	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Móveis</p>	<p align="center">Projeto com as telas de autenticação</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

Nessa aula vamos trabalhar com as telas de cadastrar e login. Inicialmente verifique se o seu projeto utiliza o expo versão 51. Caso não esteja utilizando crie um projeto novo seguindo o PDF publicado no sigaa com o nome



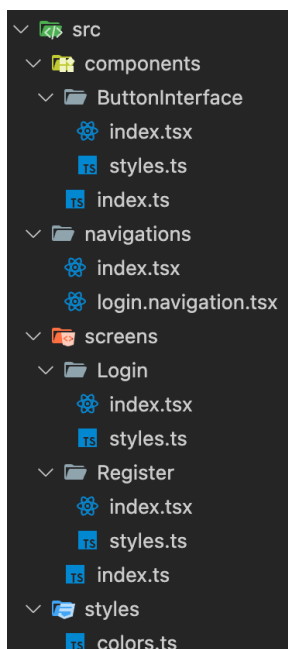
[Criando um projeto Expo React-native com TypeScript](#)

Feito isso, você precisará instalar os pacotes e realizar as configurações para uso dos recursos de navegação. Para isso, você precisará seguir o PDF publicado no sigaa com o nome



[React Navigation Stack Bottom Tabs e Drawer](#)

Com esses pacotes instalados, copie a pasta src do seu projeto caso queira continuá-lo, ou crie um com a seguinte estrutura de pastas



O arquivo de cor possui as definições ao lado.

```
export const colors = {
  primary: '#37474F',
  secondary: '#C51162',
  third: '#D9D9D9',
  black: '#000',
  white: '#FFF'
}
```

O componente ButtonInterface foi utilizado para aproveitar uma única definição para criar os botões de Enviar, Salvar e Voltar.

Abaixo tem-se o arquivo de estilo e o index.tsx com o código que recebe os parâmetros e altera o estilo do botão.

```
import { StyleSheet } from 'react-native';
import { colors } from '../styles/colors';

export const styles = StyleSheet.create({
  buttonPrimary: {
    backgroundColor: colors.primary,
    borderRadius: 5,
    margin: 10,
  },
  buttonSecondary: {
    backgroundColor: colors.secondary,
    borderRadius: 5,
    margin: 10,
  },
  buttonThird: {
    backgroundColor: colors.third,
    borderRadius: 5,
    margin: 10,
  },
  text: {
    color: colors.white,
    fontWeight: "bold",
    padding: 10,
    fontSize: 18,
    textAlign: "center"
  }
});
```

```
import { TouchableOpacityProps, TouchableOpacity, Text } from 'react-native'
import { styles } from './styles'
You, há 3 minutos | 1 author (You)
export interface IBInterface extends TouchableOpacityProps {
  onPressI: () => void
  title: string
  type: 'primary' | 'secondary' | 'third'
}
export function ButtonInterface({ onPressI, title, type, ...rest }: IBInterface) {
  return (
    <TouchableOpacity style={
      type == "primary" ? styles.buttonPrimary :
      type == "secondary" ? styles.buttonSecondary :
      styles.buttonThird
    }
      onPress={onPressI}
      {...rest}
    >
      <Text style={styles.text}>{title}</Text>
    </TouchableOpacity>
  )
}
```

Os arquivos que estão na pasta navigations possuem os seguintes conteúdos:

```
src > navigations > index.tsx > ...
You, há 1 segundo | 1 author (You)
1 import React from "react";
2 import { NavigationContainer } from '@react-navigation/native';
3 import { LoginNavigation } from "../login.navigation";
4 export function Navigation() {
5   return (
6     <NavigationContainer>
7       <LoginNavigation />
8     </NavigationContainer>
9   )
10 }
```

O arquivo index.tsx define o container de navegação

```
src > navigations > login.navigation.tsx > ...
You, há 1 segundo | 1 author (You)
1 import { createStackNavigator, StackNavigationProp } from '@react-navigation/stack';
2 import { ScreenLogin, ScreenRegister } from '../screens'
3 type LoginStackParamList = {
4   Login: undefined
5   Register: undefined
6 }
7 type LoginScreenNavigationProp = StackNavigationProp<LoginStackParamList, 'Login'>
8 export type LoginTypes = {
9   navigation: LoginScreenNavigationProp
10 }
11 export function LoginNavigation() {
12   const Stack = createStackNavigator<LoginStackParamList>()
13   return (
14     <Stack.Navigator id='login' screenOptions={{ headerShown: false }}>
15       <Stack.Screen name='Login' component={ScreenLogin} />
16       <Stack.Screen name='Register' component={ScreenRegister} />
17     </Stack.Navigator>
18   )
19 }
```

O arquivo login.navigation.tsx monta uma Stack Navigator para mover entre as telas de Login e Register que é a de cadastrar.

```
src > screens > Login > index.tsx > ...
You, há 4 horas | 1 author (You)
1 import React, { useState } from 'react';
2 import { KeyboardAvoidingView, View, Text, TextInput, Alert } from 'react-native';
3 import { MaterialIcons, Entypo } from "@expo/vector-icons";
4 import { styles } from '../styles';
5 import { colors } from '../../styles/colors';
6 import { ComponentButtonInterface } from '../../components';
7 import { LoginTypes } from '../../navigations/login.navigation';
You, há 4 horas | 1 author (You)
8 export interface IAuthenticate {
9   email?: string;
10   password?: string;
11 }
12 export function Login({ navigation }: LoginTypes) {
13   const [data, setData] = useState<IAuthenticate>();
14   async function handleSignIn() {
15     if (data?.email && data.password) {
16       console.log(data)
17     } else {
18       Alert.alert("Preencha todos os campos!!!");
19     }
20   }
21   function handleRegister() {
22     navigation.navigate("Register")
23   }
24   function handleChange(item: IAuthenticate) {
25     setData({ ...data, ...item });
26   }
```

A tela de Login possui o código ao lado.

Esta primeira parte mostra os imports a interface utilizada para tipar os dados que serão enviados para a API e as funções importantes para recuperar os dados o input, verificar se todos os dados foram preenchidos e navegar entre as telas.

```
src > screens > Login > index.tsx > ...
12 export function Login({ navigation }: LoginTypes) {
27   return (
28     <View style={styles.container}>
29       <KeyboardAvoidingView>
30         <Text style={styles.title}>Login</Text>
31         <View style={styles.formRow}>
32           <MaterialIcons name="email" style={styles.icon} />
33           <TextInput
34             placeholderTextColor={colors.third}
35             style={styles.input}
36             placeholder="Email"
37             keyboardType="email-address"
38             autoCapitalize="none"
39             onChangeText={(i) => handleChange({ email: i })}
40           />
41         </View>
42         <View style={styles.formRow}>
43           <Entypo name="key" style={styles.icon} />
44           <TextInput
45             placeholderTextColor={colors.third}
46             style={styles.input}
47             placeholder="Senha"
48             secureTextEntry={true}
49             autoCapitalize="none"
50             onChangeText={(i) => handleChange({ password: i })}
51           />
52         </View>
53         <ComponentButtonInterface title='Login' type='primary' onPressI={handleSignIn} />
54         <ComponentButtonInterface title='Cadastre-se' type='secondary' onPressI={handleRegister} />
55       </KeyboardAvoidingView>
56     </View>
57   );
58 }
```

A segunda parte monta a interface com os componentes e as configurações necessárias.

```
src > screens > Login > styles.ts > ...
You, há 1 segundo | 1 autor (You)
1 import { StyleSheet } from 'react-native';
2 import { colors } from '../styles/colors';
3 export const styles = StyleSheet.create({
4   container: {
5     flex: 1,
6     alignItems: "center",
7     justifyContent: "center"
8   },
9   title: {
10    fontSize: 30,
11    fontWeight: "bold",
12    textAlign: 'center',
13    color: colors.black
14  },
15   formRow: {
16    margin: 10,
17    flexDirection: "row",
18    alignItems: "center",
19    borderWidth: 0.7,
20    borderColor: colors.primary,
21    borderRadius: 5
22  },
23   icon: {
24    fontSize: 28,
25    color: colors.primary,
26    padding: 5
27  },
28   input: {
29    fontSize: 18,
30    padding: 10,
31    width: "70%"
32  },
33 })
```

O arquivo de estilo é o mesmo para o login e o registrar.

```
src > screens > Register > index.tsx > ...
You, há 4 horas | 1 author (You)
1  import React, { useState } from 'react';
2  import { KeyboardAvoidingView, View, Text, TextInput, Alert } from 'react-native';
3  import { MaterialIcons, Entypo, Ionicons } from '@expo/vector-icons';
4  import { styles } from '../styles';
5  import { colors } from '../../styles/colors';
6  import { LoginTypes } from '../../navigations/login.navigation';
7  import { ComponentButtonInterface } from '../../components';
You, há 4 horas | 1 author (You)
8  export interface IRegister {
9      name?: string
10     email?: string
11     password?: string
12 }
13 export function Register({ navigation }: LoginTypes) {
14     const [data, setData] = useState<IRegister>();
15     async function handleRegister() {
16         if (data?.email && data.name && data.password) {
17             console.log(data)
18         } else {
19             Alert.alert("Preencha todos os campos!!!");
20         }
21     }
22     function handleGoBack() {
23         navigation.navigate('Login')
24     }
25     function handleChange(item: IRegister) {
26         setData({ ...data, ...item });
27     }
28     return (
29         <View style={styles.container}>
30             <KeyboardAvoidingView>
31                 <Text style={styles.title}>Cadastre-se</Text>
32                 <View style={styles.formRow}>
33                     <Ionicons name="person" style={styles.icon} />
34                     <TextInput
35                         placeholderTextColor={colors.third}
36                         style={styles.input}
37                         placeholder="Nome"
38                         onChangeText={(i) => handleChange({ name: i })}
39                     />
40                 </View>
41                 <View style={styles.formRow}>
42                     <MaterialIcons name="email" style={styles.icon} />
43                     <TextInput
44                         placeholderTextColor={colors.third}
45                         style={styles.input}
46                         placeholder="Email"
47                         keyboardType="email-address"
48                         autoCapitalize="none"
49                         onChangeText={(i) => handleChange({ email: i })}
50                     />
51                 </View>
52                 <View style={styles.formRow}>
53                     <Entypo name="key" style={styles.icon} />
54                     <TextInput
55                         placeholderTextColor={colors.third}
56                         style={styles.input}
57                         placeholder="Senha"
58                         secureTextEntry={true}
59                         autoCapitalize="none"
60                         onChangeText={(i) => handleChange({ password: i })}
61                     />
62                 </View>
63                 <ComponentButtonInterface title='Salvar' type='secondary' onPress={handleRegister} />
64                 <ComponentButtonInterface title='Voltar' type='primary' onPress={handleGoBack} />
65             </KeyboardAvoidingView>
66         </View>
67     );
68 }
```

A tela de Cadastrar possui o código ao lado.

Esta primeira parte mostra os imports a interface utilizada para tipar os dados que serão enviados para a API e as funções importantes para recuperar os dados o input, verificar se todos os dados foram preenchidos e navegar entre as telas.

A segunda parte monta a interface com os componentes e as configurações necessárias.



Centro Federal de Educação Tecnológica de Minas Gerais
Campus VIII – Varginha
Curso Técnico em Informática

Disciplina
Lab. Aplicações Móveis

**Projeto com as telas de
autenticação**

Professor
Lázaro Eduardo da Silva

```
App.tsx > ...  
You, há 4 horas | 1 author (You)  
1 import { StatusBar } from 'expo-status-bar';  
2 import { Navigation } from './src/navigation'  
3  
4 export default function App() {  
5   return (  
6     <>  
7       <Navigation />  
8       <StatusBar style="auto" />  
9     </>  
10  );  
11 }
```

Por fim, o arquivo App.tsx deve chamar o componente Navigation.

Caso queira manter a informação na barra de status, mantenho o componente StatusBar, caso não, basta chamar o Navigation no return

Feito isso, teste sua aplicação e verifique se os dados preenchidos no formulário estão chegando até as variáveis de estado.

Bom trabalho!