Mobile Application Development

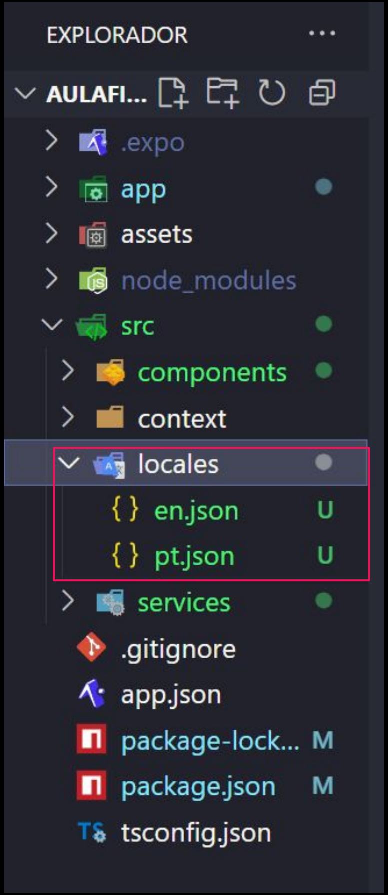
Prof. Fernando Pinéo

Internacionalização

Instalar as dependências

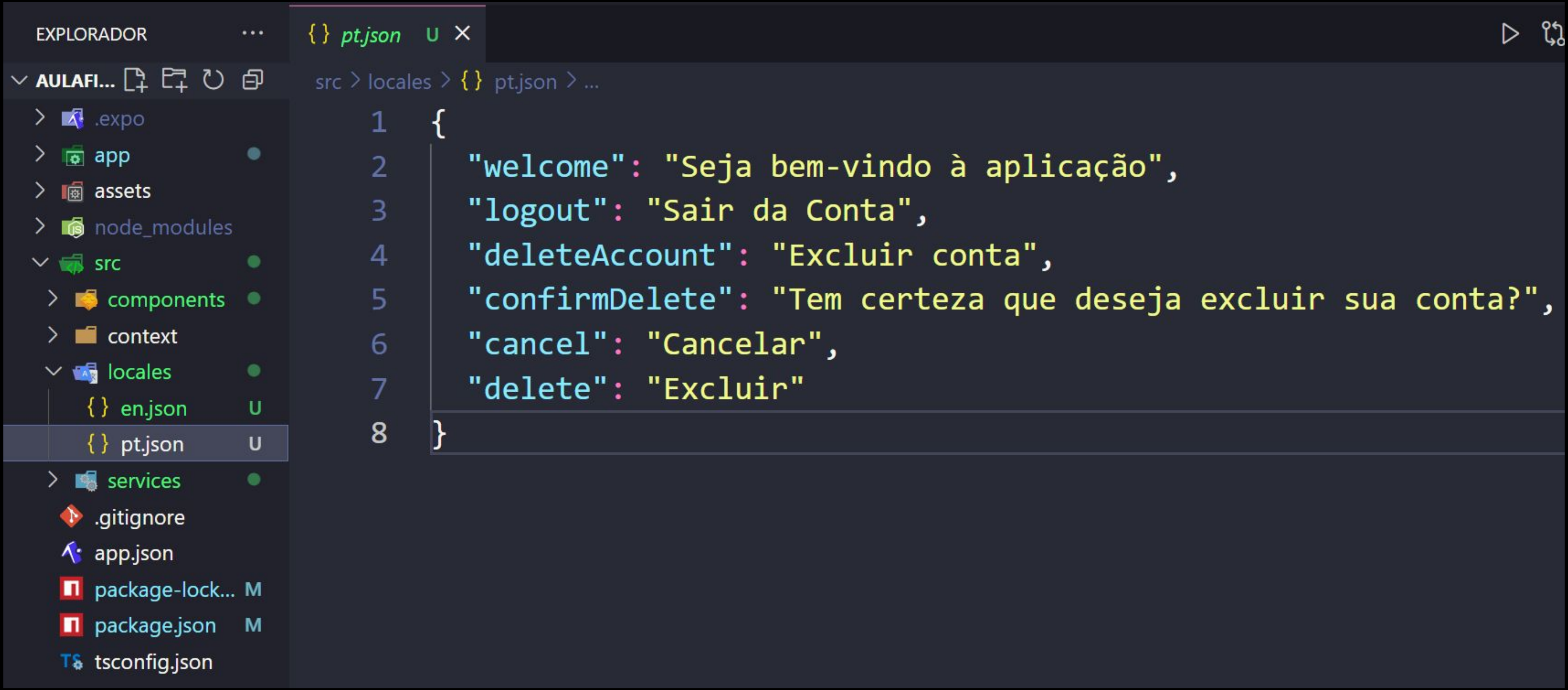
```
npm install react-i18next  i18next  react-native-localize
```

Criando os arquivos de tradução



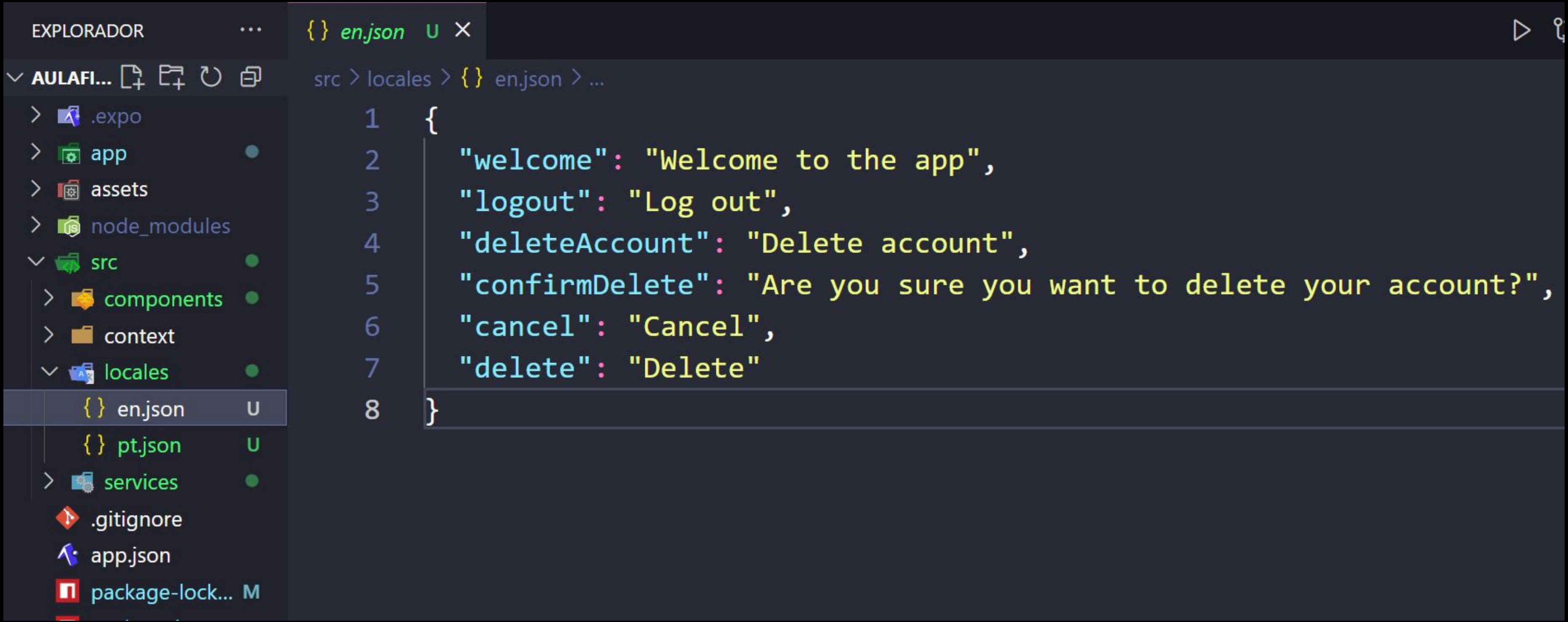
pt.json

FIAP



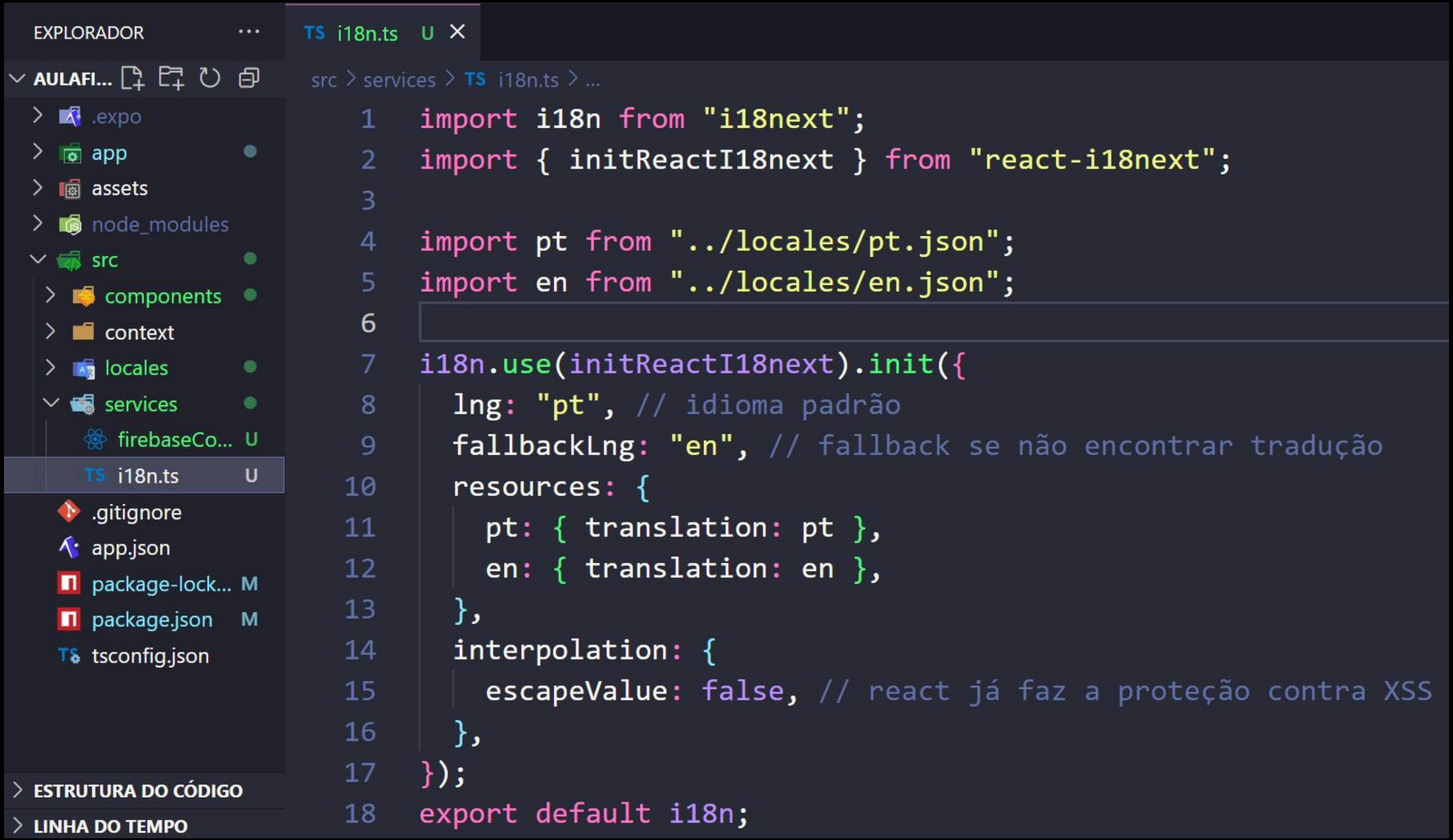
```json
{
  "welcome": "Seja bem-vindo à aplicação",
  "logout": "Sair da Conta",
  "deleteAccount": "Excluir conta",
  "confirmDelete": "Tem certeza que deseja excluir sua conta?",
  "cancel": "Cancelar",
  "delete": "Excluir"
}
```

en.json



```json
{
  "welcome": "Welcome to the app",
  "logout": "Log out",
  "deleteAccount": "Delete account",
  "confirmDelete": "Are you sure you want to delete your account?",
  "cancel": "Cancel",
  "delete": "Delete"
}
```

Configurando o arq i18n.ts

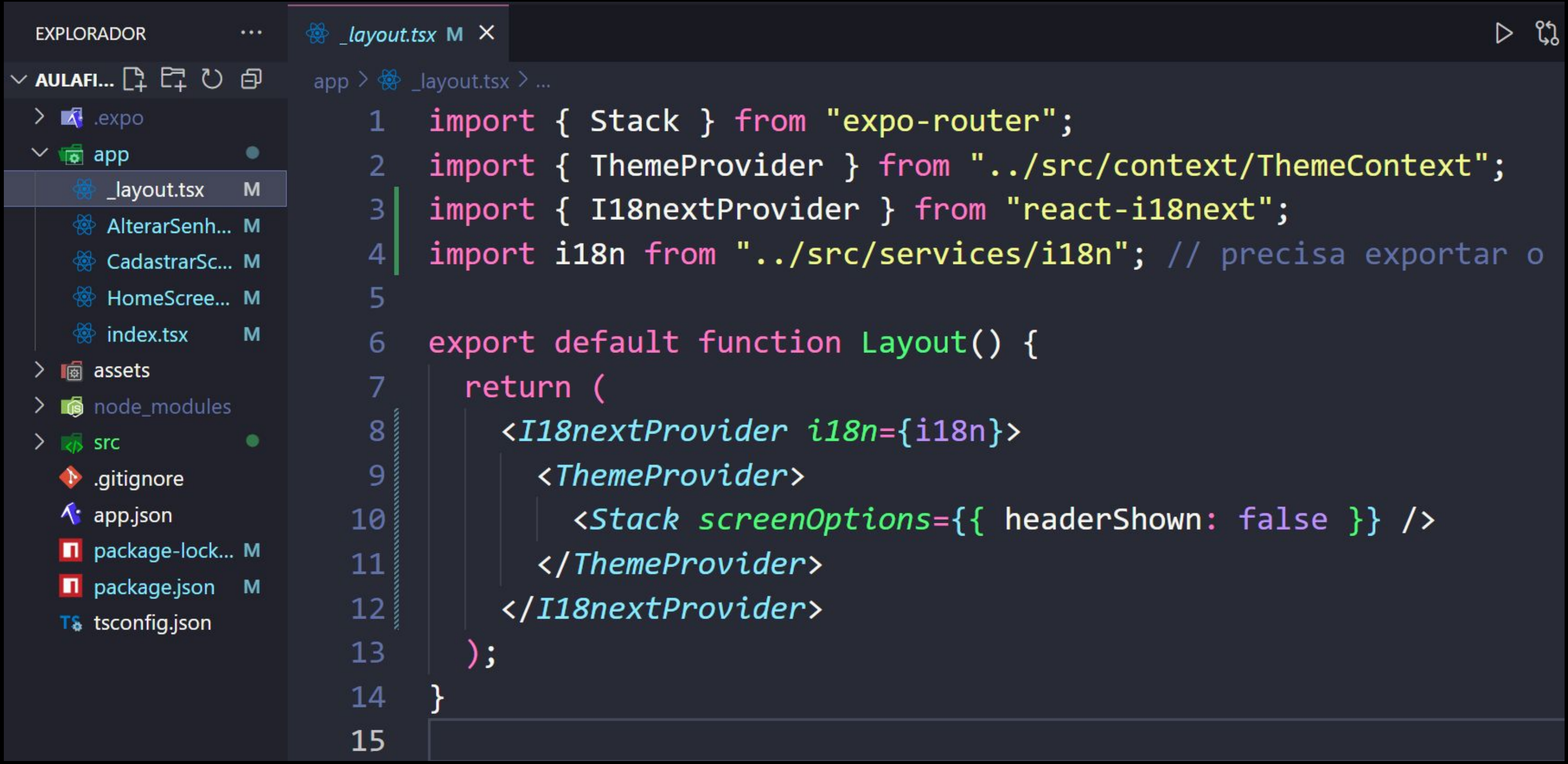


```ts
import i18n from "i18next";
import { initReactI18next } from "react-i18next";

import pt from "../locales/pt.json";
import en from "../locales/en.json";

i18n.use(initReactI18next).init({
  lng: "pt", // idioma padrão
  fallbackLng: "en", // fallback se não encontrar tradução
  resources: {
    pt: { translation: pt },
    en: { translation: en },
  },
  interpolation: {
    escapeValue: false, // react já faz a proteção contra XSS
  },
});
export default i18n;
```

Integrando ao App

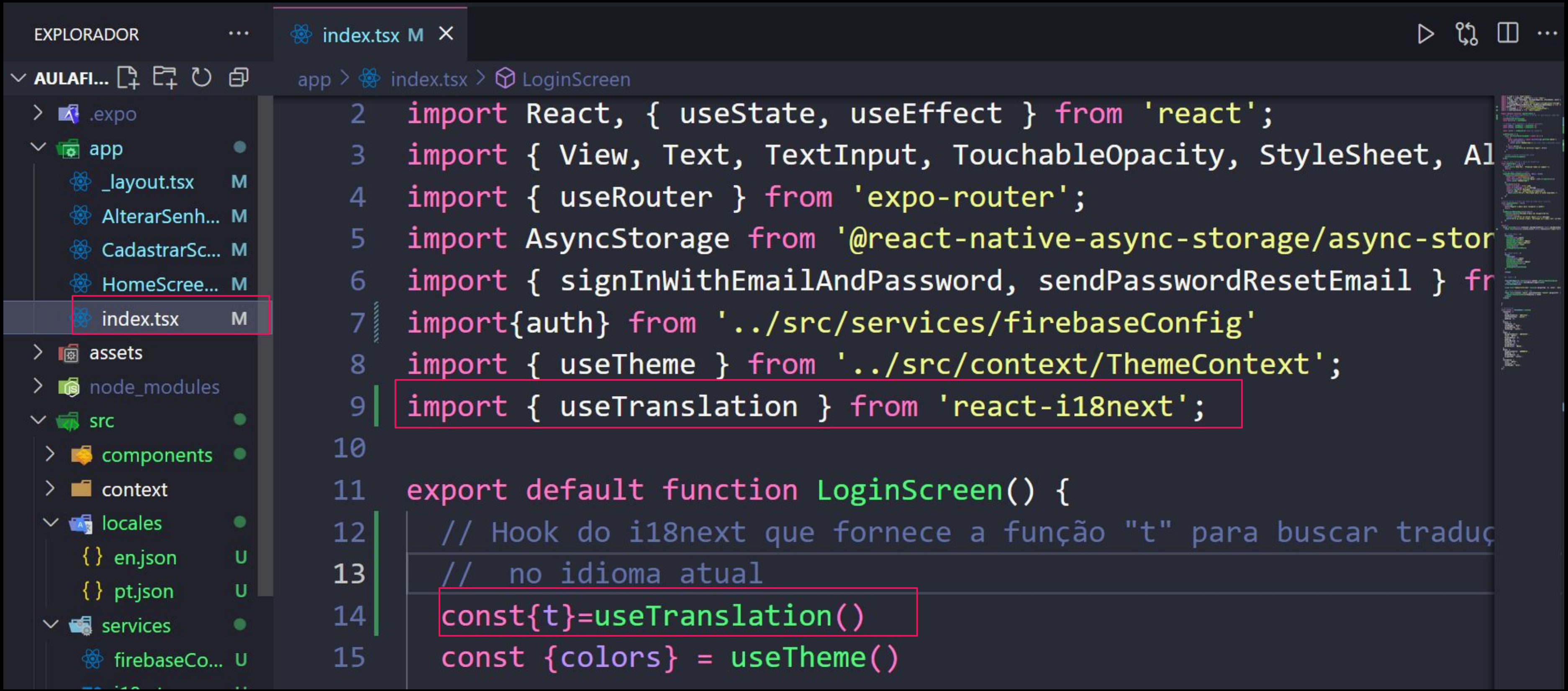


```tsx
import { Stack } from "expo-router";
import { ThemeProvider } from "../src/context/ThemeContext";
import { I18nextProvider } from "react-i18next";
import i18n from "../src/services/i18n"; // precisa exportar o

export default function Layout() {
  return (
    <I18nextProvider i18n={i18n}>
      <ThemeProvider>
        <Stack screenOptions={{ headerShown: false }} />
      </ThemeProvider>
    </I18nextProvider>
  );
}

```

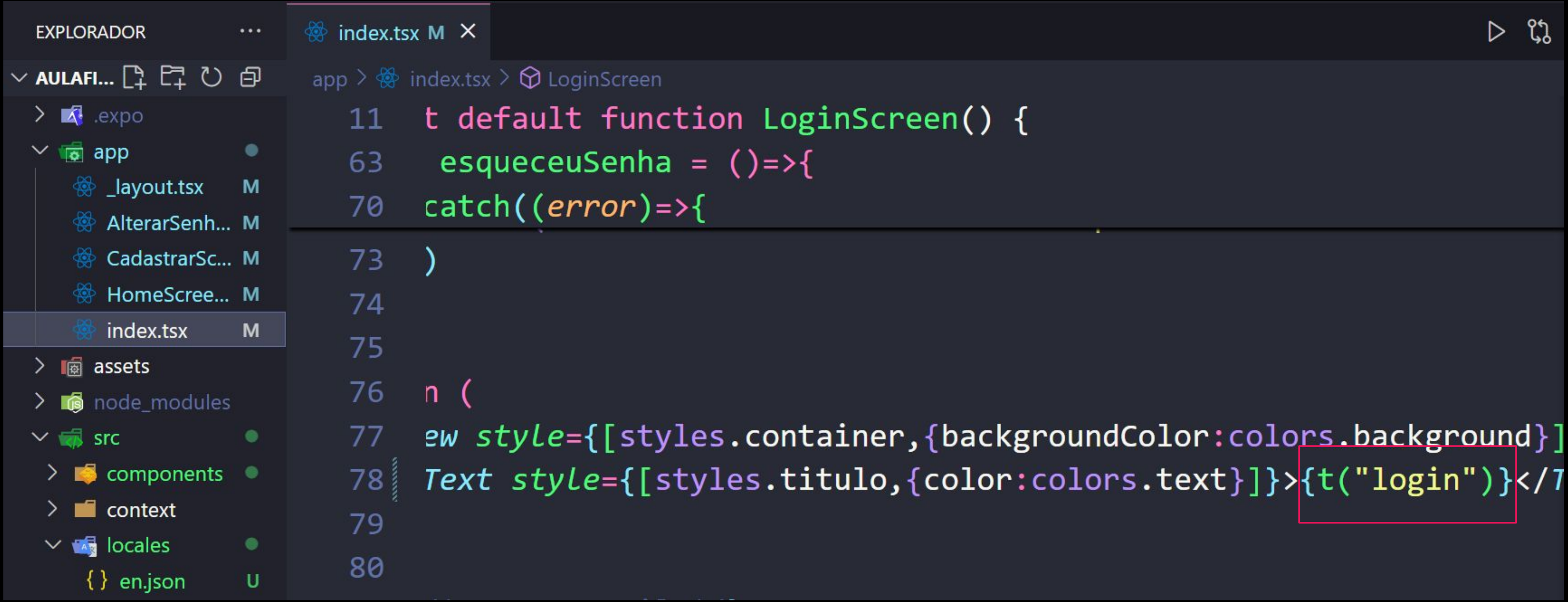
Utilizando na tela de login(index.tsx)



```
import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, TouchableOpacity, StyleSheet, Al
import { useRouter } from 'expo-router';
import AsyncStorage from '@react-native-async-storage/async-stor
import { signInWithEmailAndPassword, sendPasswordResetEmail } fr
import{auth} from '../src/services/firebaseConfig'
import { useTheme } from '../src/context/ThemeContext';
import { useTranslation } from 'react-i18next';

export default function LoginScreen() {
  // Hook do i18next que fornece a função "t" para buscar traduç
  //  no idioma atual
  const{t}=useTranslation()
  const {colors} = useTheme()
```

Configurando a tradução para msg inicial

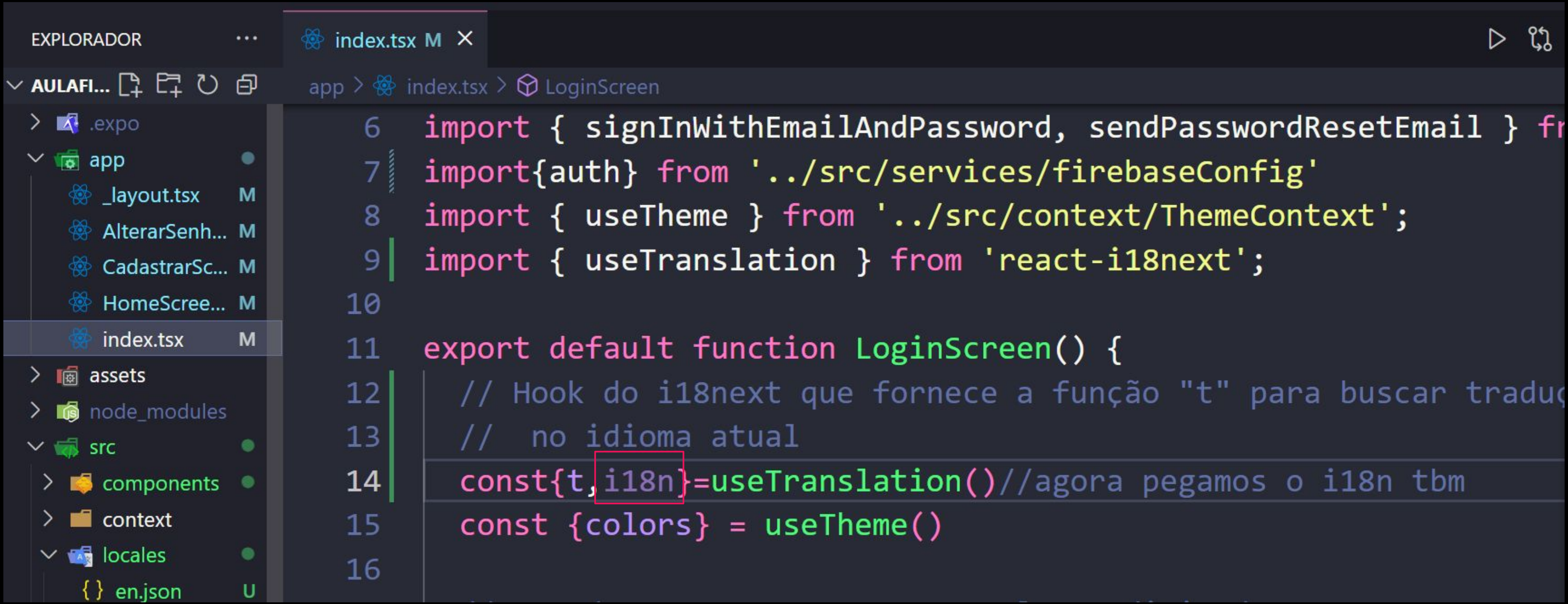


```
t default function LoginScreen() {
 esqueceuSenha = ()=>{
catch((error)=>{
)


n (
ew style={[styles.container,{backgroundColor:colors.background}]
Text style={[styles.titulo,{color:colors.text}]}>{t("login")}</T
```

Incluindo um botão para alterar o idioma

Pegando o i18n

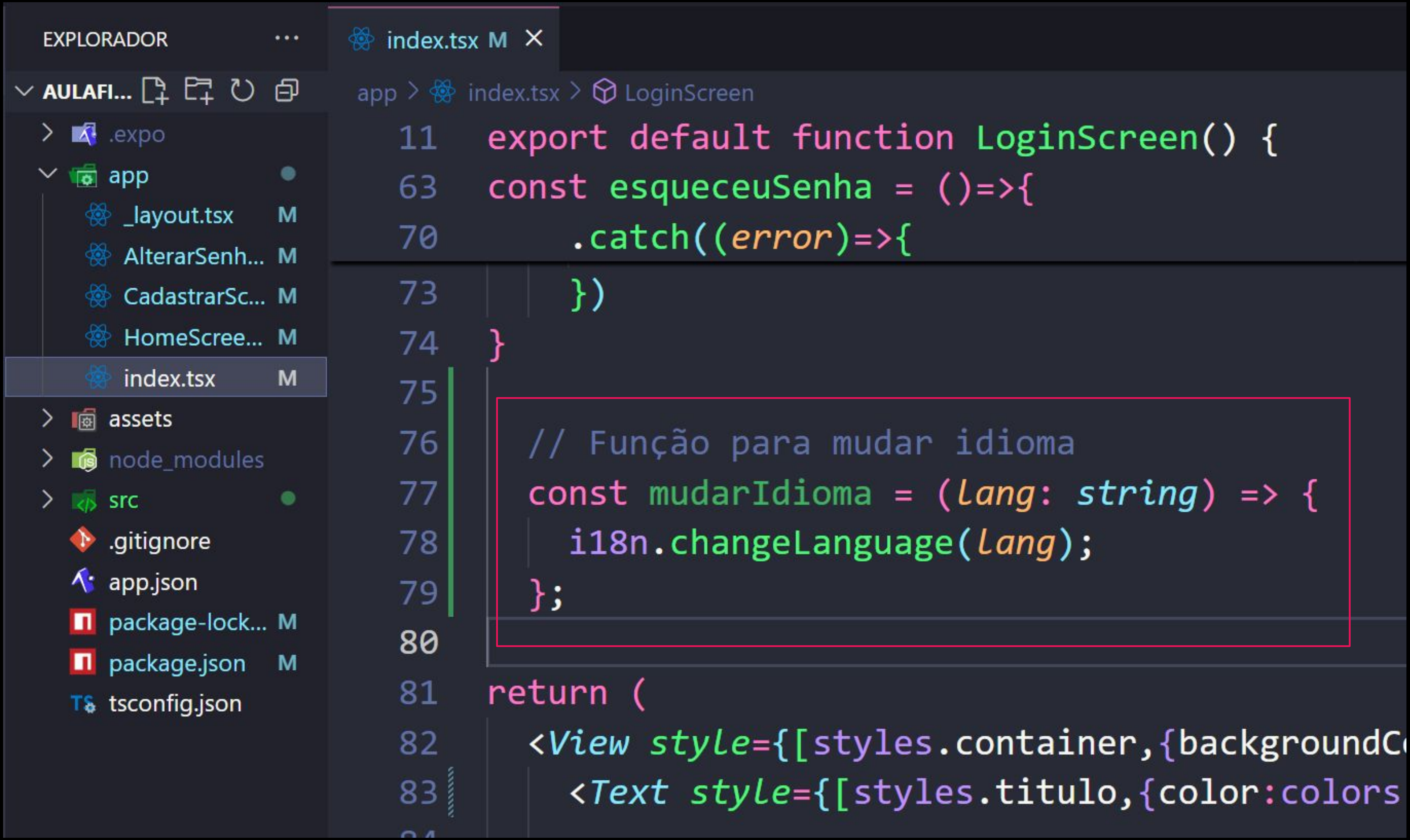


```
import { signInWithEmailAndPassword, sendPasswordResetEmail } fr
import{auth} from '../src/services/firebaseConfig'
import { useTheme } from '../src/context/ThemeContext';
import { useTranslation } from 'react-i18next';

export default function LoginScreen() {
  // Hook do i18next que fornece a função "t" para buscar traduç
  //  no idioma atual
  const{t,i18n}=useTranslation()//agora pegamos o i18n tbm
  const {colors} = useTheme()

```

Função para alterar o idioma

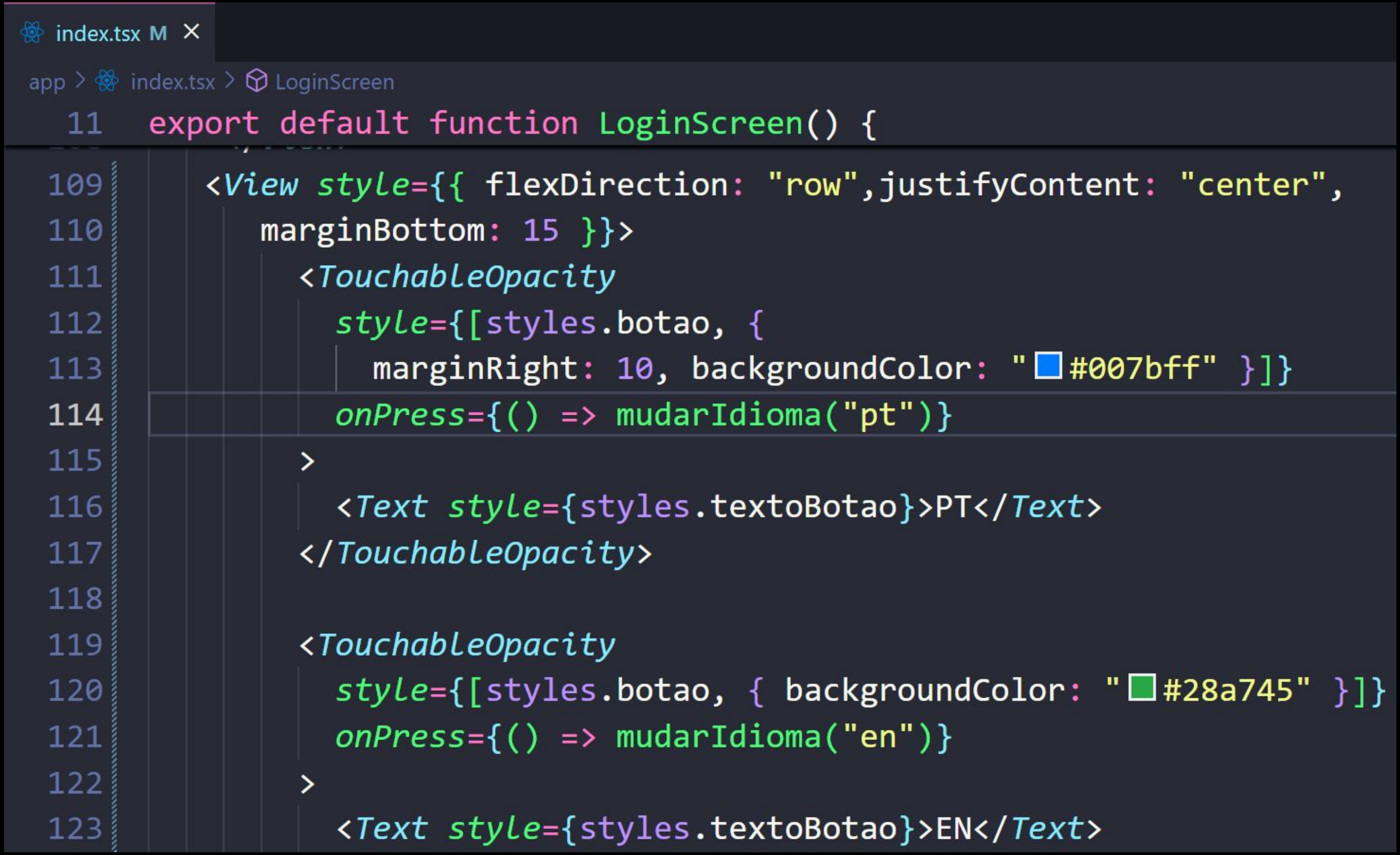


```
export default function LoginScreen() {
const esqueceuSenha = ()=>{
    .catch((error)=>{
    })
}

  // Função para mudar idioma
  const mudarIdioma = (lang: string) => {
    i18n.changeLanguage(lang);
  };

return (
  <View style={[styles.container,{backgroundC
    <Text style={[styles.titulo,{color:colors
```

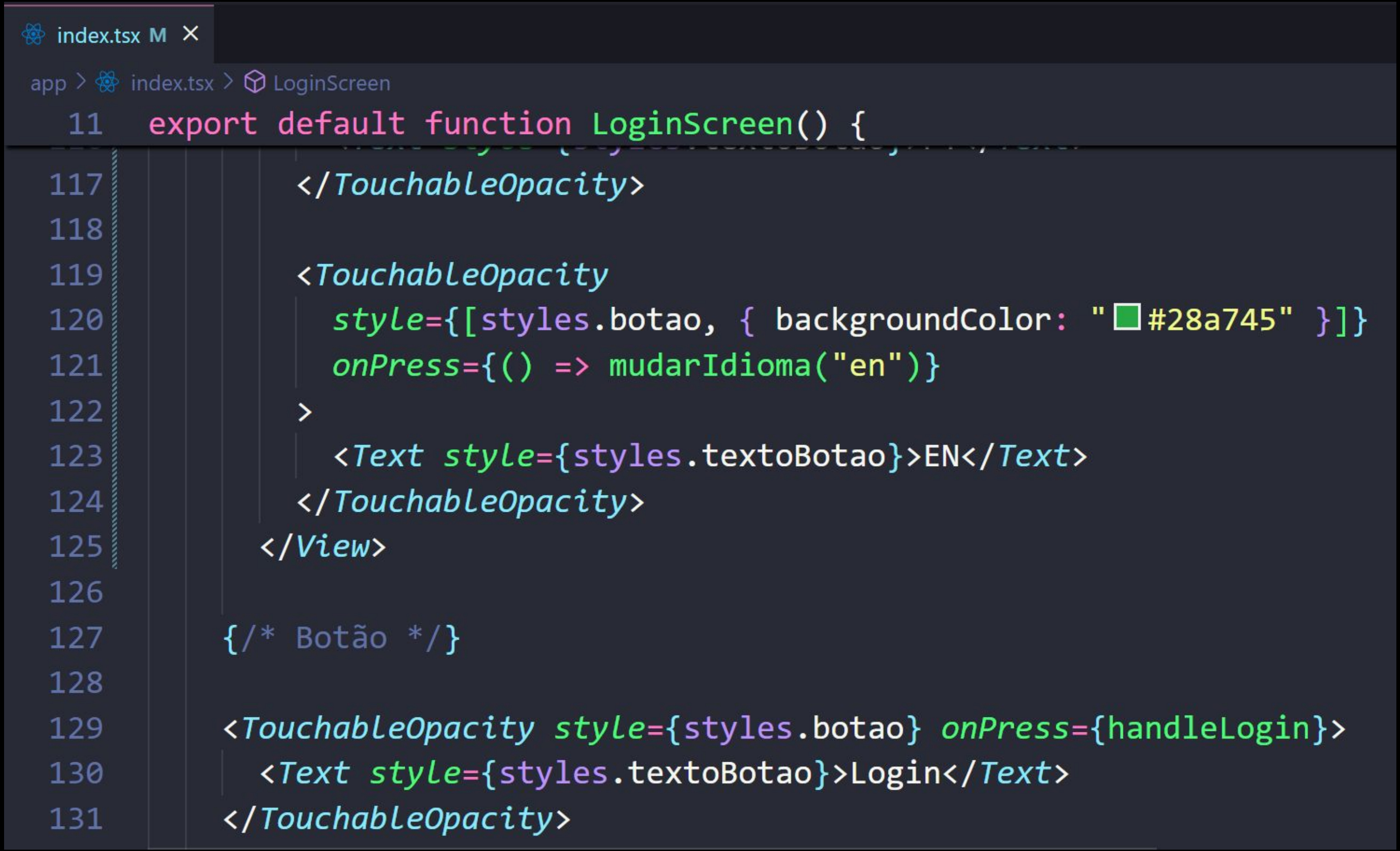
Criando o visual para os botões



```
export default function LoginScreen() {
    <View style={{ flexDirection: "row",justifyContent: "center",
       marginBottom: 15 }}>
         <TouchableOpacity
           style={[styles.botao, {
             marginRight: 10, backgroundColor: "#007bff" }]}
           onPress={() => mudarIdioma("pt")}
         >
           <Text style={styles.textoBotao}>PT</Text>
         </TouchableOpacity>

         <TouchableOpacity
           style={[styles.botao, { backgroundColor: "#28a745" }]}
           onPress={() => mudarIdioma("en")}
         >
           <Text style={styles.textoBotao}>EN</Text>
```

Criando o visual para os botões



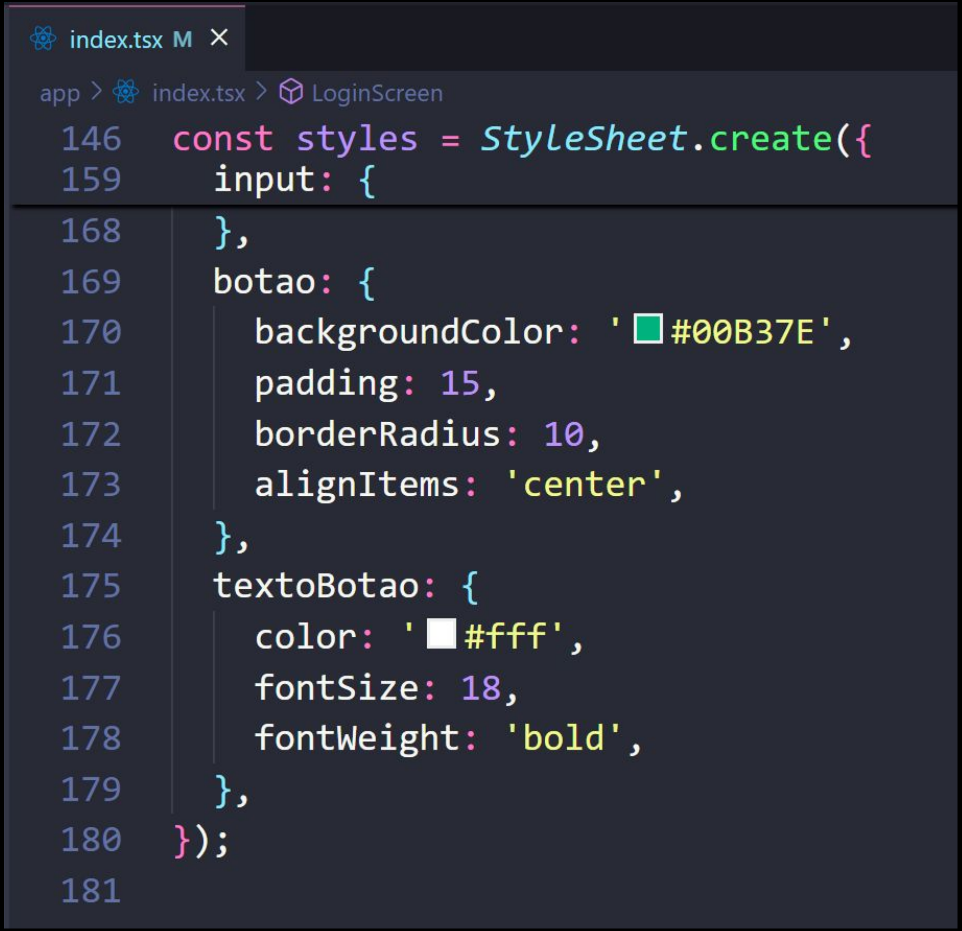
```tsx
export default function LoginScreen() {
        </TouchableOpacity>

        <TouchableOpacity
          style={[styles.botao, { backgroundColor: "#28a745" }]}
          onPress={() => mudarIdioma("en")}
        >
          <Text style={styles.textoBotao}>EN</Text>
        </TouchableOpacity>
      </View>

    {/* Botão */}

    <TouchableOpacity style={styles.botao} onPress={handleLogin}>
      <Text style={styles.textoBotao}>Login</Text>
    </TouchableOpacity>
```

Estilo dos botões



```
const styles = StyleSheet.create({
  input: {
  },
  botao: {
    backgroundColor: '#00B37E',
    padding: 15,
    borderRadius: 10,
    alignItems: 'center',
  },
  textoBotao: {
    color: '#fff',
    fontSize: 18,
    fontWeight: 'bold',
  },
});
```

Dúvidas?