

Mobile Application Development

Prof. Fernando Pinéo

Internacionalização

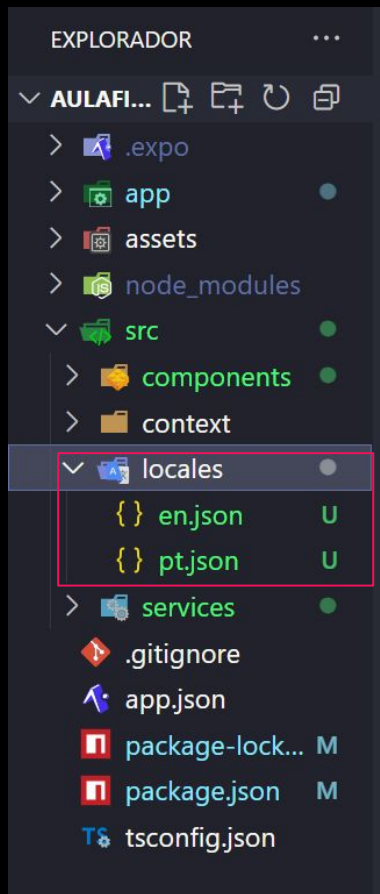
Instalar as dependências

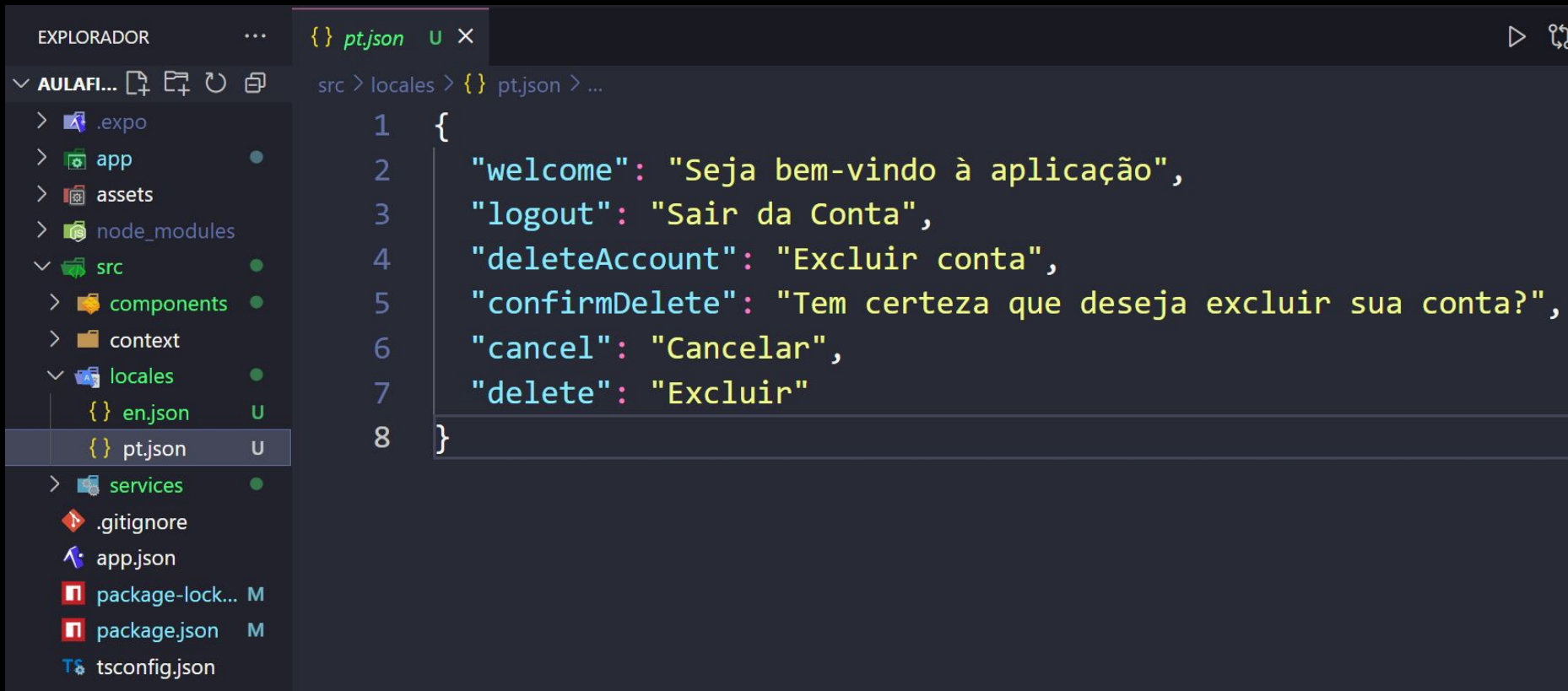


```
npm install react-i18next i18next react-native-localize
```

Criando os arquivos de tradução

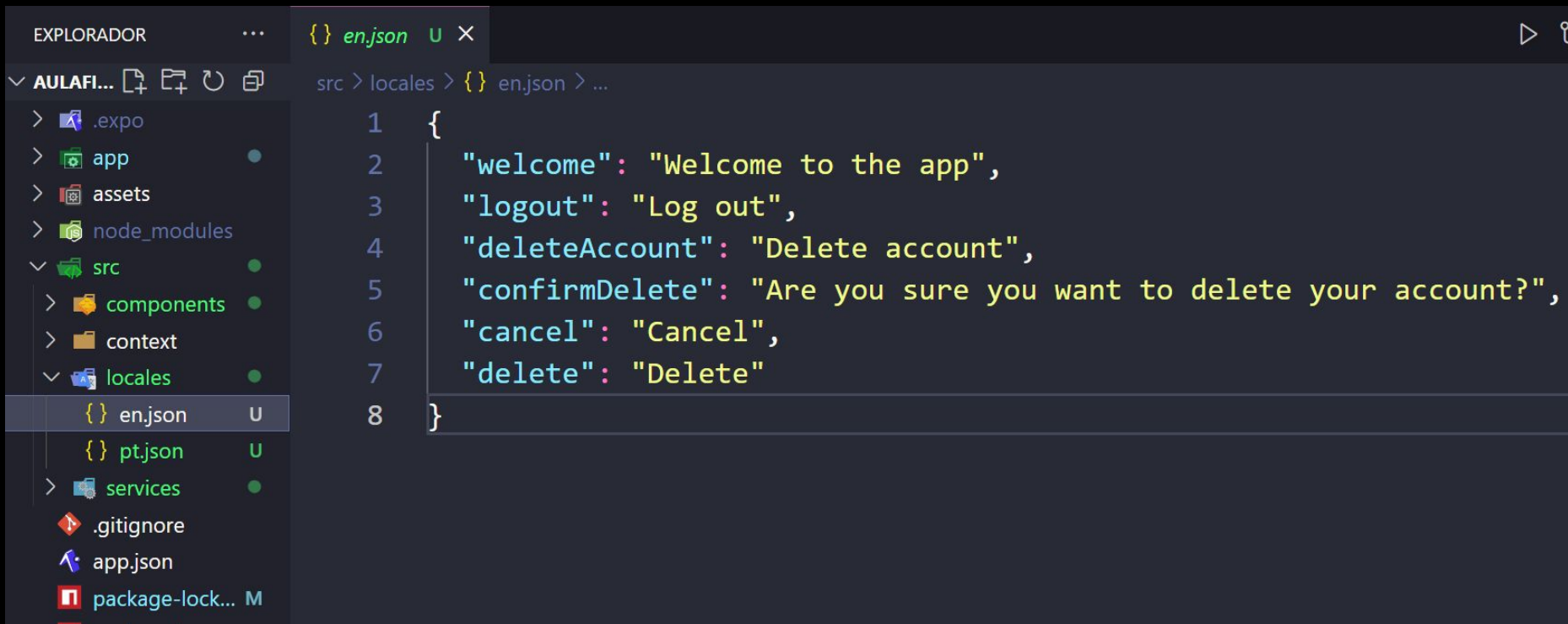
FIAP





The image shows a screenshot of the Visual Studio Code editor interface. On the left, the 'EXPLORADOR' (Explorer) sidebar displays the project structure. The file 'pt.json' is selected under the 'locales' directory. The main editor area shows the content of 'pt.json', which is a JSON object with localization strings for Portuguese. The file path 'src > locales > {} pt.json > ...' is visible above the code editor.

```
src > locales > {} pt.json > ...  
1  {  
2    "welcome": "Seja bem-vindo à aplicação",  
3    "logout": "Sair da Conta",  
4    "deleteAccount": "Excluir conta",  
5    "confirmDelete": "Tem certeza que deseja excluir sua conta?",  
6    "cancel": "Cancelar",  
7    "delete": "Excluir"  
8  }
```



The image shows a screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar is open, showing a file tree for a project named 'AULAFI...'. The tree structure is as follows:

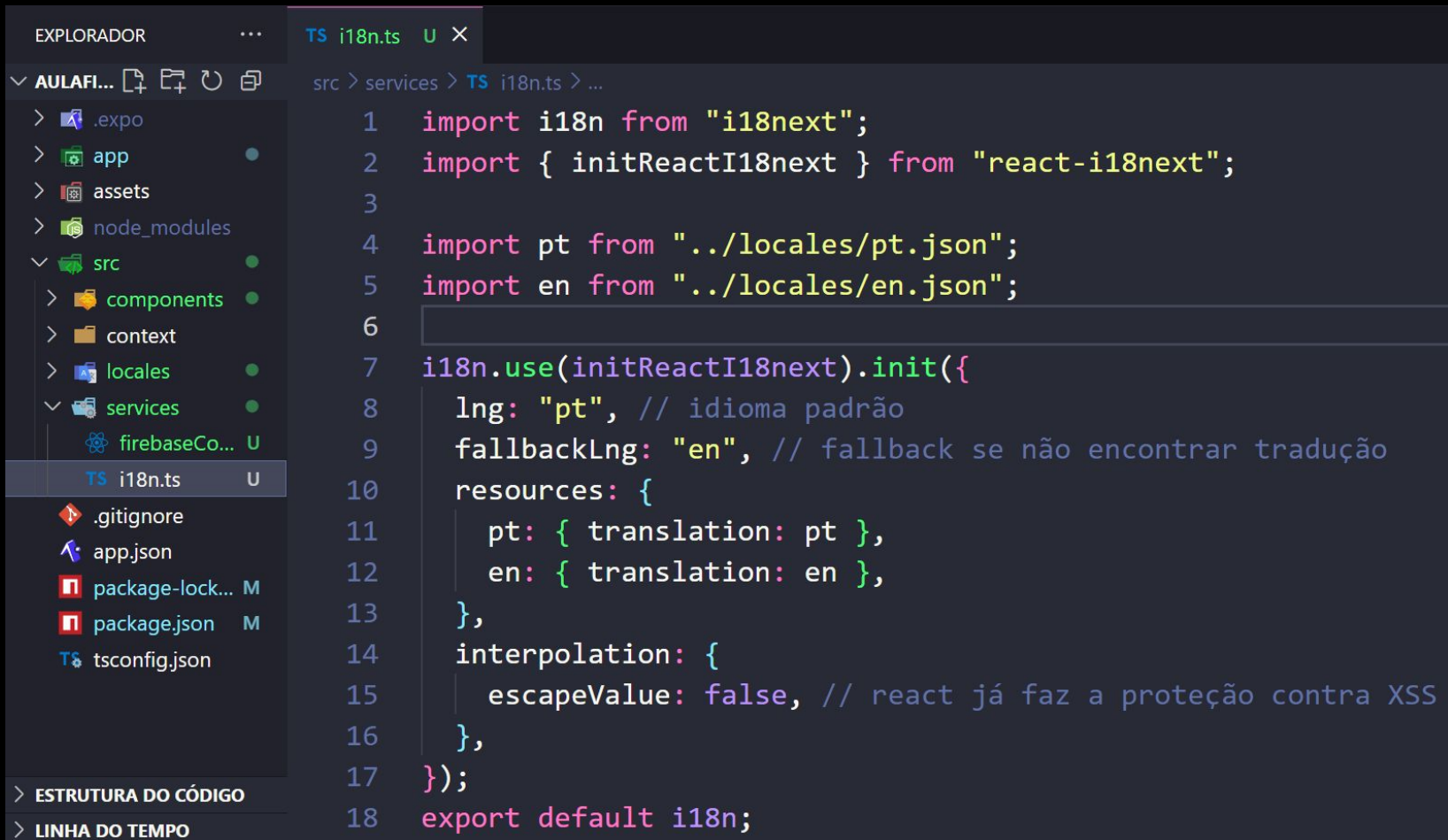
- src
 - components
 - context
 - locales
 - en.json (selected)
 - pt.json
 - services
- .gitignore
- app.json
- package-lock...

The main editor area displays the content of 'en.json' with the following JSON code:

```
1 {  
2   "welcome": "Welcome to the app",  
3   "logout": "Log out",  
4   "deleteAccount": "Delete account",  
5   "confirmDelete": "Are you sure you want to delete your account?",  
6   "cancel": "Cancel",  
7   "delete": "Delete"  
8 }
```

Configurando o arq i18n.ts

FIA/P



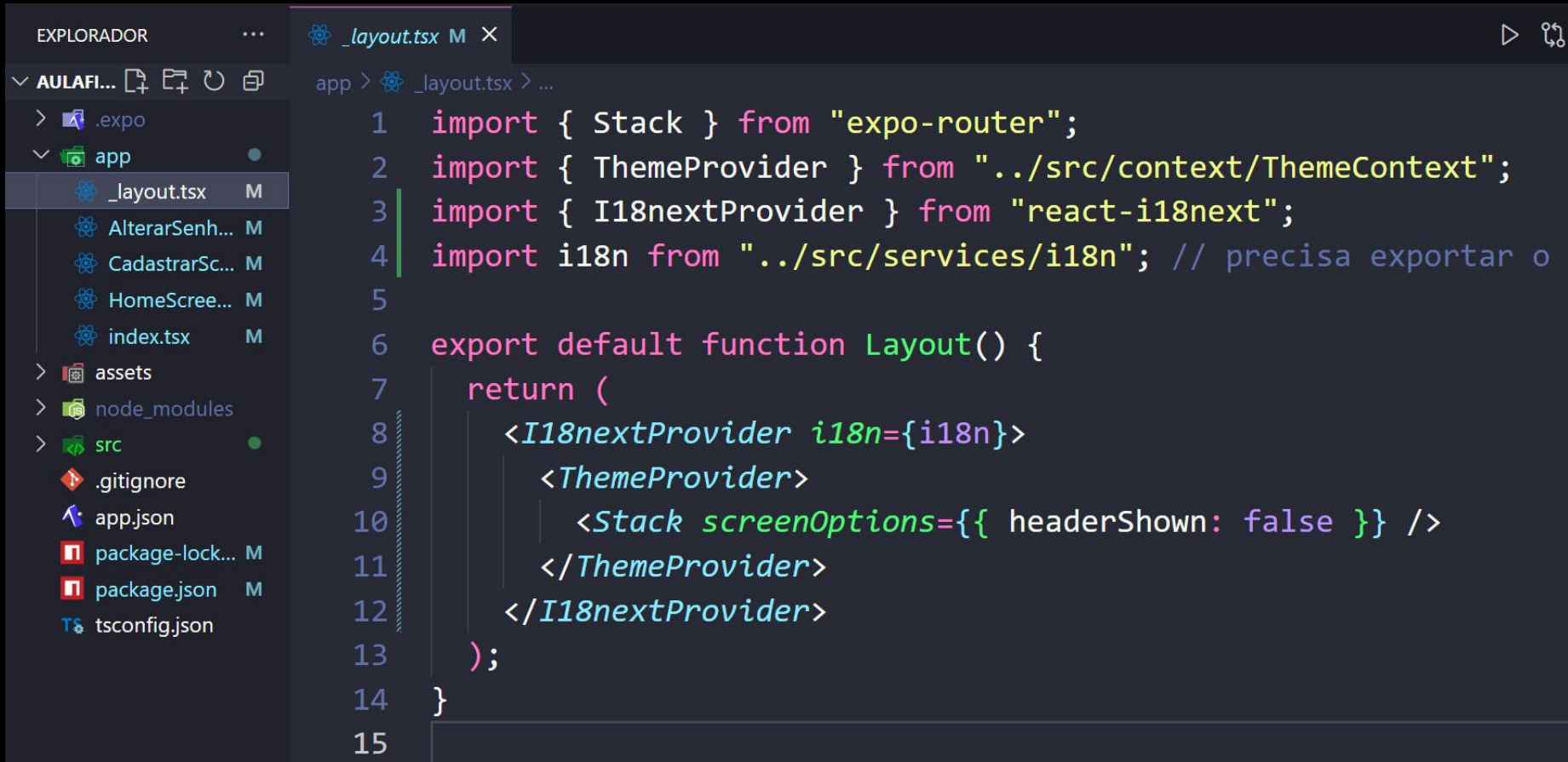
The image shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .expo, app, assets, node_modules, src, components, context, locales, and services. The services folder is expanded, showing firebaseCo... and TS i18n.ts. The code editor shows the content of TS i18n.ts, which is a TypeScript file for configuring i18n. The code includes imports for i18nnext, react-i18next, and locale files (pt.json and en.json). It then uses i18n.use to initialize the i18n instance with a default language of 'pt', a fallback language of 'en', and a set of resources for both languages. The interpolation options are also configured to escape values to prevent XSS. Finally, the i18n instance is exported as the default export.

```
src > services > TS i18n.ts > ...

1  import i18n from "i18next";
2  import { initReactI18next } from "react-i18next";
3
4  import pt from "../locales/pt.json";
5  import en from "../locales/en.json";
6
7  i18n.use(initReactI18next).init({
8    lng: "pt", // idioma padrão
9    fallbackLng: "en", // fallback se não encontrar tradução
10   resources: {
11     pt: { translation: pt },
12     en: { translation: en },
13   },
14   interpolation: {
15     escapeValue: false, // react já faz a proteção contra XSS
16   },
17 });
18 export default i18n;
```

Integrando ao App

FIAP



```
EXPLORADOR  ...  _layout.tsx M X

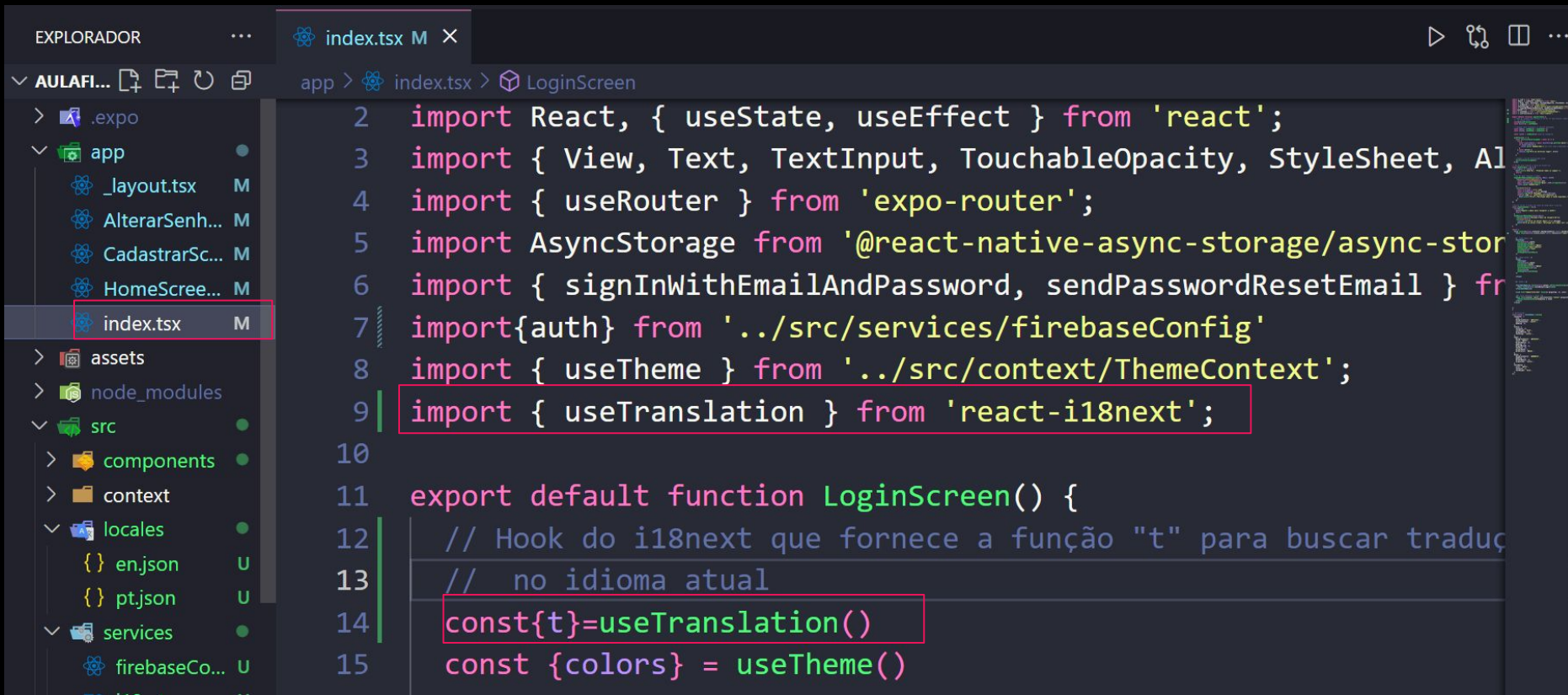
v AULAFI...  [+][+][↺][📄]
> .expo
v app
  _layout.tsx M
  AlterarSenh... M
  CadastrarSc... M
  HomeScree... M
  index.tsx M
> assets
> node_modules
> src
  .gitignore
  app.json
  package-lock... M
  package.json M
  tsconfig.json

app > _layout.tsx > ...

1  import { Stack } from "expo-router";
2  import { ThemeProvider } from "../src/context/ThemeContext";
3  import { I18nextProvider } from "react-i18next";
4  import i18n from "../src/services/i18n"; // precisa exportar o
5
6  export default function Layout() {
7    return (
8      <I18nextProvider i18n={i18n}>
9        <ThemeProvider>
10          <Stack screenOptions={{ headerShown: false }} />
11        </ThemeProvider>
12      </I18nextProvider>
13    );
14  }
15
```

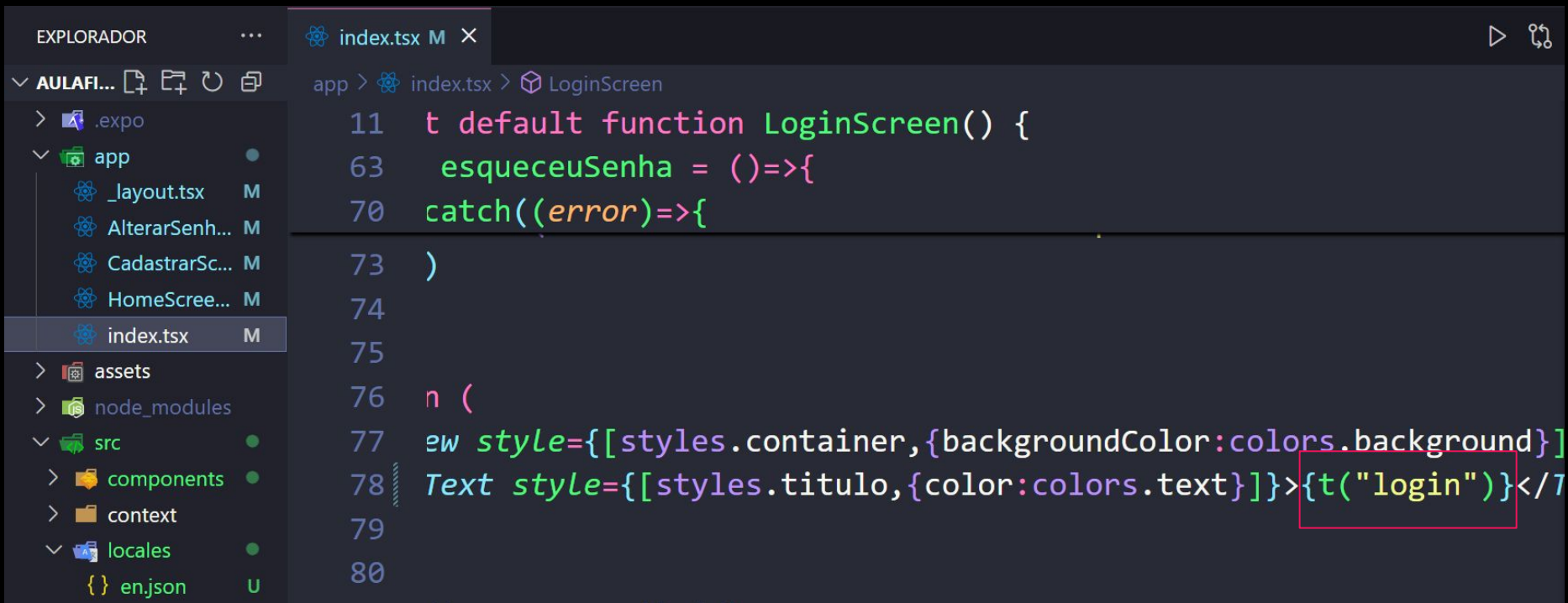

Utilizando na tela de login(index.tsx)

FILAP



```
EXPLORADOR  ...  index.tsx M X  > AULAFI...  > .expo  > app  _layout.tsx M  AlterarSenh... M  CadastrarSc... M  HomeScrie... M  index.tsx M  assets  node_modules  src  components  context  locales  en.json U  pt.json U  services  firebaseCo... U  2 import React, { useState, useEffect } from 'react';  3 import { View, Text, TextInput, TouchableOpacity, StyleSheet, Al  4 import { useRouter } from 'expo-router';  5 import AsyncStorage from '@react-native-async-storage/async-stor  6 import { signInWithEmailAndPassword, sendPasswordResetEmail } fr  7 import { auth } from '../src/services/firebaseConfig'  8 import { useTheme } from '../src/context/ThemeContext';  9 import { useTranslation } from 'react-i18next';  10  11 export default function LoginScreen() {  12   // Hook do i18next que fornece a função "t" para buscar traduç  13   // no idioma atual  14   const { t } = useTranslation()  15   const { colors } = useTheme()
```

Configurando a tradução para msg inicial



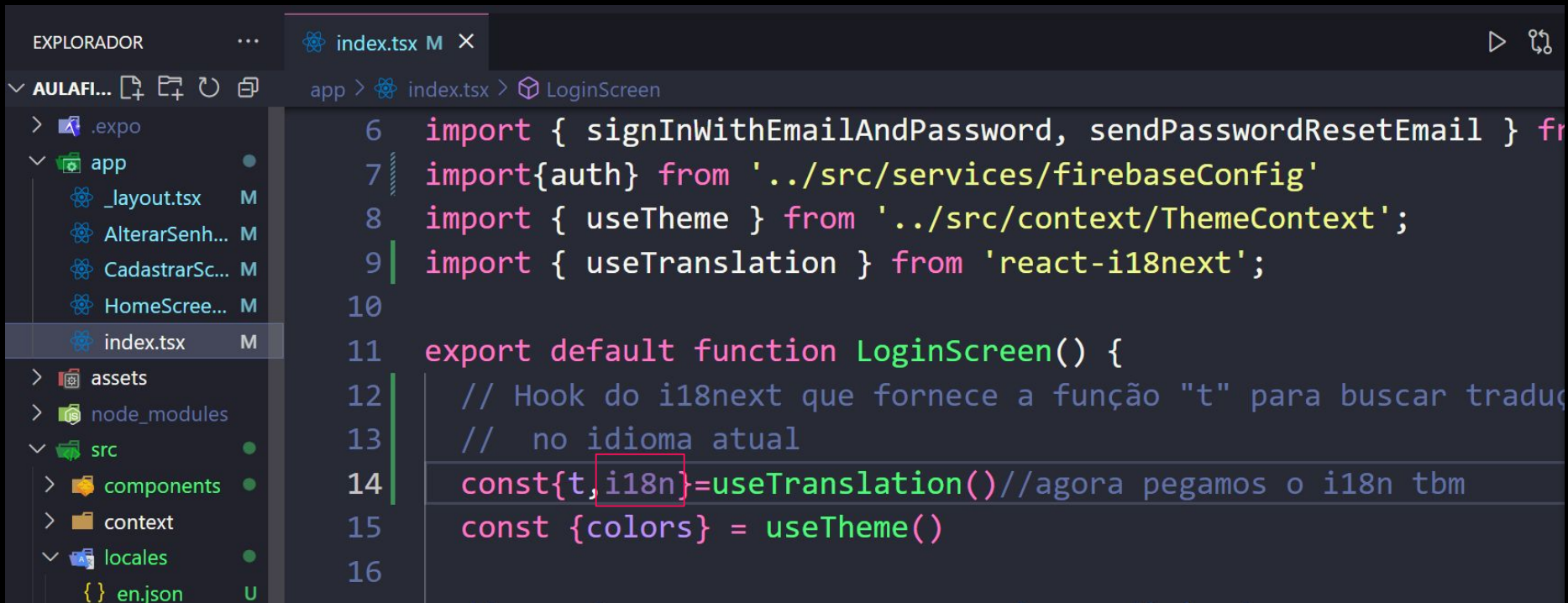
The image shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .expo, app, assets, node_modules, src, components, context, and locales. The code editor shows the index.tsx file, which contains a default function LoginScreen(). The function has a useState hook for esqueceuSenha, a catch block for error, and a return statement for a Text component. The Text component's children are wrapped in a t() function, which is highlighted with a red box. The t() function is used to translate the initial message "login".

```
11  t default function LoginScreen() {
63    esqueceuSenha = ()=>{
70    catch((error)=>{
73  )
74
75
76  n (
77    ew style=[styles.container,{backgroundColor:colors.background}]
78    Text style=[styles.titulo,{color:colors.text}]]>{t("login")}</7
79
80
```

Incluindo um botão para alterar o idioma 

Pegando o i18n

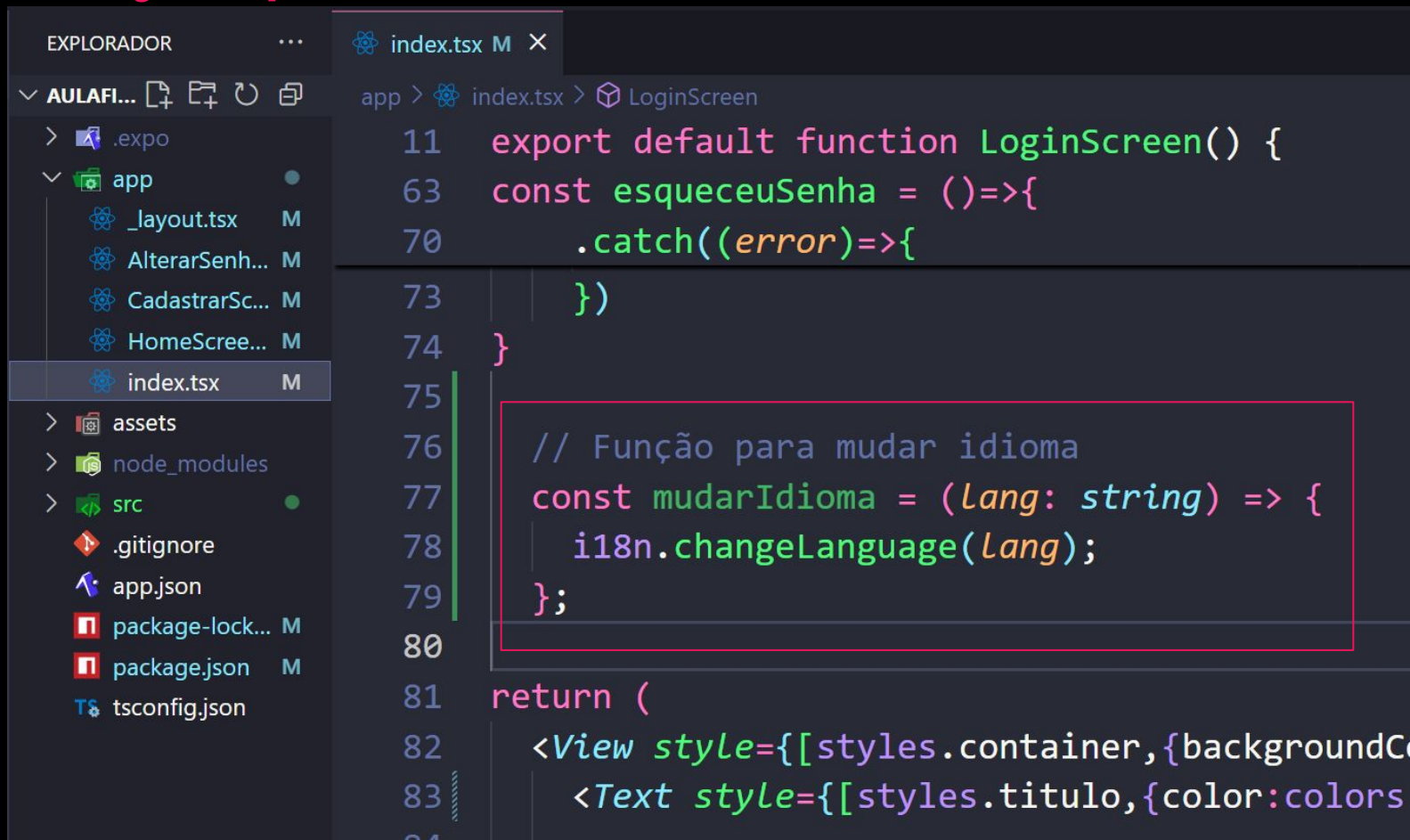
FLANP



```
EXPLORADOR  ...  index.tsx M X  ▶ 🔗  
v AULAFI...  [📁] [📁] [↺] [📄]  
  > .expo  
  v app  
    _layout.tsx M  
    AlterarSenh... M  
    CadastrarSc... M  
    HomeScrie... M  
    index.tsx M  
  > assets  
  > node_modules  
  v src  
    > components  
    > context  
    v locales  
      {} en.json U  
  app > index.tsx > LoginScreen  
6  import { signInWithEmailAndPassword, sendPasswordResetEmail } fr  
7  import { auth } from '../src/services/firebaseConfig'  
8  import { useTheme } from '../src/context/ThemeContext';  
9  import { useTranslation } from 'react-i18next';  
10  
11 export default function LoginScreen() {  
12   // Hook do i18next que fornece a função "t" para buscar tradu  
13   // no idioma atual  
14   const { t, i18n } = useTranslation() // agora pegamos o i18n tbm  
15   const { colors } = useTheme()  
16
```

Função para alterar o idioma

FIAP



```
EXPLORADOR  ...  index.tsx M X
AULA FI...  [Icons]
  > .expo
  > app
    _layout.tsx M
    AlterarSenh... M
    CadastrarSc... M
    HomeScree... M
    index.tsx M
  > assets
  > node_modules
  > src
    .gitignore
    app.json
    package-lock... M
    package.json M
    tsconfig.json

app > index.tsx > LoginScreen
11  export default function LoginScreen() {
63  const esqueceuSenha = ()=>{
70      .catch((error)=>{
73      })
74  }
75
76  // Função para mudar idioma
77  const mudarIdioma = (lang: string) => {
78      i18n.changeLanguage(lang);
79  };
80
81  return (
82      <View style={[styles.container,{backgroundC
83      <Text style={[styles.titulo,{color:colors
```


Criando o visual para os botões

FLANP

index.tsx M X

app > index.tsx > LoginScreen

```
11 export default function LoginScreen() {  
  
109     <View style={{ flexDirection: "row",justifyContent: "center",  
110         marginBottom: 15 }}>  
111         <TouchableOpacity  
112             style={[styles.botao, {  
113                 marginRight: 10, backgroundColor: "#007bff" }]}  
114             onPress={() => mudarIdioma("pt")}  
115         >  
116             <Text style={styles.textoBotao}>PT</Text>  
117         </TouchableOpacity>  
118  
119         <TouchableOpacity  
120             style={[styles.botao, { backgroundColor: "#28a745" }]}  
121             onPress={() => mudarIdioma("en")}  
122         >  
123             <Text style={styles.textoBotao}>EN</Text>  
}
```

Criando o visual para os botões

index.tsx M X

app > index.tsx > LoginScreen

```
11  export default function LoginScreen() {  
117    </TouchableOpacity>  
118  
119    <TouchableOpacity  
120      style={[styles.botao, { backgroundColor: "#28a745" }]}  
121      onPress={() => mudarIdioma("en")}  
122    >  
123      <Text style={styles.textoBotao}>EN</Text>  
124    </TouchableOpacity>  
125  </View>  
126  
127  {/* Botão */}  
128  
129  <TouchableOpacity style={styles.botao} onPress={handleLogin}>  
130    <Text style={styles.textoBotao}>Login</Text>  
131  </TouchableOpacity>
```

Estilo dos botões

index.tsx M X

app > index.tsx > LoginScreen

```
146 const styles = StyleSheet.create({
159   input: {
168   },
169   botao: {
170     backgroundColor: '■ #00B37E',
171     padding: 15,
172     borderRadius: 10,
173     alignItems: 'center',
174   },
175   textoBotao: {
176     color: '■ #fff',
177     fontSize: 18,
178     fontWeight: 'bold',
179   },
180 });
181
```


Dúvidas?