



# Função - React Native

PROFESSOR: CRISPIM LUIZ MARTINS

# Atenção

- ▶ No uso de IA lembre-se sempre de colocar que é React Native e a versão que está usando.
- ▶ Sendo assim, evita perder tempo e colocar bibliotecas necessárias.



▶ No final trarei todas as imagens

# Sobre páginas de navegação.

- ▶ Bibliotecas essenciais para navegação
- ▶ Darei comando install depois...
- ▶ Estas são as bibliotecas fundamentais para que a navegação funcione corretamente:
- ▶ `@react-navigation/native`
- ▶ → Biblioteca principal para navegação no React Native. Ela gerencia a navegação entre telas.
- ▶ `@react-navigation/native-stack`
- ▶ → Biblioteca específica para criar uma navegação do tipo Stack, que permite empilhar telas e navegar para frente e para trás.

# Dependências necessárias para React Navigation.

- ▶ Estas bibliotecas são requeridas para o funcionamento do React Navigation:
- ▶ `react-native-screens`
  - Otimiza o gerenciamento de telas para melhorar a performance.
- ▶ `react-native-safe-area-context`
  - Garante que o conteúdo da tela respeite áreas seguras, como recortes de tela em dispositivos modernos.
- ▶ `react-native-gesture-handler`
  - Permite gestos na interface, essenciais para navegação e componentes como Drawer e Swipe.
- ▶ `react-native-reanimated`
  - Necessário para animações fluidas na navegação (como transições entre telas).
- ▶ `react-native-vector-icons`
  - Adiciona suporte a ícones que podem ser usados na navegação e interface.

# Página Cadastro

- ▶ Sobre navegação consulte a documentação [Getting started | React Navigation](#)
- ▶ Para criar a página cadastro precisamos habilitar a navegação.
- ▶ Para isso veja se já tem instalado
- ▶ `npm install @react-navigation/native`
- ▶ `npm install react-native-screens react-native-safe-area-context react-native-gesture-handler react-native-reanimated react-native-vector-icons`
- ▶ `npx expo install react-native-screens react-native-safe-area-context`
- ▶ `npm install @react-navigation/stack`
- ▶ `npm install @react-navigation/native @react-navigation/native-stack`
- ▶ `npm install react-native-gesture-handler`
- ▶ `npm install @react-navigation/native @react-navigation/drawer`
- ▶ `npm list react-native-gesture-handler`

# Ajustar nosso App.js para navegar

```
App.tsx M X  index.tsx U  AppNavigation.tsx U
App.tsx > ...
1  import React from "react";
2  import { GestureHandlerRootView } from "react-native-gesture-handler";
3  import { StyleSheet, Text, View } from 'react-native';
4  import { createStackNavigator } from "@react-navigation/stack";
5  import AppNavigation from "../src/pages/AppNavigation";
6
7  const Stack = createStackNavigator();
8
9  export default function App() {
10   return (
11     <GestureHandlerRootView style={{ flex: 1 }}>
12       <AppNavigation />
13     </GestureHandlerRootView>
14   );
15
16 }
17
18 const styles = StyleSheet.create({
19   container: {
20     flex: 1,
21     backgroundColor: '#fff',
22     alignItems: 'center',
23     justifyContent: 'center',
24   },
25 });
26
```

Usando nossas bibliotecas. Será criada o AppNavigation, que será responsável pela navegação. Estamos aqui no App.tsx avisando em será o responsável pela navegação.



AppNavigation e quem cuida agora da navegação

Usa as bibliotecas que instalamos.

```
src > pages > AppNavigation.tsx > AppNavigation
1  import React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3  import Login from '../login/index';
4  import Cadastro from '../cadastro/index';
5  import { createNativeStackNavigator } from '@react-navigation/native-stack';
6  import Home from '../home/index';
```

```
AppNavigation.tsx U X
src > pages > AppNavigation.tsx > RootStackParamList

9  type RootStackParamList = {
10    Login: undefined;
11    Cadastro: undefined;
12  };
13
14  // ♦ Aplique o tipo no Stack Navigator
15  const Stack = createNativeStackNavigator<RootStackParamList>();
16
17
18  export default function AppNavigation() {
19    return (
20      <NavigationContainer>
21        <Stack.Navigator initialRouteName="Login">
22          <Stack.Screen name="Login" component={Login} />
23          <Stack.Screen name="Cadastro" component={Cadastro} />
24        </Stack.Navigator>
25      </NavigationContainer>
26    );
27  }
```



# Vamos criar nossas funções de teste. Em Login

```
15 export default function Login(){
16   const [email, setEmail] = useState('');
17   const [senha, setSenha] = useState('');
18
19   function getLogin(){
20     try {
21       if(!email || !senha){
22         return Alert.alert('Preencha todos os campos')
23       }
24       Alert.alert('Logado com sucesso!!!')
25     } catch (error) {
26       console.log('Erro ao logar:');
27     }
28   }
29 };
30
31 return(
```

Temos um comportamento quase igual a uma estrutura JavaScript. Criamos a função que será chamada nos elementos.

Linha 65 a 67 chama nossa função com alerta. 71 a 73 vai para cadastro

```
63     </View>
64     <View style={style.boxBottom}>
65         <TouchableOpacity style={style.button} onPress={getLogin}>
66             <Text>Entrar</Text>
67         </TouchableOpacity>
68
69         {/* Link para cadastro */}
70         {/* Link para a tela de Cadastro */}
71         <TouchableOpacity onPress={() => navigation.navigate("Cadastro")}>
72             <Text>Não tem conta? Cadastre-se aqui!</Text>
73         </TouchableOpacity>
74
75     </View>
76
77 </View>
78
79 )
80 }
```

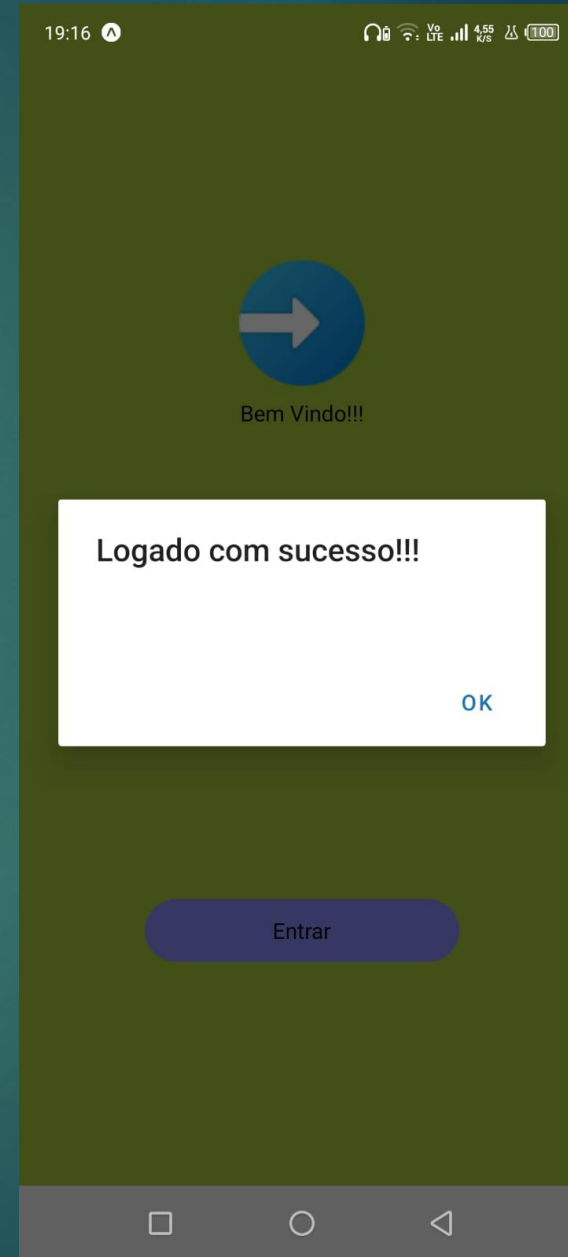
Faz a navegação

Chama  
a  
função

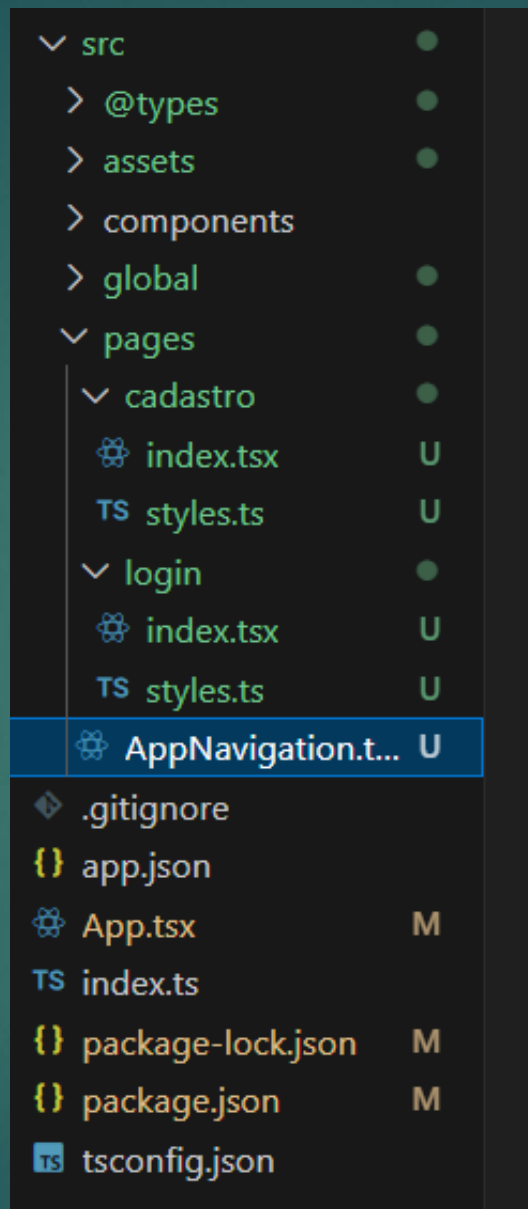
# Teste a função de login

Para a função de cadastro crie a página de cadastro primeiro.  
Se já está feita pode testar

Depois teste a navegação!!!



# Como estamos!!!



App.tsx M X

App.tsx > ...

```
1  import React from "react";
2  import { GestureHandlerRootView } from "react-native-gesture-handler";
3  import { StyleSheet, Text, View } from 'react-native';
4  import { createStackNavigator } from "@react-navigation/stack";
5  import AppNavigation from "../src/pages/AppNavigation";
6
7  const Stack = createStackNavigator();
8
9  export default function App() {
10    return (
11      <GestureHandlerRootView style={{ flex: 1 }}>
12        <AppNavigation />
13      </GestureHandlerRootView>
14    );
15  }
16
17
18  const styles = StyleSheet.create({
19    container: {
20      flex: 1,
21      backgroundColor: '#fff',
22      alignItems: 'center',
23      justifyContent: 'center',
24    },
25  });
26
```

AppNavigation.tsx U X

src > pages > AppNavigation.tsx > ...

```
1  import React from 'react';
2  import { NavigationContainer } from '@react-navigation/native';
3  import Login from '../login/index';
4  import Cadastro from '../cadastro/index';
5  import { createNativeStackNavigator } from '@react-navigation/native-stack';
6
7  // ♦ Definição correta do tipo das rotas
8  type RootStackParamList = {
9    Login: undefined;
10   Cadastro: undefined;
11 };
12
13 // ♦ Aplique o tipo no Stack Navigator
14 const Stack = createNativeStackNavigator<RootStackParamList>();
15
16
17 export default function AppNavigation() {
18   return (
19     <NavigationContainer>
20       <Stack.Navigator initialRouteName="Login">
21         <Stack.Screen name="Login" component={Login} />
22         <Stack.Screen name="Cadastro" component={Cadastro} />
23       </Stack.Navigator>
24     </NavigationContainer>
25   );
26 }
```

src &gt; pages &gt; AppNavigation.tsx &gt; ...

```
1  import React from 'react';|
2  import { NavigationContainer } from '@react-navigation/native';
3  import Login from '../login/index';
4  import Cadastro from '../cadastro/index';
5  import { createNativeStackNavigator } from '@react-navigation/native-stack';
6
7  // ♦ Definição correta do tipo das rotas
8  type RootStackParamList = {
9    Login: undefined;
10   Cadastro: undefined;
11 };
12
13 // ♦ Aplique o tipo no Stack Navigator
14 const Stack = createNativeStackNavigator<RootStackParamList>();
15
16
17 export default function AppNavigation() {
18   return (
19     <NavigationContainer>
20       <Stack.Navigator initialRouteName="Login">
21         <Stack.Screen name="Login" component={Login} />
22         <Stack.Screen name="Cadastro" component={Cadastro} />
23       </Stack.Navigator>
24     </NavigationContainer>
25   );
26 }
```



## Login - index

```
index.tsx U X
src > pages > login > index.tsx > Login
1  import React, { useState } from "react";
2
3  import {
4    Text,
5    View,
6    Image,
7    TextInput,
8    TouchableOpacity,
9    Alert,
10 } from 'react-native';
11
12 import {style} from './styles';
13 import {MaterialIcons} from '@expo/vector-icons';
14 import Logo from '../../assets/login-96.png';
15 import { themes } from "../../global/themes";
16 import { NativeStackScreenProps } from '@react-navigation/native-stack';
17 type RootStackParamList = {
18   Login: undefined;
19   Cadastro: undefined;
20 };
21
22 type Props = NativeStackScreenProps<RootStackParamList, 'Login'>;
23
24 export default function Login({ navigation }: Props){
```

```
24 export default function Login({ navigation }: Props){
25   const [email, setEmail] = useState('');
26   const [senha, setSenha] = useState('');
27
28   function getLogin(){
29     try {
30       if(!email || !senha){
31         return Alert.alert('Preencha todos os campos')
32       }
33       Alert.alert('Logado com sucesso!!!')
34
35     } catch (error) {
36       console.log('Erro ao logar:');
37     }
38   };
39
40   return(
41     <View style={style.container}>
42       <View style={style.boxTop}>
43         <Image source={Logo} />
44         <Text>Bem Vindo!!!</Text>
45       </View>
46       <View style={style.boxMid}>
47         <Text style={style.title}>E-mail</Text>
48         <View style={style.boxInputEmail}>
49           <TextInput
50             style={style.input}
51             value={email}
52             onChangeText={(e) =>setEmail(e)}/>
53           <MaterialIcons name="email" size={20} color={themas.colors.gray} />
54         </View>
```

index.tsx U X

src > pages > login > index.tsx > Login

```
24 export default function Login({ navigation }: Props){
25
26   <View style={style.container} >
27     <Text style={style.title}>Coloque sua Senha</Text>
28     <View style={style.boxInputSenha}>
29       <TextInput style={style.input}
30         value={senha}
31         onChangeText={(e) => setSenha(e)}
32         secureTextEntry={true}/>
33       <MaterialIcons name="password" size={20} color={themas.colors.gray} />
34     </View>
35   </View>
36   <View style={style.boxBottom}>
37     <TouchableOpacity style={style.button} onPress={getLogin}>
38       <Text>Entrar</Text>
39     </TouchableOpacity>
40
41     { /* Link para cadastro */ }
42     { /* Link para a tela de Cadastro */ }
43     <TouchableOpacity onPress={() => navigation.navigate("Cadastro")}>
44       <Text>Não tem conta? Cadastre-se aqui!</Text>
45     </TouchableOpacity>
46   </View>
47 </View>
48 )
49 }
```

# Cadastro index


```
index.tsx U X
src > pages > cadastro > index.tsx > Cadastro
1  import React, { useState } from "react";
2  import {
3    Text,
4    View,
5    TextInput,
6    TouchableOpacity,
7    Alert,
8  } from "react-native";
9  import { MaterialIcons } from "@expo/vector-icons";
10 import { style } from "../styles";
11
12 export default function Cadastro() {
13   const [nome, setNome] = useState("");
14   const [email, setEmail] = useState("");
15   const [senha, setSenha] = useState("");
16
17   function handleCadastro() {
18     try {
19       if (!nome || !email || !senha) {
20         return Alert.alert("Preencha todos os campos");
21       }
22       Alert.alert("Cadastro realizado com sucesso!");
23     } catch (error) {
24       console.log("Erro ao cadastrar:", error);
25     }
26   }
27
28   return (
```

```
12 export default function Cadastro() {  
28   return (  
29     <View style={style.container}>  
30       <Text style={style.title}>Página de Cadastro</Text>  
31  
32       {/* Campo Nome */}  
33       <Text style={style.label}>Nome</Text>  
34       <View style={style.inputContainer}>  
35         <TextInput  
36           style={style.input}  
37           value={nome}  
38           onChangeText={(text) => setNome(text)}  
39           placeholder="Digite seu nome"  
40         />  
41         <MaterialIcons name="person" size={20} color="gray" />  
42       </View>  
43  
44       {/* Campo Email */}  
45       <Text style={style.label}>Email</Text>  
46       <View style={style.inputContainer}>  
47         <TextInput  
48           style={style.input}  
49           value={email}  
50           onChangeText={(text) => setEmail(text)}  
51           placeholder="Digite seu email"  
52           keyboardType="email-address"  
53         />  
54         <MaterialIcons name="email" size={20} color="gray" />  
55       </View>  
}
```

index.tsx U X

src > pages > cadastro > index.tsx > Cadastro

```
12  export default function Cadastro() {  
56  
57      {/* Campo Senha */}  
58      <Text style={style.label}>Senha</Text>  
59      <View style={style.inputContainer}>  
60          <TextInput  
61              style={style.input}  
62              value={senha}  
63              onChangeText={(text) => setSenha(text)}  
64              placeholder="Digite sua senha"  
65              secureTextEntry={true}  
66          />  
67          <MaterialIcons name="lock" size={20} color="gray" />  
68      </View>  
69  
70      {/* Botão de Cadastro */}  
71      <TouchableOpacity style={style.button} onPress={handleCadastro}>  
72          <Text style={style.buttonText}>Cadastrar</Text>  
73      </TouchableOpacity>  
74  </View>  
75  );  
76  }
```



Os estilos busque nos slides  
anteriores