.Screen` podemos passar uma propriedade chamada `options={}`

Nessa propriedade nós podemos personalizar todo o header que vem por padrão.

```

<Stack.Screen
  options={{
    title: '', //nome que aparece no header
    headerStyle: {
      // aqui podemos passar a cor de fundo
    },
    headerTintColor: '', // aqui passamos a cor do texto
  }}
/

```

```
    headerShown: false / true, // aqui podemos exibir ou não o header
  }}

/>
```

Passando dados pela navegação

Podemos ainda enviar dados pela navegação, podemos criar uma função que além de navegar o user para uma página específica manda junto com ele algumas informações.

```
export default function Home() {
  const navigation = useNavigation()

  function navegaSobre() {
    // podemos enviar dados através da navegação dessa forma
    navigation.navigate('Sobre', { nome: 'julio', email: 'julio@teste.com' })
  }

  return(
    <Button title="Ir para Sobre" onPress={navegaSobre}/>
  )
}
```

Aqui além de informarmos para onde o user vai ser redirecionado, usamos uma vírgula e passamos um objeto com algumas informações que podem ser renderizadas na outra página.

Para acessar os dados que foram enviados vamos para o componente Sobre e recebemos como parâmetro uma desconstrução de route, e acessamos os dados com `route.params?.atributo`

```
export default function Sobre({route}) {

  return(
    <View>
      <Text> {route.params?.nome} </Text>
      <Text> {route.params?.email} </Text>
    </View>
  )
}
```

Como no exemplo anterior passamos nome e e-mail, aqui nós acessamos esses valores.

OBS: nós usamos a interrogação depois do params pois caso o atributo não seja enviado por algum motivo a aplicação não irá quebrar.

Podemos ainda receber os dados através de um Hook do react-navigate chamado `useRoute`

```
export default function Sobre() {  
  
  const route = useRoute()  
  
  return(  
    <View>  
      <Text> {route.params?.nome} </Text>  
      <Text> {route.params?.email} </Text>  
    </View>  
  )  
}
```

Voltar a tela anterior

Podemos criar um botão por exemplo, e no seu `onPress` passar um `navigation.goBack()` dessa forma o user irá voltar para a página anterior a que ele estava.

Podemos também zerar a nossa pilha de navegação e voltar direto para a página inicial, para isso importamos o `stackActions` do react-navigation/native e então podemos criar uma função que quando um botão for clicado voltamos para o início da pilha

```
const navigation = useNavigation()  
  
function VoltarTudo() {  
  navigation.dispatch(StackActions.popToTop())  
}
```