

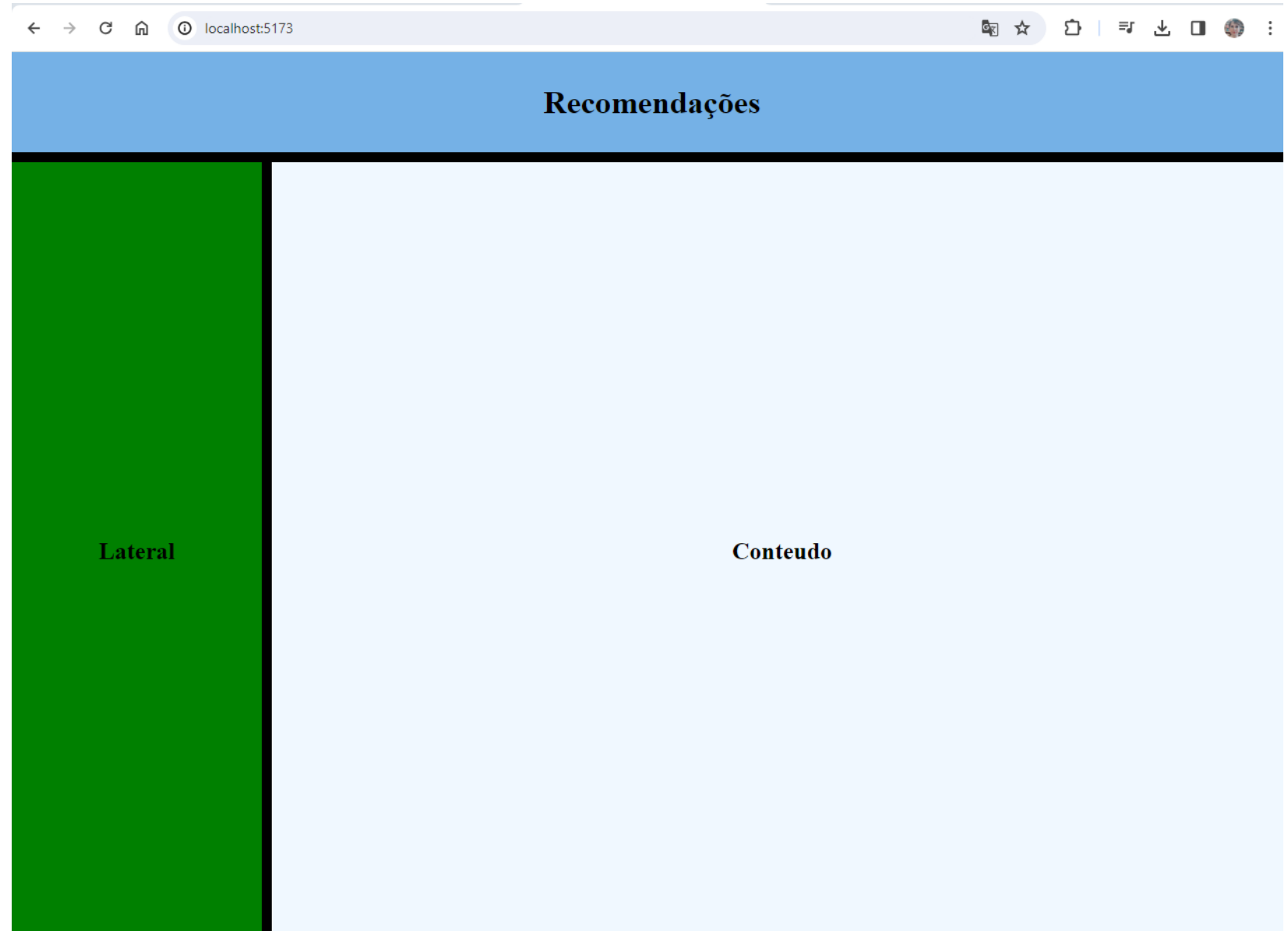
FrontEnd

FrontEnd - React

Fernanda Fretes

Nossa aplicação
ainda está com
essa carinha.

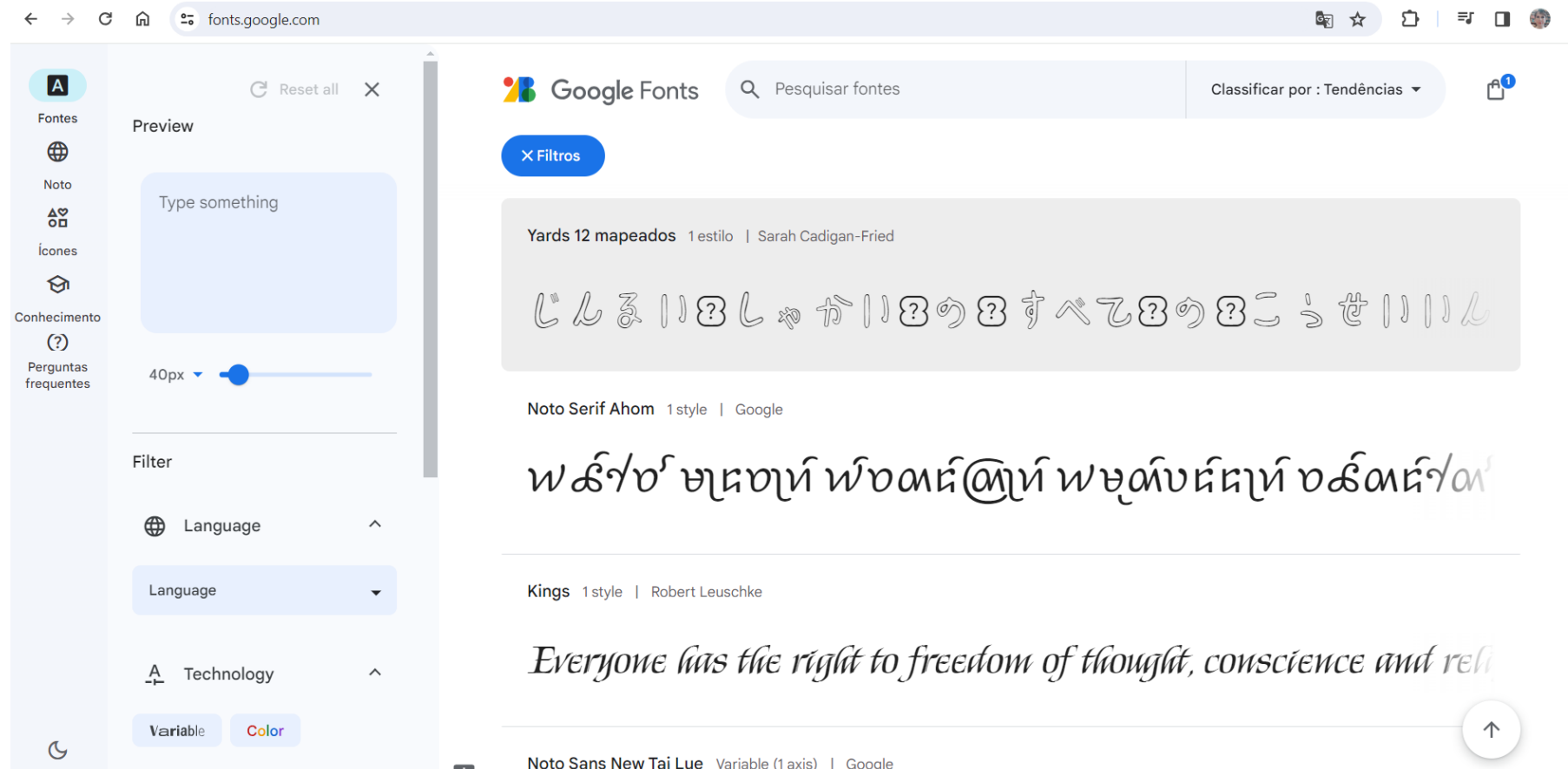
Bora melhorar?



FrontEnd - React

Através do google fonts podemos encontrar várias fontes disponíveis para a importação.

<https://fonts.google.com/>



FrontEnd - React

Selecione uma delas, devemos selecionar “Obter fonte”



FrontEnd - React

Escolhida a fonte que deseja usar, podemos usar no nosso projeto copiando o que está no link dela



← Código embutido

Reis Estático

Whereas recognition of the inhe

Regular 400



1 estilo

400 normais

Roteiro de Dança Variável ⓘ

Rede

Android

iOS

Vibração

☒ <ligação> ☐ @importar

Incorpore o código no <head> do seu html

```
<link rel="preconnect" href="https://fonts.googleapis.com"> <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin> <link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@400..700&family=Kings&display=swap" rel="stylesheet">
```

Copiar código

Reis: classe CSS

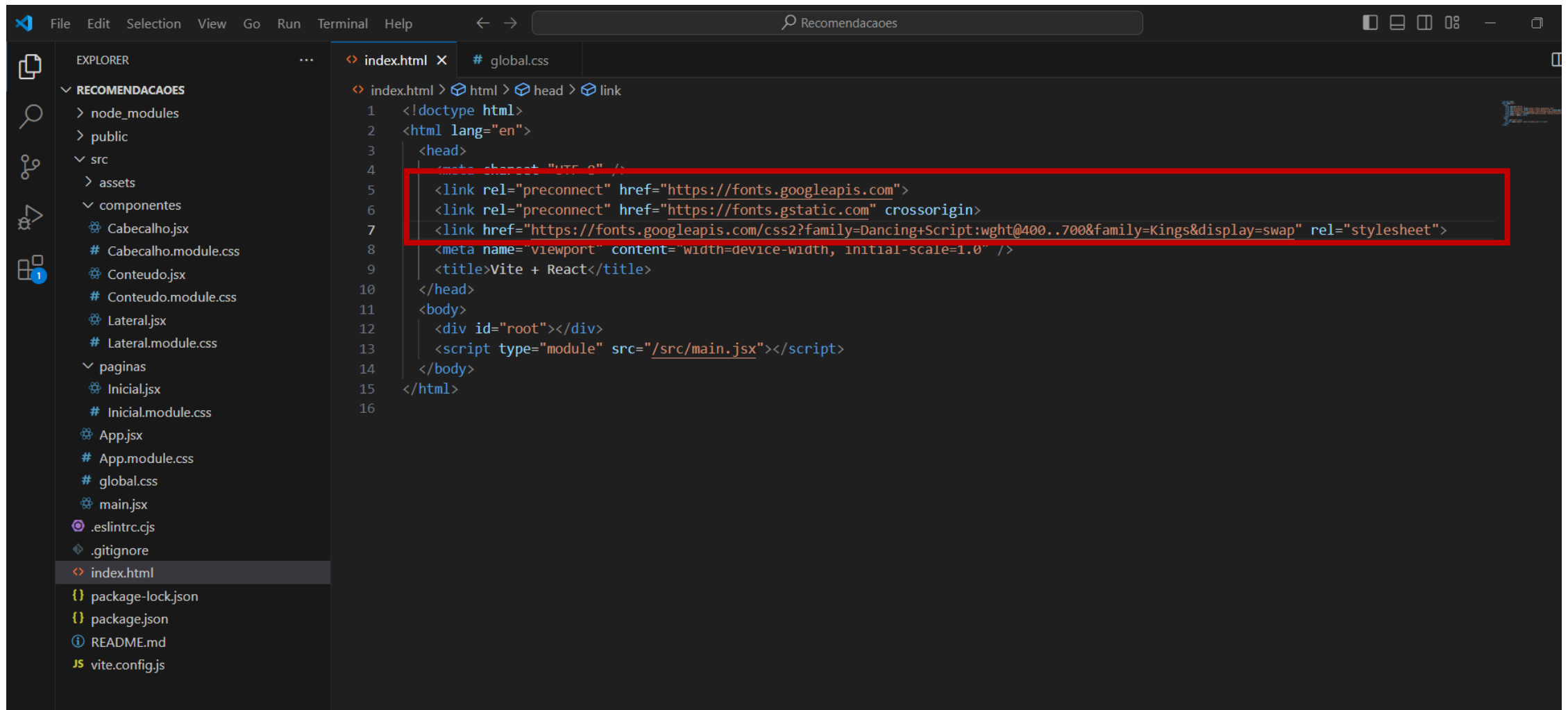
```
.kings-regular { font-family: "Kings", cursivo; peso da fonte: 400; estilo de fonte: normal; }
```

Copiar código

Dancing Script: classe CSS para um estilo variável

FrontEnd - React

Na nossa página index, colocamos o link copiado



The screenshot shows the VS Code editor interface. On the left, the Explorer sidebar displays a file tree with folders like 'RECOMENDACOES', 'node_modules', 'public', 'src', and 'assets'. The 'index.html' file is selected. The main editor area shows the content of 'index.html'. A red rectangular box highlights the following lines of code:

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="preconnect" href="https://fonts.googleapis.com">
6     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
7     <link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@400..700&family=Kings&display=swap" rel="stylesheet">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9     <title>Vite + React</title>
10  </head>
11  <body>
12    <div id="root"></div>
13    <script type="module" src="/src/main.jsx"></script>
14  </body>
15 </html>
16
```

FrontEnd - React

```
global.css > .material-symbols-outlined
:root{
  --cinza-claro: #dfd9df;
  --roxo-escuro: #1f0322;
  --roxo-claro: #ab10bc;
}

*{
  margin:0px;
  padding: 0px;
  box-sizing: border-box;
}
body{
  background-color: var(--roxo-escuro);
}

body, input{
  font-family: "Dancing Script", cursive;
}

.material-symbols-outlined {
```

Importando a fonte, podemos então usa-la em algum componente que achemos interessante, como por exemplo no corpo da aplicação.

Então, no arquivo global.css vou indica-la no body e input

FrontEnd - React

```
C:\React\Recomendacoes>npm run dev
```

```
> recomendacoes@0.0.0 dev  
> vite
```

```
VITE v5.1.6 ready in 577 ms
```

```
→ Local: http://localhost:5173/
```

```
→ Network: use --host to expose
```

```
→ press h + enter to show help
```

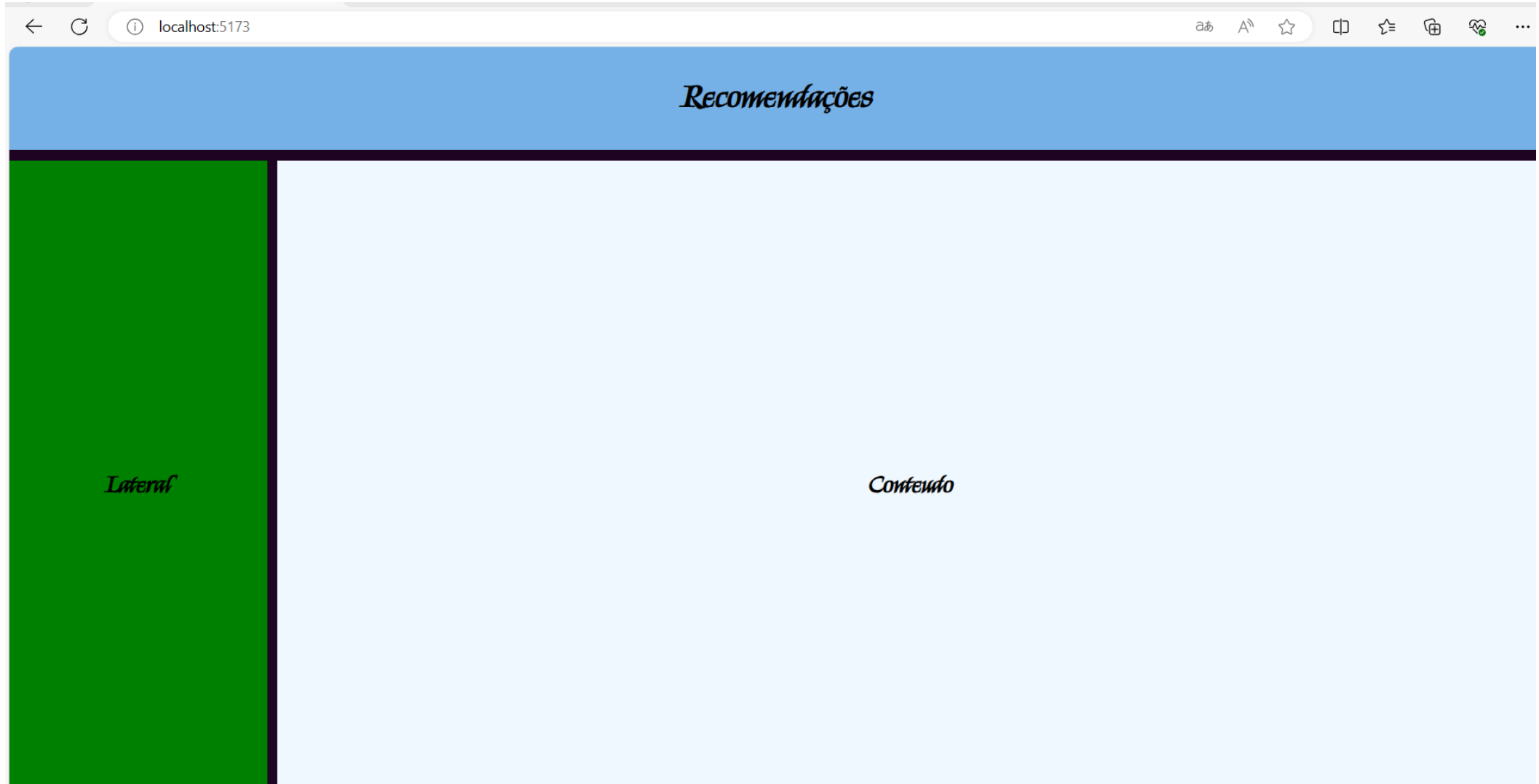
```
13:04:04 [vite] page reload index.html
```

```
13:04:04 [vite] page reload index.html (x2)
```

Para vê-la funcionando, basta que eu “ligue” o servidor, pelo prompt

FrontEnd - React

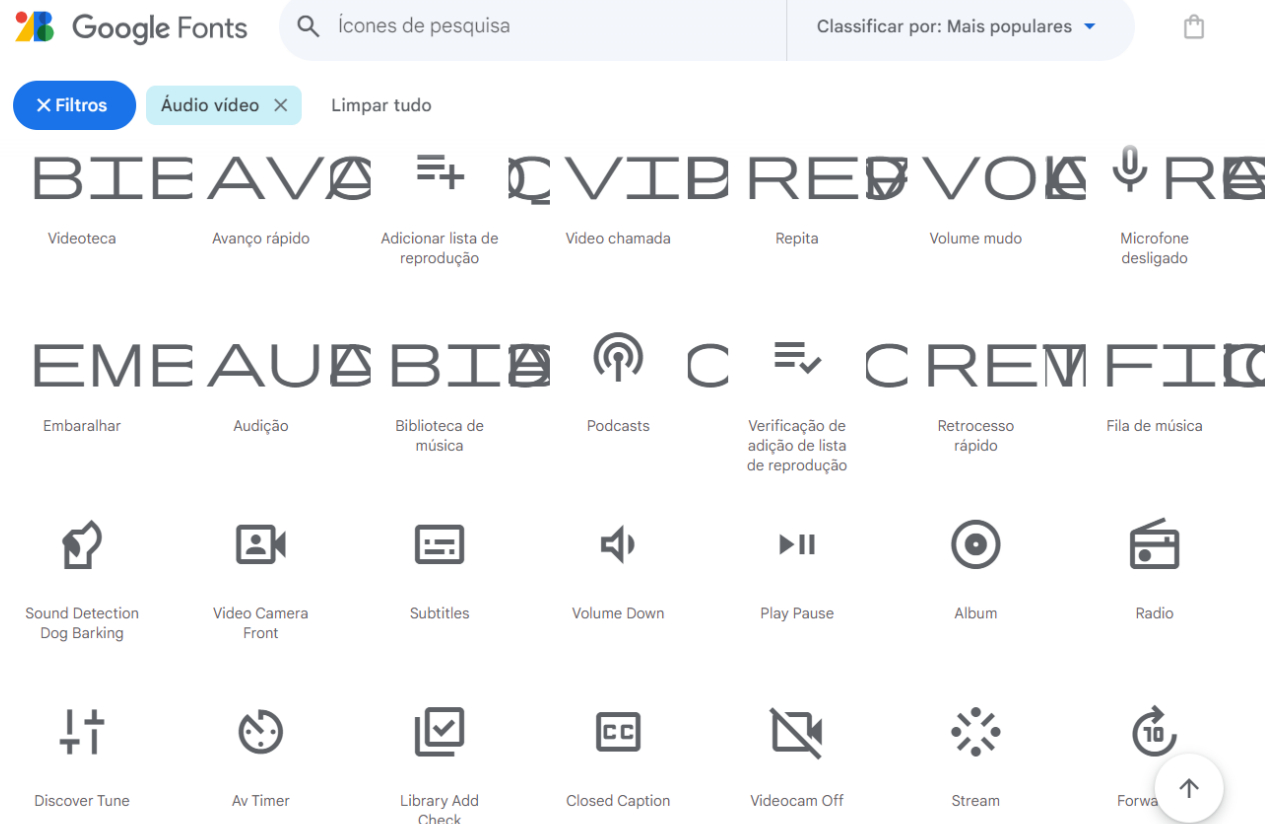
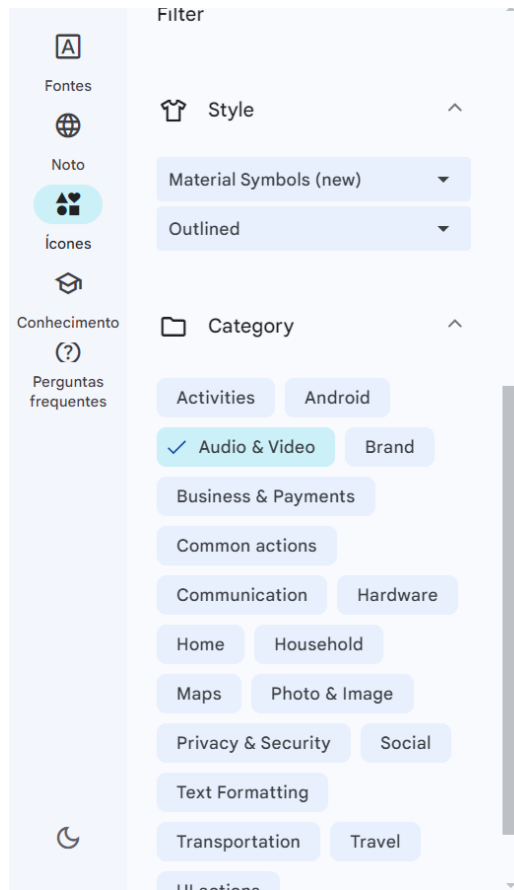
Fernanda Fretes



FrontEnd - React

O mesmo acontece se quisermos usar alguma ícone.

Posso através dos links disponibilizados usá-los na minha aplicação



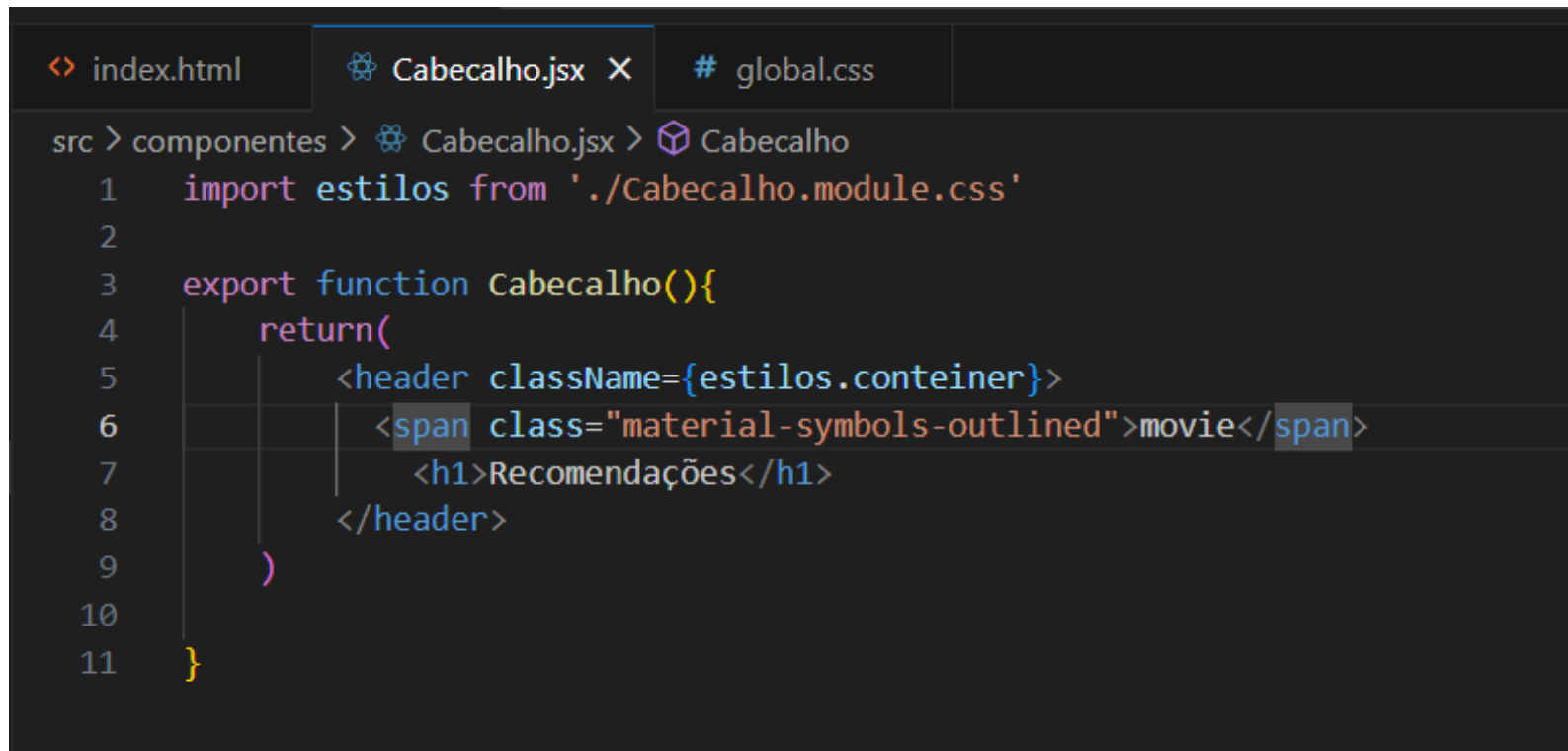
FrontEnd - React

Em index, devo colocar o link correspondente

```
index.html > html > head > link
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="preconnect" href="https://fonts.googleapis.com">
6     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
7     <link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@400..700&display=swap" rel="stylesheet">
8     <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@20..48,100..700,
9
10    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
11    <title>Vite + React</title>
12  </head>
13  <body>
14    <div id="root"></div>
15    <script type="module" src="/src/main.jsx"></script>
16  </body>
17 </html>
18
```

FrontEnd - React

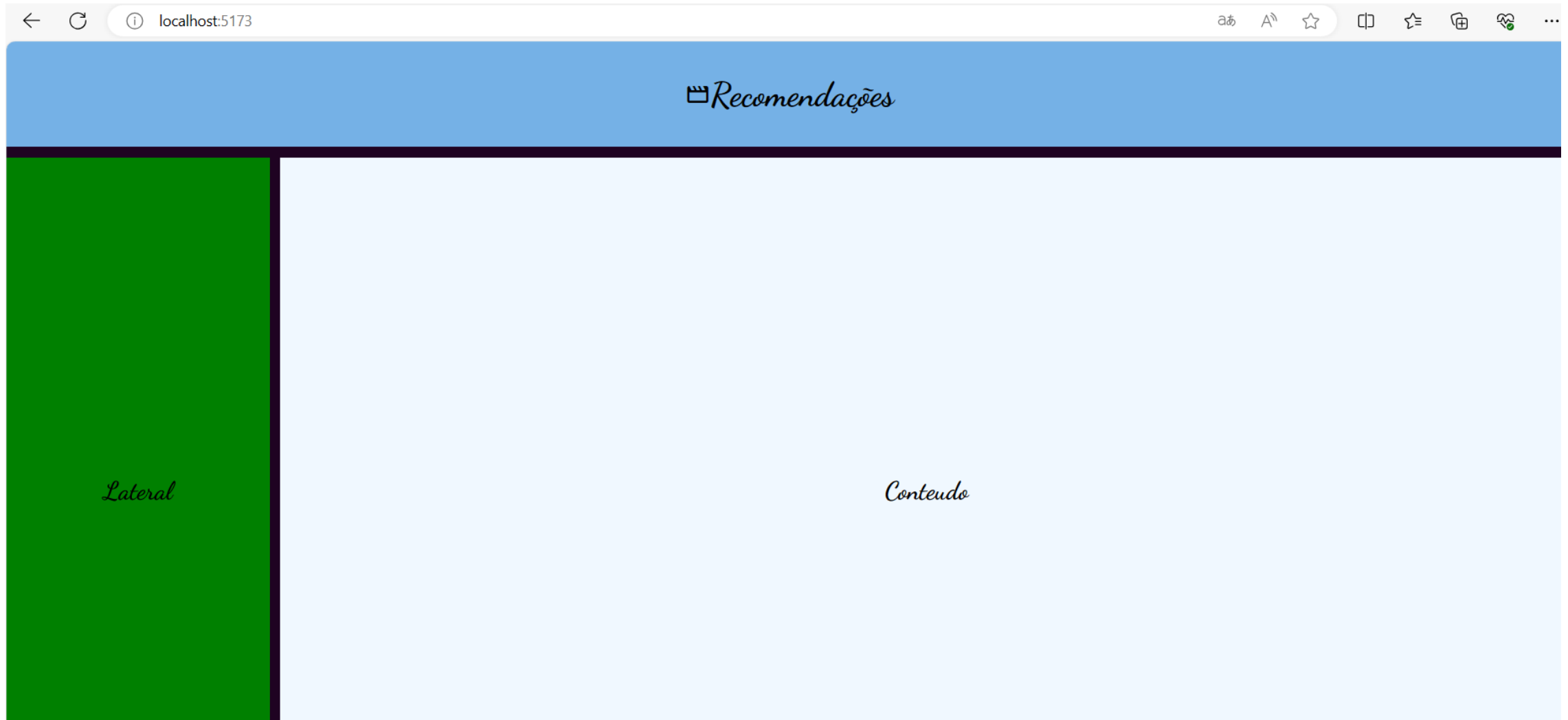
Em e coloco o ícone no componente desejado, como no cabeçalho, por exemplo.



```
src > componentes > Cabecalho.jsx > Cabecalho
1  import estilos from './Cabecalho.module.css'
2
3  export function Cabecalho(){
4    return(
5      <header className={estilos.container}>
6        <span class="material-symbols-outlined">movie</span>
7        <h1>Recomendações</h1>
8      </header>
9    )
10
11 }
```

FrontEnd - React

Fernanda Fretes



FrontEnd - React

Mas para deixar tudo mais harmonioso, podemos usar as cores que deixamos nas variáveis.

```
index.html Cabecalho.jsx Lateral.jsx ... # Lateral.module.css # global.css # Cabecalho.module.css X
src > componentes > Cabecalho.jsx > Cabecalho
1 import estilos from './Cabecalho.module.css'
2
3 export function Cabecalho(){
4   return(
5     <header className={estilos.container}>
6       <span class="material-symbols-outlined">movie</span>
7       <h1>Recomendações</h1>
8     </header>
9   )
10 }
11
12

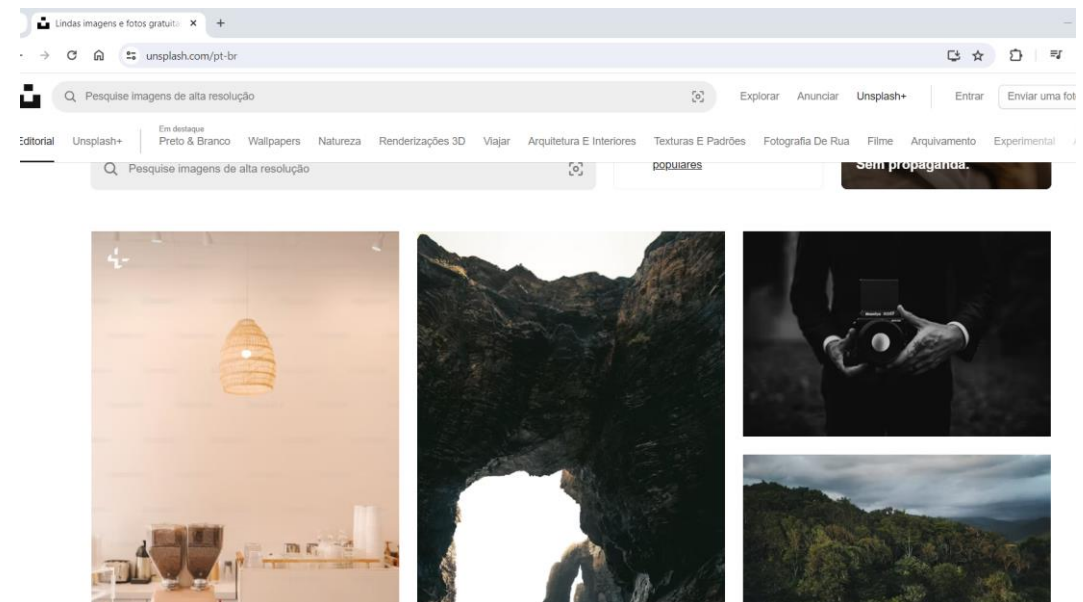
src > componentes > # Cabecalho.module.css > span
1 .container{
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   background-color: var(--roxo-escuro);
6   font-family: "Dancing Script", cursive;
7 }
8 h1{
9   color: var(--cinza-claro);
10 }
11
12 span{
13   color: var(--cinza-claro);
14 }
15
```

Recomendações

FrontEnd - React

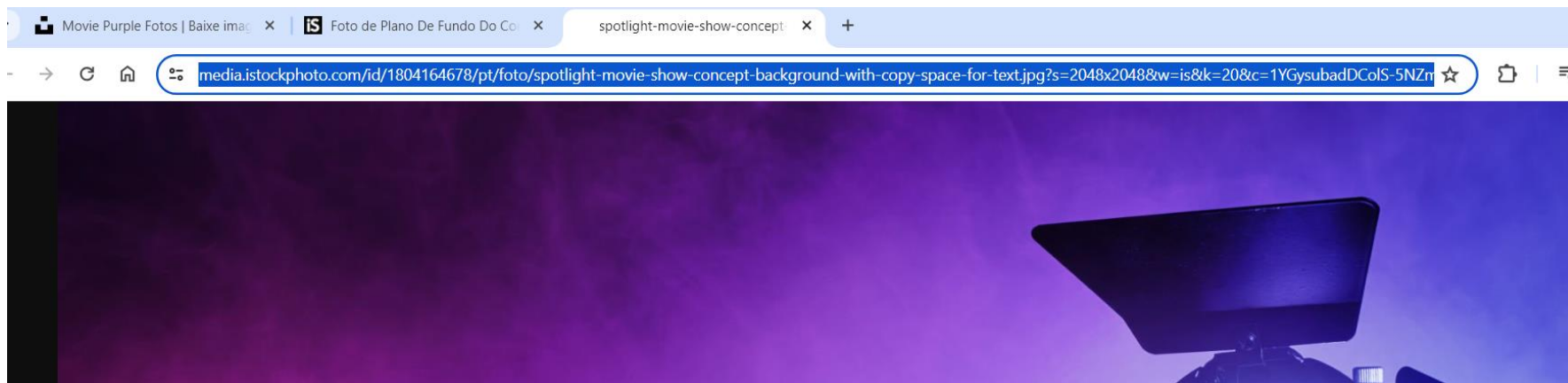
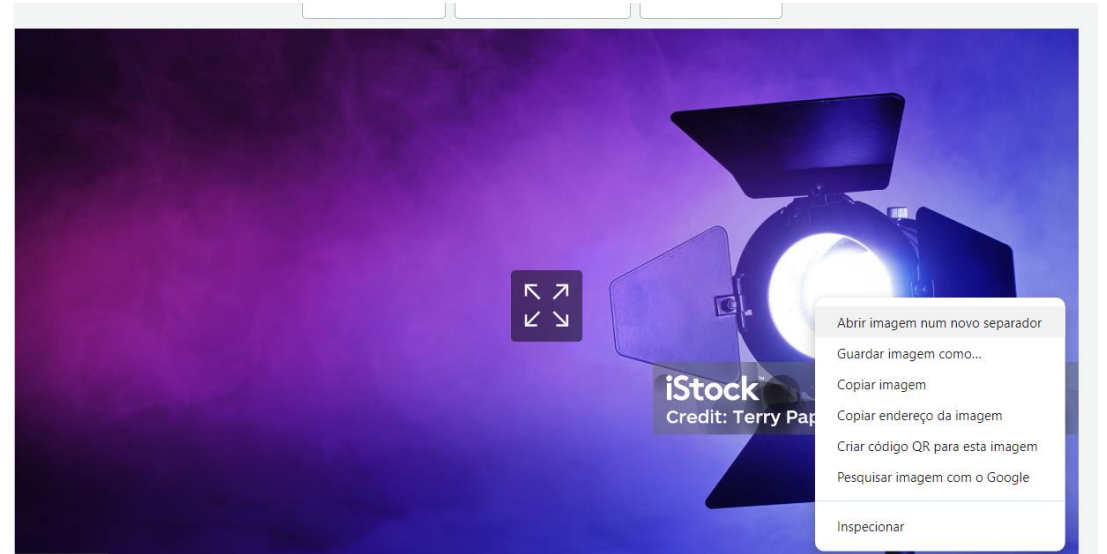
Além de fontes e ícones, também conseguimos usar imagens que estejam disponíveis na internet. Isso é uma ferramenta poderosa, mas temos que ter o cuidado pois, se não é de nossa autoria existe o risco dela ficar inacessível em algum momento.

No site <https://unsplash.com/pt-br>, temos muitas imagens de alta resolução, que podem ser gratuitas, outras já são pagas.



FrontEnd - React

Selecionando alguma imagem, podemos abrir em uma nova guia e copiar esse endereço.



FrontEnd - React

Esse endereço posso colocar na tag `` em algum componente, como por exemplo no da lateral .

```
export function Lateral(){  
  return(  
    <aside className={estilos.container}>  
      <img  
        className={estilos.imgLateral}  
        src = "https://media.istockphoto.com/id/1804164678/pt/foto/spotlight-movie-show-concept-background-with-copy-space-for-t
```

FrontEnd - React

Esse endereço posso colocar na tag `` em algum componente, como por exemplo no da lateral .

```
export function Lateral(){  
  return(  
    <aside className={estilos.container}>  
      <img  
        className={estilos.imgLateral}  
        src = "https://media.istockphoto.com/id/1804164678/pt/foto/spotlight-movie-show-concept-background-with-copy-space-for-t
```

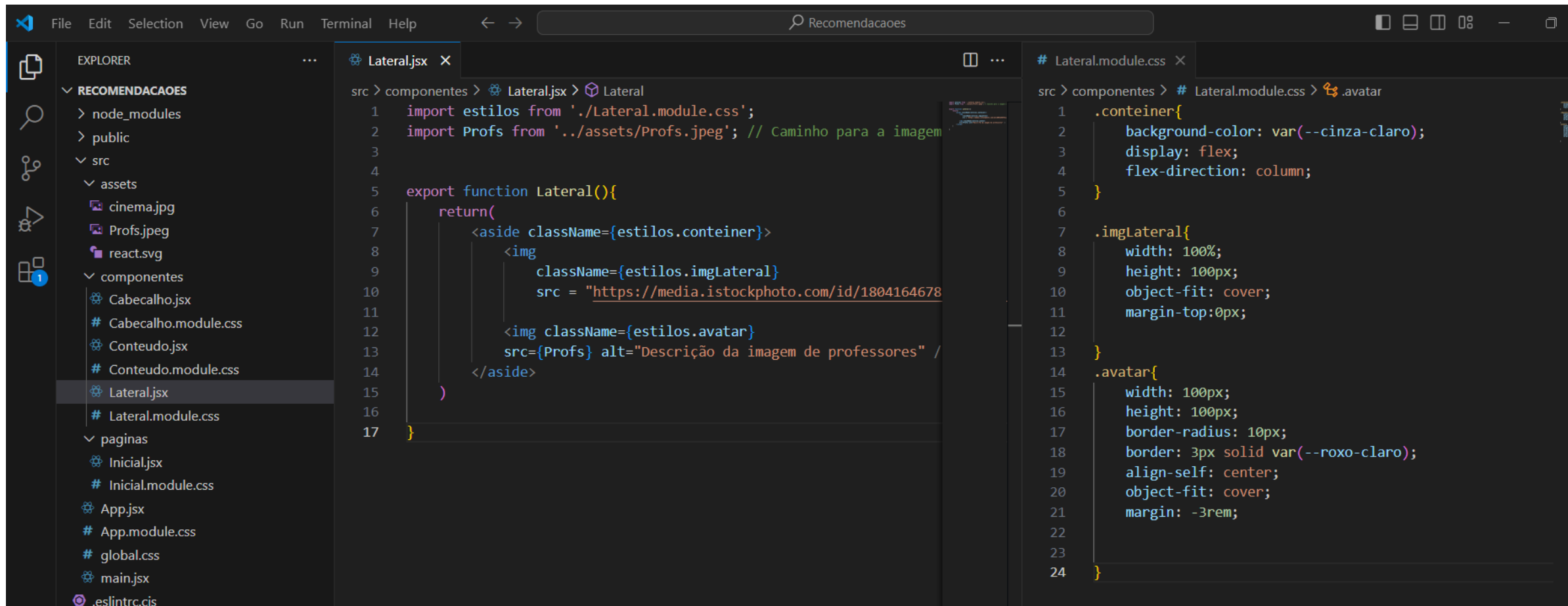
FrontEnd - React

Caso eu deseje usar uma imagem local, ou seja, uma que está armazenada no computador (servidor) , eu devo fazer a importação dela e depois chama-la com a tag em algum componente, como por exemplo no da lateral .

```
index.html  Lateral.jsx  X
src > componentes > Lateral.jsx > Lateral
1  import estilos from './Lateral.module.css';
2  import Profs from '../assets/Profs.jpeg'; // Caminho para a imagem na pasta de assets
3
4
5  export function Lateral(){
6    return(
7      <aside className={estilos.container}>
8        <img
9          className={estilos.imgLateral}
10         src = "https://media.istockphoto.com/id/1804164678/pt/foto/spotlight-movie-show-concept-back
11
12         <img className={estilos.avatar}
13         src={Profs} alt="Descrição da imagem de professores" />
14       </aside>
15     )
16
17  }
```

FrontEnd - React

Fernanda Fretes



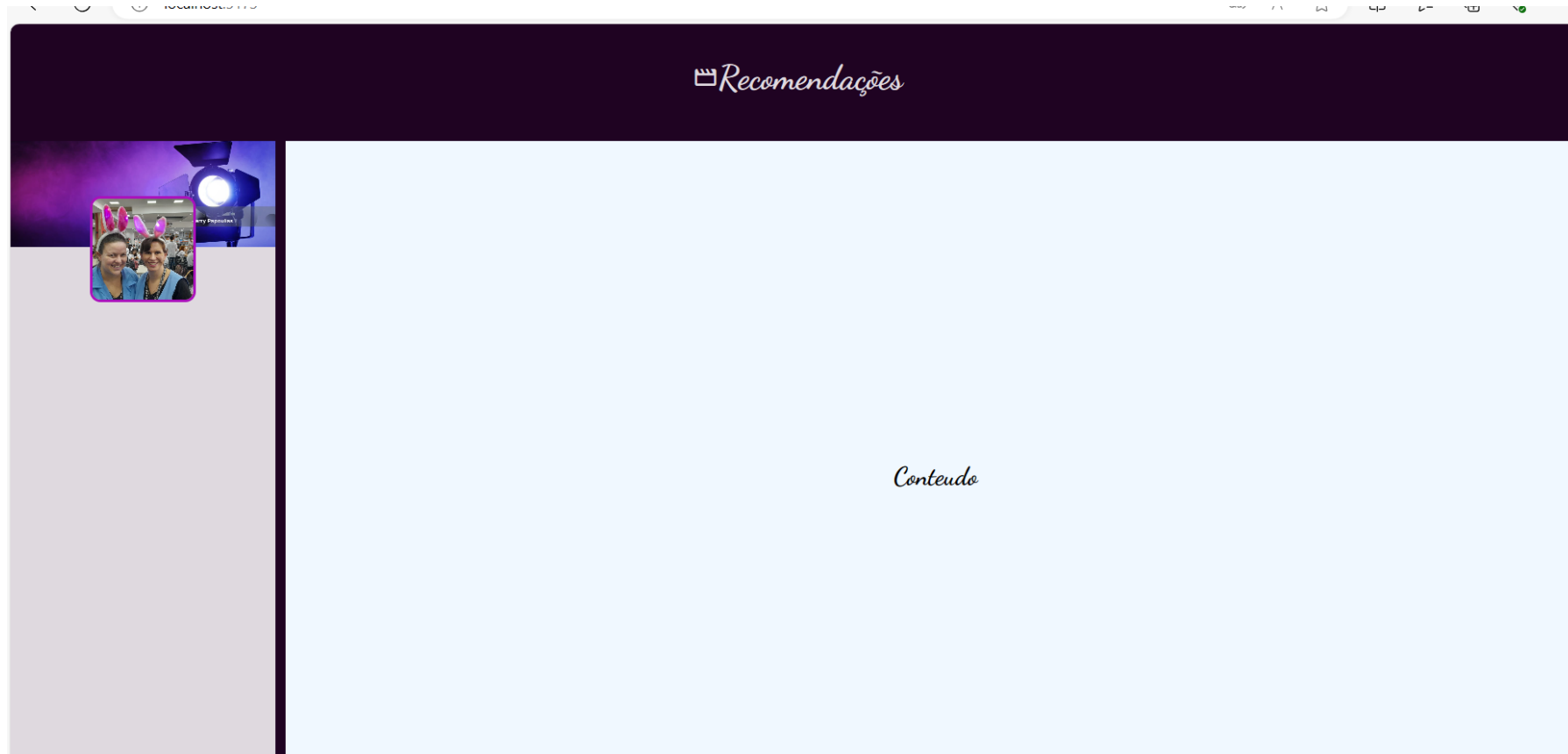
The screenshot shows a VS Code editor with a dark theme. The Explorer panel on the left displays a project structure with folders like 'RECOMENDACOES', 'src', 'assets', and 'componentes'. The 'componentes' folder is expanded, showing files like 'Lateral.jsx' and 'Lateral.module.css'. The main editor area shows the code for 'Lateral.jsx' and 'Lateral.module.css'.

```
src > componentes > Lateral.jsx > Lateral
1  import estilos from './Lateral.module.css';
2  import Profs from '../assets/Profs.jpeg'; // Caminho para a imagem
3
4
5  export function Lateral(){
6    return(
7      <aside className={estilos.container}>
8        <img
9          className={estilos.imgLateral}
10         src = "https://media.istockphoto.com/id/1804164678
11
12         <img className={estilos.avatar}
13         src={Profs} alt="Descrição da imagem de professores" /
14       </aside>
15     )
16   }
17 }
```

```
# Lateral.module.css > .avatar
1  .container{
2    background-color: var(--cinza-claro);
3    display: flex;
4    flex-direction: column;
5  }
6
7  .imgLateral{
8    width: 100%;
9    height: 100px;
10   object-fit: cover;
11   margin-top: 0px;
12 }
13
14 .avatar{
15   width: 100px;
16   height: 100px;
17   border-radius: 10px;
18   border: 3px solid var(--roxo-claro);
19   align-self: center;
20   object-fit: cover;
21   margin: -3rem;
22 }
23
24 }
```

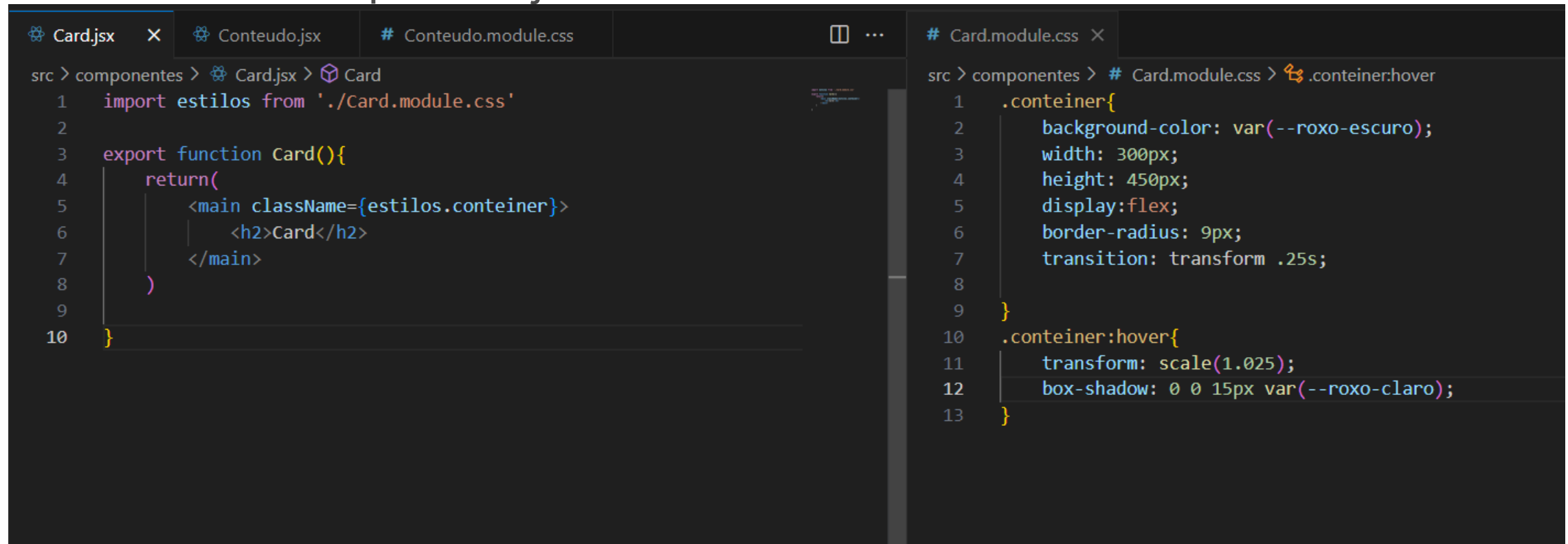
FrontEnd - React

Caso eu deseje usar uma imagem local, ou seja, uma que está armazenada no computador (servidor) , eu devo fazer a importação dela e depois chama-la com a tag `` em algum componente, como por exemplo no da lateral .



FrontEnd - React

O conteúdo, é o único componente que ainda não mexemos, então, vamos começar a popular com algumas informações. Vamos criar mais dois arquivos em componentes. O “Card.jsx” e o “Card.module.css”. É Neste card que vamos colocar os filmes que desejamos ver.



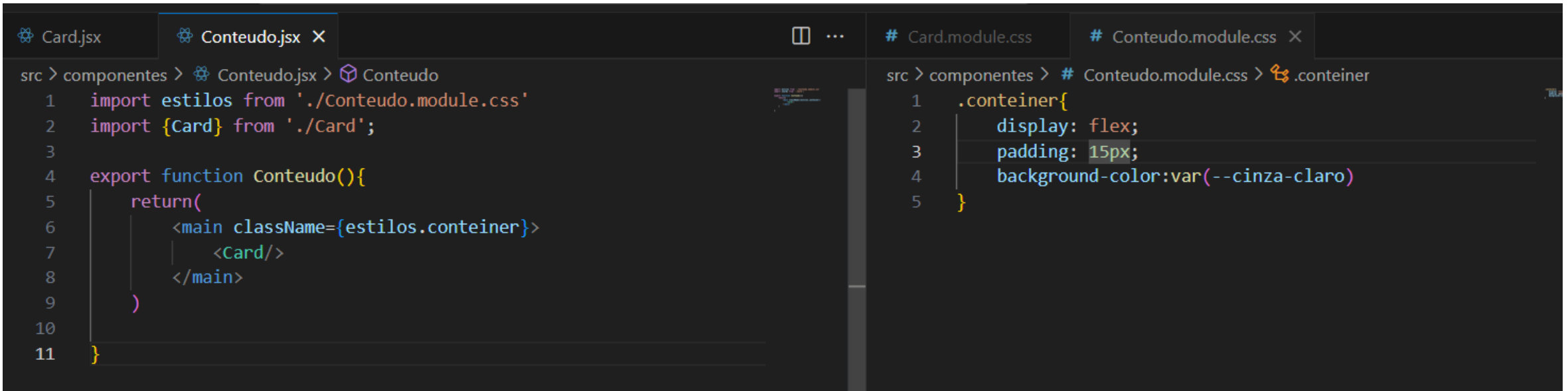
The screenshot shows a code editor with two files open: Card.jsx and Card.module.css. The Card.jsx file is on the left, and Card.module.css is on the right. The Card.jsx file contains a function Card() that returns a JSX element. The Card.module.css file contains CSS rules for the container and its hover state.

```
src > componentes > Card.jsx > Card
1  import estilos from './Card.module.css'
2
3  export function Card(){
4    return(
5      <main className={estilos.container}>
6        <h2>Card</h2>
7      </main>
8    )
9
10 }
```

```
src > componentes > # Card.module.css > .container: hover
1  .container{
2    background-color: var(--roxo-escuro);
3    width: 300px;
4    height: 450px;
5    display: flex;
6    border-radius: 9px;
7    transition: transform .25s;
8
9  }
10 .container: hover{
11   transform: scale(1.025);
12   box-shadow: 0 0 15px var(--roxo-claro);
13 }
```

FrontEnd - React

E para ficar melhor alinhado e ter um uso para esse componente, vamos fazer algumas mudanças no componente conteúdo e no seu CSS.



The screenshot shows a code editor with two files open: `Card.jsx` and `Conteudo.jsx`. The `Conteudo.jsx` file is active, showing the following code:

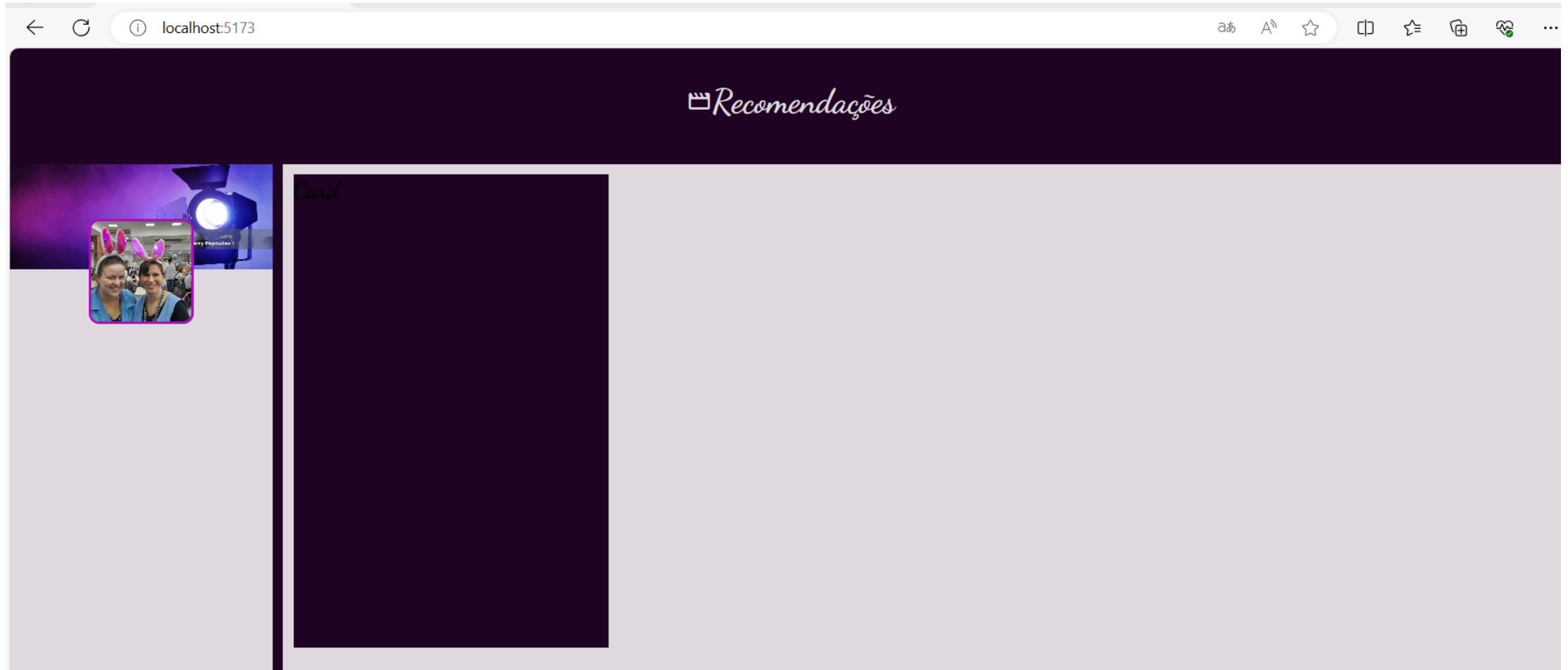
```
src > componentes > Conteudo.jsx > Conteudo
1  import estilos from './Conteudo.module.css'
2  import {Card} from './Card';
3
4  export function Conteudo(){
5    return(
6      <main className={estilos.container}>
7        <Card/>
8      </main>
9    )
10
11 }
```

The `Conteudo.module.css` file is also open, showing the following CSS code:

```
src > componentes > Conteudo.module.css > .container
1  .container{
2    display: flex;
3    padding: 15px;
4    background-color:var(--cinza-claro)
5  }
```

FrontEnd - React

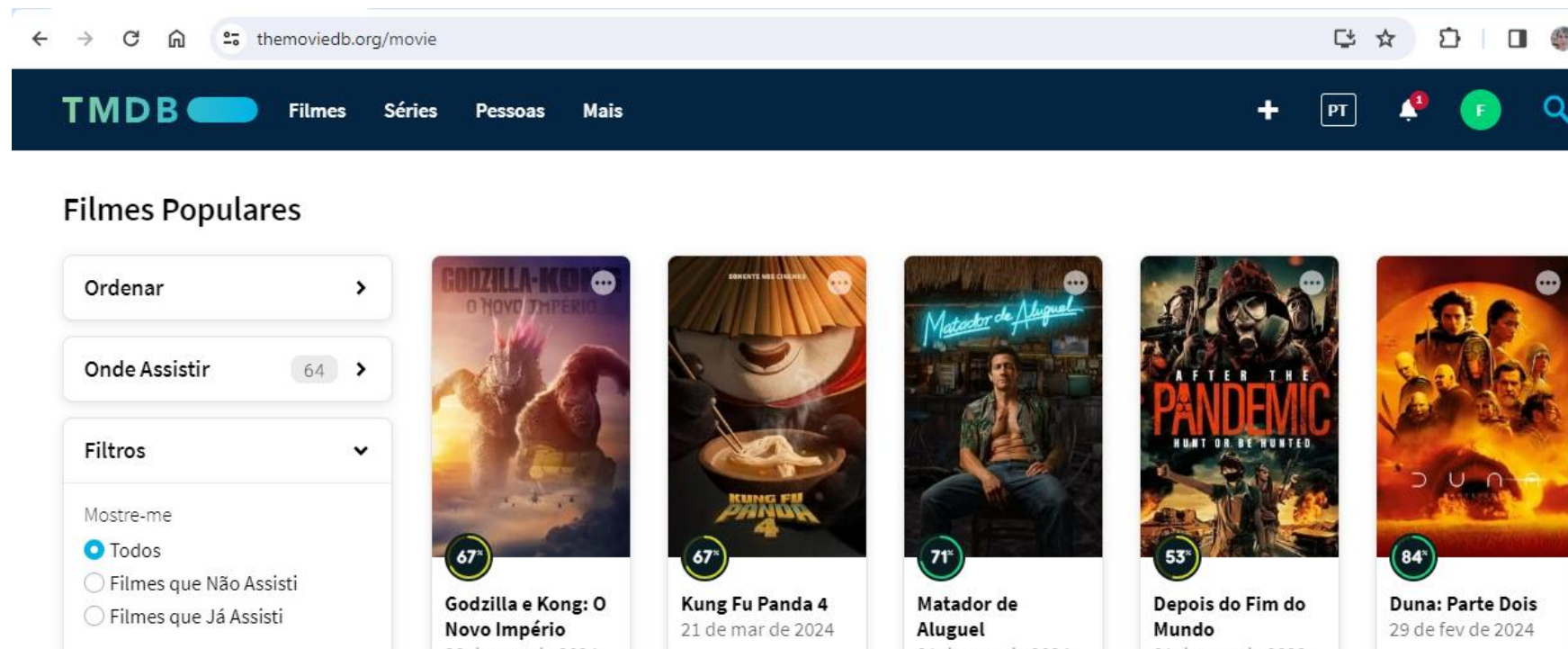
A aplicação ficará com essa apresentação fazendo esses comandos.



FrontEnd - React

A base inicial da nossa aplicação já está pronta, mas é interessante que consigamos colocar no “card” filmes de um catálogo. E para isso vamos usar uma API.

O site The movie DB disponibiliza uma API para consumo quando o uso é acadêmico. Para isso é necessário fazer um pequeno cadastro, para que seja enviado uma chave no e-mail.



FrontEnd - React

Uma **API** (Application Programming Interface – Interface de Programação de Aplicações) é um conjunto de códigos, regras, protocolos e opções que possibilitam a comunicação entre diferentes aplicações.

As APIs podem ser usadas para acessar recursos e funcionalidades de sistemas ou serviços de software, como sistemas operacionais, bancos de dados, serviços web, aplicativos e outros.

Através das APIs, podemos acessar esse conjunto de códigos que nos fornecem diversas funcionalidades de programação, sem a necessidade de nos preocuparmos com sua construção e funcionamento detalhados.

FrontEnd - React

Para que usemos essa API, precisamos inicialmente de um endereço, e de uma chave.

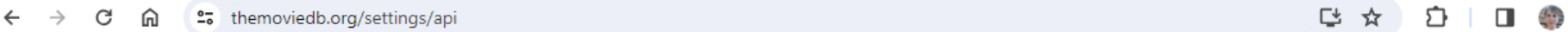
Inicialmente nosso endereço será:

<https://api.themoviedb.org/3>

A chave será enviado por email após o cadastro, e também pode ser consultado na própria plataforma:

<https://www.themoviedb.org/settings/api>

Fernanda Fretes



TMDB

Filmes

Séries

Pessoas

Mais

+

PT



F

Q

F

fernanda.fretes

Configurações

Perfil

Conta

Serviços de Streaming

Notificações

Usuários bloqueados

Importar listas

Compartilhamento

Sessões

API

Excluir conta

API

Visão geral

Detalhes

Sessões

Estatísticas

Chave Regenerada

TMDb offers a powerful API service that is free to use as long as you properly attribute us as the source of the data and/or images you use. You can find the logos for attribution [here](#).

Documentação

Our primary documentation is located at developer.themoviedb.org.

Suporte

If you have questions or comments about the information covered here, please create a post on our [support forums](#).

Detalhes da API

If you'd like to edit the details of your app, [click here](#).

Chave da API

af26cce282aecf5c6cc39a264f29d0a7

Token de Leitura da API

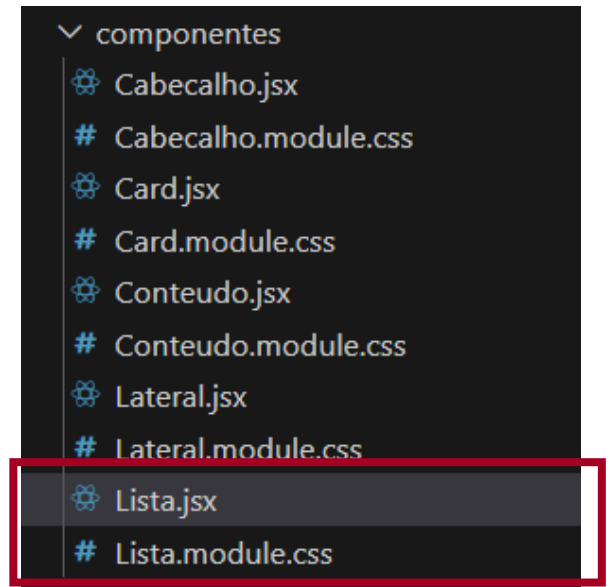
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJhZjI2Y2NlMgyYWVjZjVjNmNmJmZlhkMyJ0ZjI5ZDBhNyIsbnN1Yil6IjY2MTI2MjdhdnMmY0M2VmDE3ZGMzY2Y2OTQwLWY3KSrNecPNIHwArOtGiuX9fOtrDyKXo

FrontEnd - React

Vamos precisar instalar o axios também, através do CMD com o comando:

```
npm install axios
```

E não desejamos administrar manualmente os card para exibir a lista de filmes, por isso, podemos criar mais um componente, chamado lista, onde para cada retorno da API seja renderizado um card para aquele ID.



FrontEnd - React

```
import React, { useState, useEffect } from 'react';
```

useState: é usado para adicionar estado aos componentes funcionais. Lembra daquela página inicial do React que tinha um contador? É graças a ele que ele consegue ser atualizado sem que seja necessário dar um refresh.

```
const [contador, setContador] = useState(0);
```

```
<button onClick={() => setContador(contador + 1)}>Incrementar</button>
```

O estado é mantido pelo useState

Para usá-lo, devemos chama-lo no início do componente.

Em resumo é uma maneira simples de adicionar estado ao componentes permitindo interatividade.

FrontEnd - React

O **useEffects**, são efeitos que ocorrem após cada renderização, como buscar alguma informação de uma API.

```
import React, { useState, useEffect } from 'react';
```

```
import axios from 'axios';
```

O Axios é usado para fazer as solicitações HTTP para a API do The MovieDB.

Como queremos que sejam exibidas as respostas das solicitações em uma lista de Cards, então devo fazer o import desse componente.

```
import { Card } from './Card';
```

FrontEnd - React

Credenciais de acesso a API

Devemos também declarar variáveis para fazer a comunicação com a API

```
const API_KEY = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx';
```

A chave gerada na aplicação

```
const API_URL = 'https://api.themoviedb.org/3';
```

E o endereço onde conseguimos resgatar os valores desejados

FrontEnd - React

Feito isso os importes e credenciais necessárias, então, agora podemos abrir nosso componente:

```
export function Lista() {  
  const [movies, setMovies] = useState([]);
```

Estamos inicializando o estado movies como um array vazio até que seja atualizado através da função setMovies.

Essa abordagem é comum ao trabalhar com listas de dados em React. O estado inicial vazio permite que você preencha os dados da lista mais tarde, por exemplo, após a busca de dados de uma API, como está sendo feito com a chamada axios.get no useEffect.

FrontEnd - React

Hooks

Em resumo, um Hook em React é uma função que permite que você use recursos do React em componentes funcionais, tornando-os mais poderosos e flexíveis. Eles são uma parte essencial da programação React moderna e ajudam a simplificar o código e melhorar a reutilização de lógica entre componentes.

Os Hooks têm algumas regras de uso:

Eles devem ser chamados apenas no nível superior de um componente ou dentro de outros Hooks personalizados.

Eles não devem ser chamados dentro de loops, condições ou funções aninhadas.

O nome dos Hooks deve começar com use

FrontEnd - React

O `useEffect` é outro Hook fornecido pelo React que permite executar efeitos colaterais em componentes funcionais. Efeitos colaterais são operações que ocorrem após a renderização do componente, como buscar dados de uma API, manipular o DOM, ou executar subscrições a eventos.

É chamado após cada renderização do componente, incluindo a renderização inicial e todas as renderizações subsequentes.

FrontEnd - React

O `useEffect` é outro Hook fornecido pelo React que permite executar efeitos colaterais em componentes funcionais. Efeitos colaterais são operações que ocorrem após a renderização do componente, como buscar dados de uma API, manipular o DOM, ou executar subscrições a eventos.

É chamado após cada renderização do componente, incluindo a renderização inicial e todas as renderizações subsequentes.

```
useEffect(() => {  
  axios.get(`${API_URL}/movie/popular?api_key=${API_KEY}&language=pt-BR`)
```

FrontEnd - React

Quando os dados são recebidos com sucesso, armazenamos os resultados no estado movies usando a função setMovies. Se houver algum erro durante a solicitação, registramos o erro no console.

```
.then(response => {  
    console.log(response.data.results);  
    .then(response => {  
        console.log(response.data.results);  
        setMovies(response.data.results);  
    })  
    .catch(error => {  
        console.error('Error fetching data: ', error);  
    });  
}, []);  
setMovies(response.data.results);  
})
```

FrontEnd - React

Agora que temos nossa comunicação com a API, vamos precisa-los exibi-los através de um retorno.

```
return (  
  <div>  
    <figure style={{ display: 'flex',  
flexWrap: 'wrap' }}>  
      {movies.map(movie => (  
        <Card key={movie.id} movie={movie} />  
      ))}  
    </figure>  
  </div>  
>;  
)
```

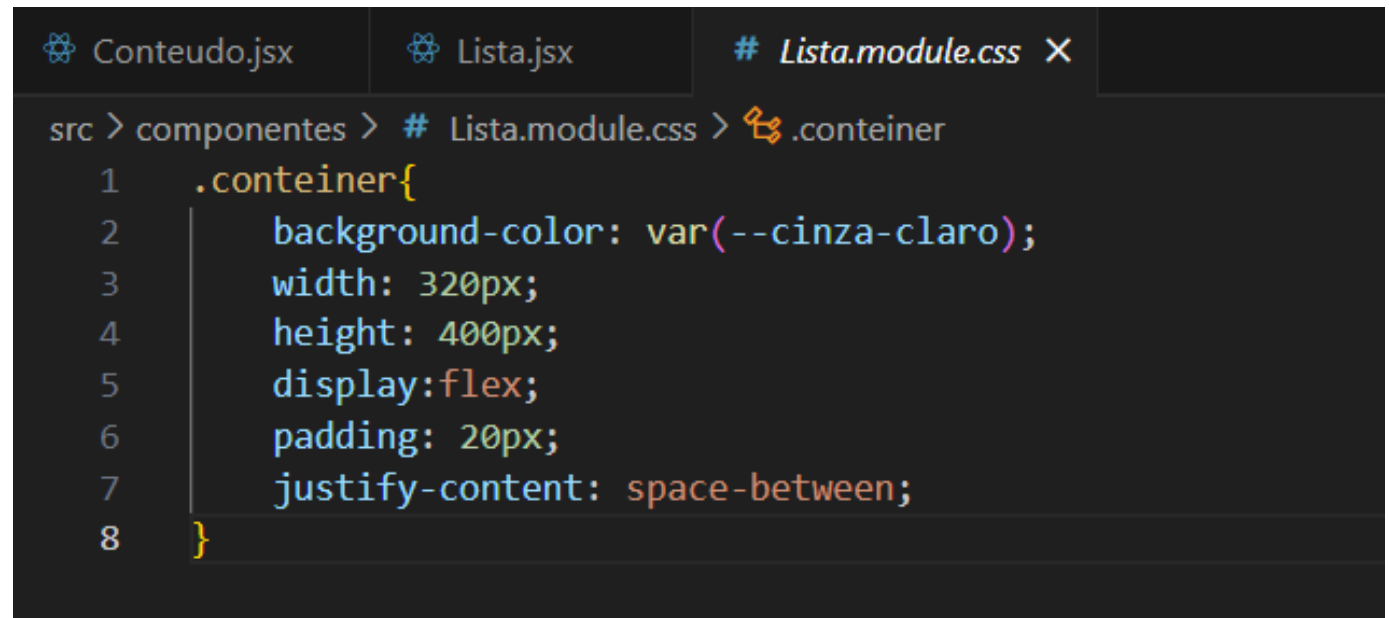
FrontEnd - React

Ficando assim o código completo.

```
Conteudo.jsx  Lista.jsx  X  # Lista.module.css
src > componentes > Lista.jsx > ...
1  import React, { useState, useEffect } from 'react';
2  import axios from 'axios';
3  import { Card } from './Card';
4
5  const API_KEY = 'af26cce282aecf5c6cc39a264f29d0a7';
6  const API_URL = 'https://api.themoviedb.org/3';
7
8  export function Lista() {
9    const [movies, setMovies] = useState([]);
10
11    useEffect(() => {
12      axios.get(`${API_URL}/movie/popular?api_key=${API_KEY}&language=pt-BR`)
13        .then(response => {
14          console.log(response.data.results);
15          setMovies(response.data.results);
16        })
17        .catch(error => {
18          console.error('Error fetching data: ', error);
19        });
20    }, []);
21
22    return (
23      <div>
24        <figure style={{ display: 'flex', flexWrap: 'wrap' }}>
25          {movies.map(movie => (
26            <Card key={movie.id} movie={movie} />
27          ))}
28        </figure>
29      </div>
30    );
31  }
```

FrontEnd - React

Já a estilização, podemos deixar assim



```
Conteudo.jsx  Lista.jsx  # Lista.module.css X
src > componentes > # Lista.module.css > .container
1  .container{
2      background-color: var(--cinza-claro);
3      width: 320px;
4      height: 400px;
5      display: flex;
6      padding: 20px;
7      justify-content: space-between;
8  }
```


FrontEnd - React

Em card, também será necessária algumas mudanças

```
src > componentes > Card.jsx > ...
1 import estilos from './Card.module.css'
2
3 export function Card({ movie }) {
4   return (
5     <div className={estilos.container}>
6       <h2>{movie.title}</h2>
7       <img src={`https://image.tmdb.org/t/p/w500/${movie.poster_path}`} alt={movie.title} />
8       <p>{movie.overview}</p>
9     </div>
10   );
11 }
12
```

```
1 .container{
2   background-color: var(--roxo-escuro);
3   width:280px;
4   height: 450px;
5   display:flex;
6   border-radius: 9px;
7   transition: transform .25s;
8   flex-direction: column;
9   align-items: center;
10  color:var(--cinza-claro);
11  margin-right:15px;
12  margin-bottom: 15px;
13  padding: 10px;
14 }
15 .container:hover{
16   transform: scale(1.025);
17   box-shadow: 0 0 15px var(--roxo-claro);
18 }
19
20 img{
21   width: 200px;
22   height: 250px;
23 }
24
25 h2{
26
27   color:var(--cinza-claro);
28   font-size: 20px;
29 }
30 p{
31   font-family: Arial, Helvetica, sans-serif;
32   font-size: 12px;
33 }
34
```

FrontEnd - React

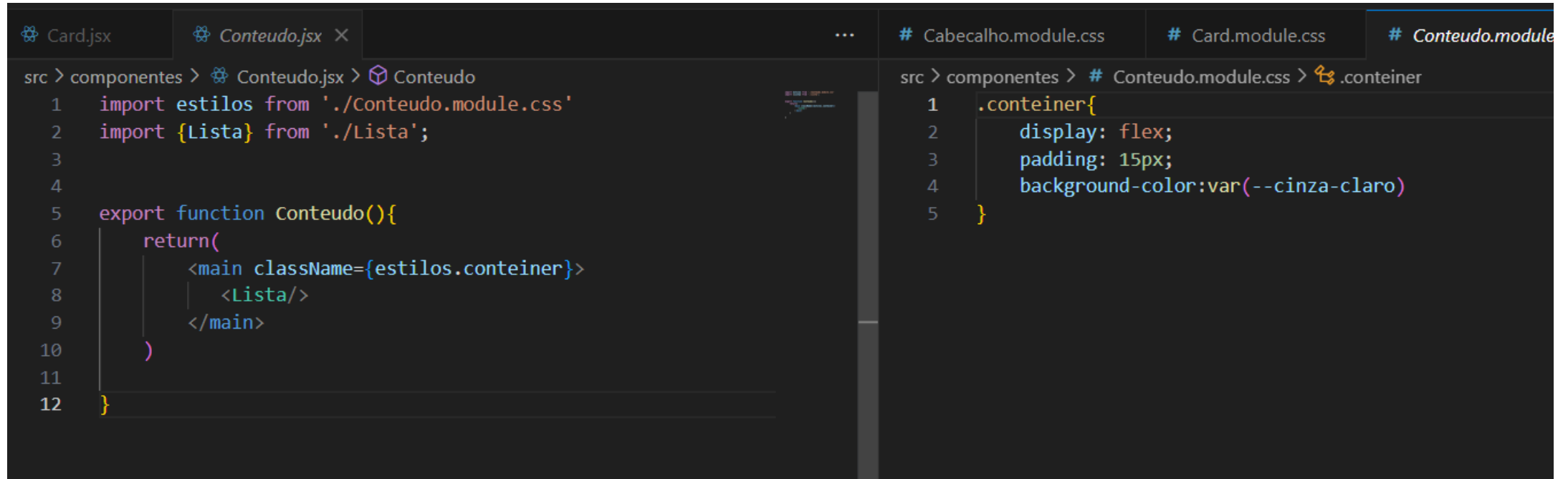
Repassando nossos componentes - Cabeçalho

```
Cabecalho.jsx X
src > componentes > Cabecalho.jsx > ...
1 import estilos from './Cabecalho.module.css'
2
3 export function Cabecalho(){
4   return(
5     <header className={estilos.container}>
6
7       <span class="material-symbols-outlined">movie</span>
8       <h1>Recomendações</h1>
9     </header>
10   )
11
12 }

# Cabecalho.module.css X
src > componentes > # Cabecalho.module.css > .container
1 .container{
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   background-color: var(--roxo-escuro);
6
7 }
8
9 h1{
10   color: var(--cinza-claro);
11 }
12
13 span{
14   color: var(--cinza-claro);
15 }
```

FrontEnd - React

Repassando nossos componentes - Conteudo



The screenshot shows a code editor with two files open. The left file is `Conteudo.jsx` and the right file is `Conteudo.module.css`.

Conteudo.jsx

```
src > componentes > Conteudo.jsx > Conteudo
1  import estilos from './Conteudo.module.css'
2  import {Lista} from './Lista';
3
4
5  export function Conteudo(){
6    return(
7      <main className={estilos.container}>
8        <Lista/>
9      </main>
10   )
11
12 }
```

Conteudo.module.css

```
src > componentes > Conteudo.module.css > .container
1  .container{
2    display: flex;
3    padding: 15px;
4    background-color:var(--cinza-claro)
5  }
```

FrontEnd - React

Repassando nossos componentes - Lateral

```
Card.jsx  Lateral.jsx  ...  module.css  # Card.module.css  # Conteudo.module.css  # Lateral.module.css  X  [icon]

src > componentes > Lateral.jsx > ...
1  import estilos from './Lateral.module.css';
2  import Profs from '../assets/Profs.jpeg'; // Caminho para a imagem n
3
4
5  export function Lateral(){
6    return(
7      <aside className={estilos.container}>
8        <img
9          className={estilos.imgLateral}
10         src = "https://media.istockphoto.com/id/1804164678/p
11
12        <img className={estilos.avatar}
13         src={Profs} alt="Descrição da imagem de professores" />
14      </aside>
15    )
16
17  }

src > componentes > # Lateral.module.css > .container
1  .container{
2    background-color: var(--cinza-claro);
3    display: flex;
4    flex-direction: column;
5  }
6
7  .imgLateral{
8    width: 100%;
9    height: 100px;
10   object-fit: cover;
11   margin-top: 0px;
12 }
13
14 .avatar{
15   width: 100px;
16   height: 100px;
17   border-radius: 10px;
18   border: 3px solid var(--roxo-claro);
19   align-self: center;
20   object-fit: cover;
21   margin: -3rem;
22
23
24 }
```



localhost:5173



Recomendações



Godzilla e Kong: O Novo Império



O poderoso Kong e o temível Godzilla se unem contra uma colossal ameaça mortal escondida no mundo dos humanos, que ameaça a existência de sua espécie e da nossa. Mergulhando profundamente nos mistérios da Ilha da Caveira e nas origens da Terra Oca, o filme irá explorar a antiga batalha de Titãs que ajudou a forjar esses seres extraordinários e os ligou à humanidade para sempre.

Kung Fu Panda 4



Po está prestes a se tornar o novo líder espiritual do Vale da Paz, mas antes que possa fazer isso, ele deve encontrar um sucessor para se tornar o novo Dragão Guerreiro. Ele parece encontrar uma em Zhen, uma raposa com muitas habilidades promissoras, mas que não gosta muito da ideia de Po treiná-la.

Matador de Aluguel



Um ex-lutador do UFC consegue um emprego como segurança em um estabelecimento na Flórida Keys, apenas para descobrir que este paraíso não é tudo o que parece.

Depois do Fim do Mundo



Ellie e Quinn são duas jovens mulheres que lutam diariamente para sobreviver em um mundo pós-apocalíptico, onde um vírus mortal matou quase toda a população do planeta.

Duna: Parte Dois



Madame Teia



Patos!



Criação dos Deuses I: Reino das Tempestades

