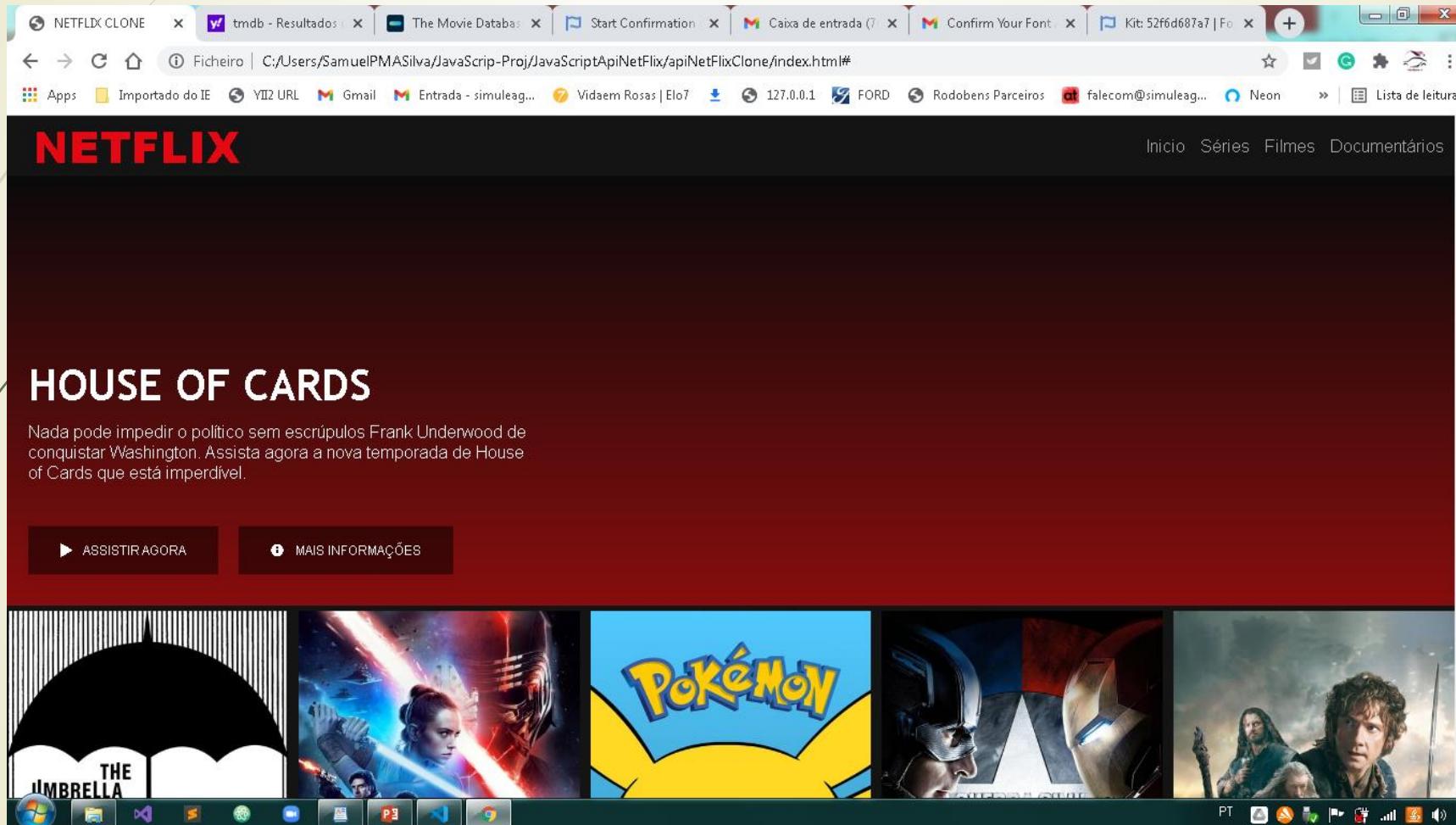


HTML

eBook – HTML Avançado Api Netflix



Williamsilva.codigo@gmail.com
[Github.com/swilliamsilva](https://github.com/swilliamsilva)

HTML Avançado – JavaScript

Visando a acessibilidade da informação produzi esta edição comentada, para aqueles que desejam recordar ou refazer este projeto passo a passo.

Espero que goste e nos envie sugestões.

JavaScript HTML – Projeto Código Comentado.

- **Criando uma API interface Netflix**

Acesso ao código: <https://github.com/swilliamsilva/ApiNetFlixHTML>

- A interface do principal site de streaming mundial utilizando tecnologia simples como, HTML, CSS3 e JavaScript.

Nesse projeto aprenderemos a estruturar um layout, técnicas de CSS3 com containers e variáveis, como posicionar os elementos com flexbox e como utilizar plug-ins JQUERY a favor da sua aplicação.

Tempo: 4 horas

Nível Avançado

Criando uma API interface Netflix

► Criando uma API interface NetFlix

Esse exercício é para quem está começando no mundo do front-end e HTML deseja aprender a programar sem se tornar mero replicador de código.

É para ter uma base de conhecimento em que você possa aplicar em qualquer código.

Entender o código que está sendo criado e com esse entendimento ir além na programação.

É um site completo com responsavidade.

Você precisa ter instalado uma IDE que esteja acostumado e o Chrome onde você já tem alguma funcionalidades como inspecionar elementos em tela.

Preparando a arquitetura

- Criando uma API interface NetFlix

- Preparando nossos arquivos para a edição,

Estando na sua IDE favorita, VS CODE, NodPad, NotPad++, Sublime.

A primeira coisa a fazer para estruturar o projeto é criar as pastas:

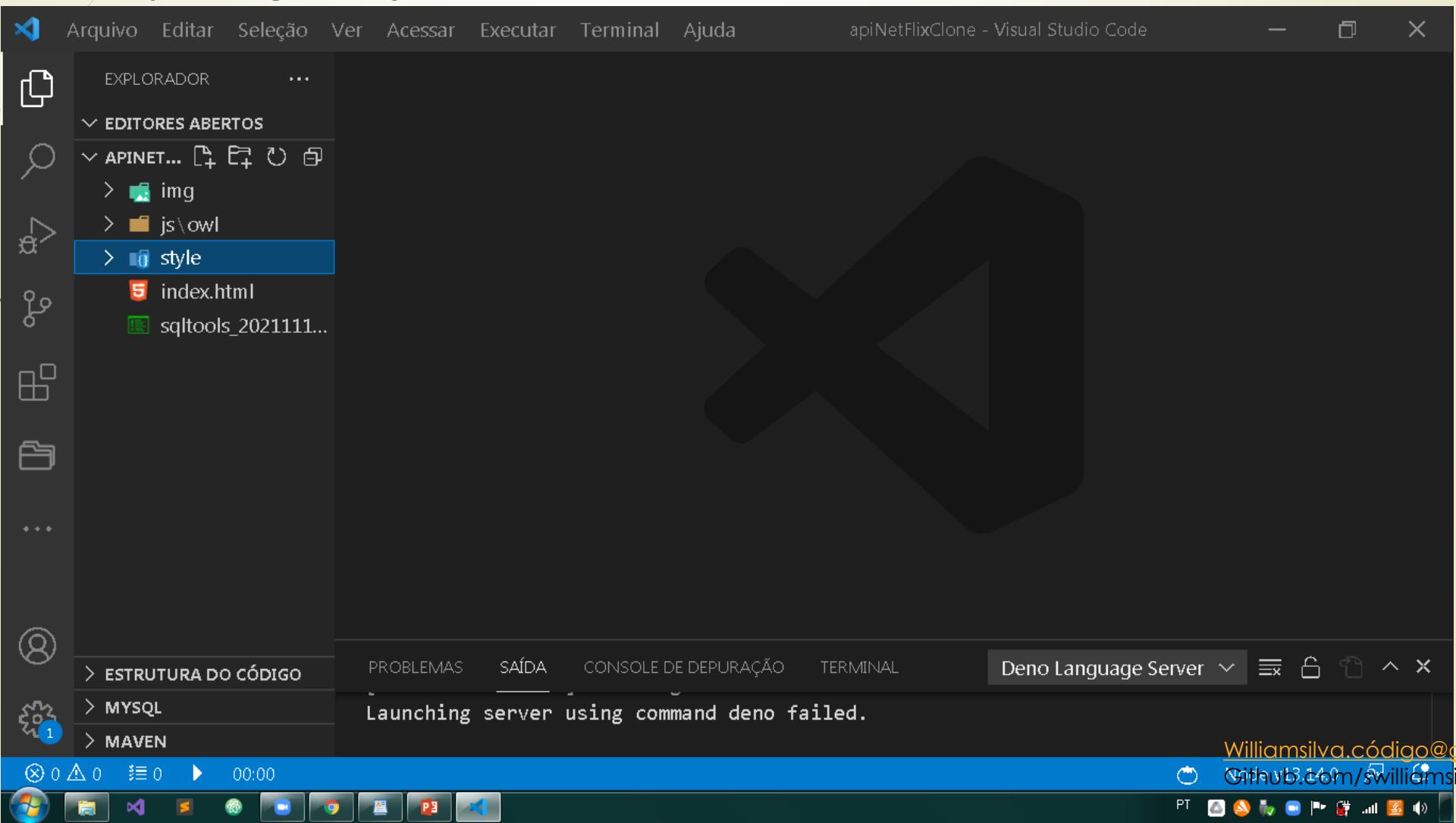
Styles: Que é a pasta onde vai ficar os arquivos de CSS

JS: pasta onde vai ficar os arquivos javascript-02 crie também o arquivo raiz que se chama **index.html**

Img: Que é a pasta onde vai ficar as imagens, que também pertence aos resources, recursos. Ainda podemos ter outras pasta para áudio ou coisas do tipo.

Preparando a arquitetura

► Veja a organização das pastas



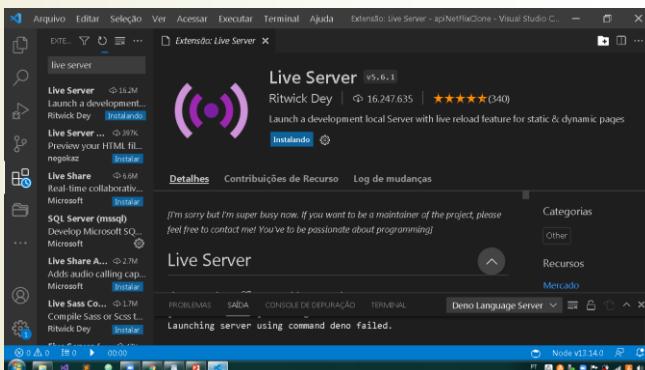
Instalando extensão no VSCode

► Criando uma API interface Netflix

No visual code uma ferramenta que ajuda muito é a extensão LiveServer, verifique se ela esta instalada. Se não tem instale.

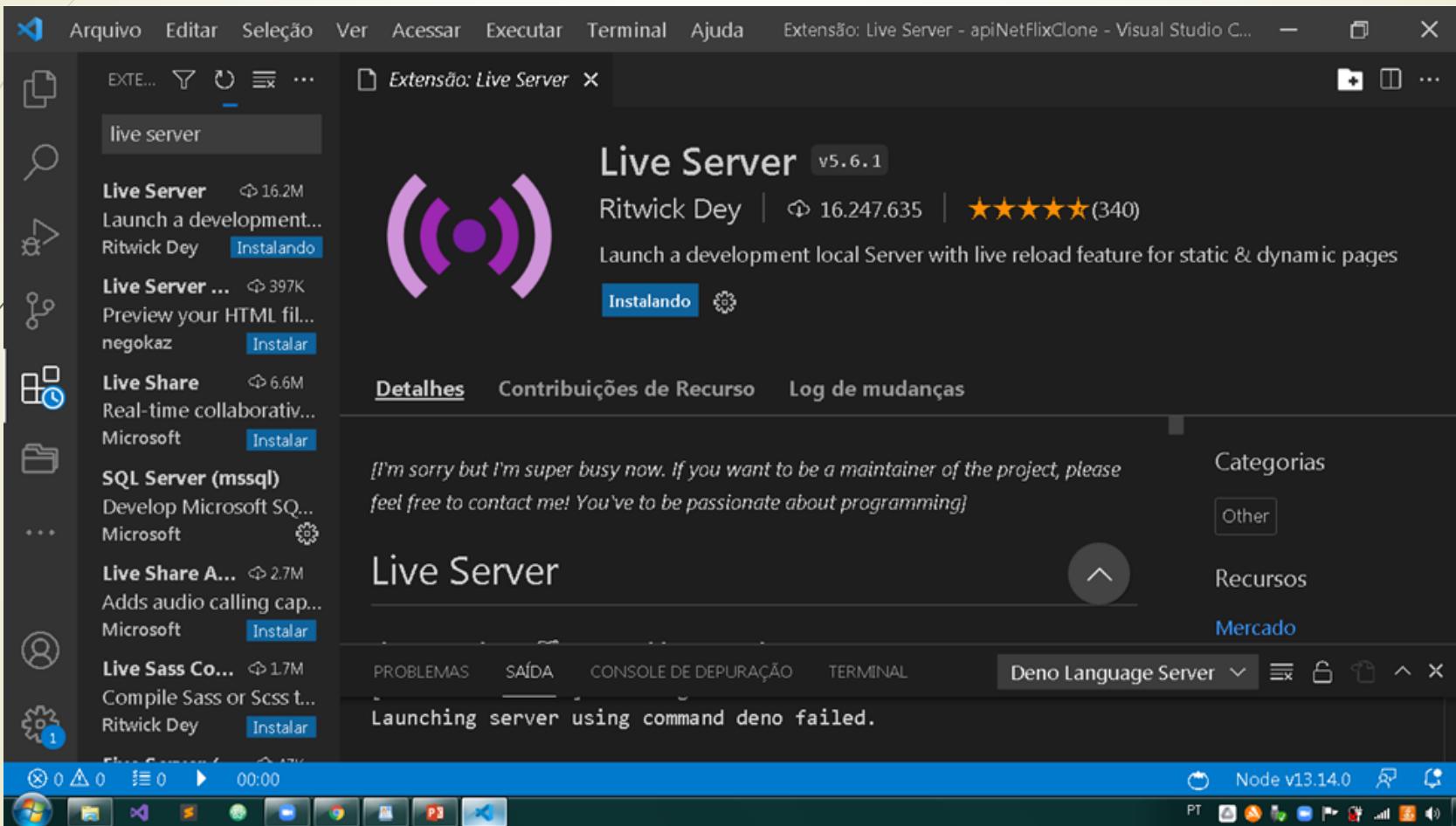
Ele abre uma janelinha que mostra o resultado quando você fizer uma alteração qualquer no seu código dispensando ter que ir sempre no Browser.

Ele cria uma porta no seu navegador que fica ouvindo e qualquer alteração no código ele já executa.



Live Server

► Aplique no VSCode



Extensão Auto Rename Tag

► Instalando no VSCode

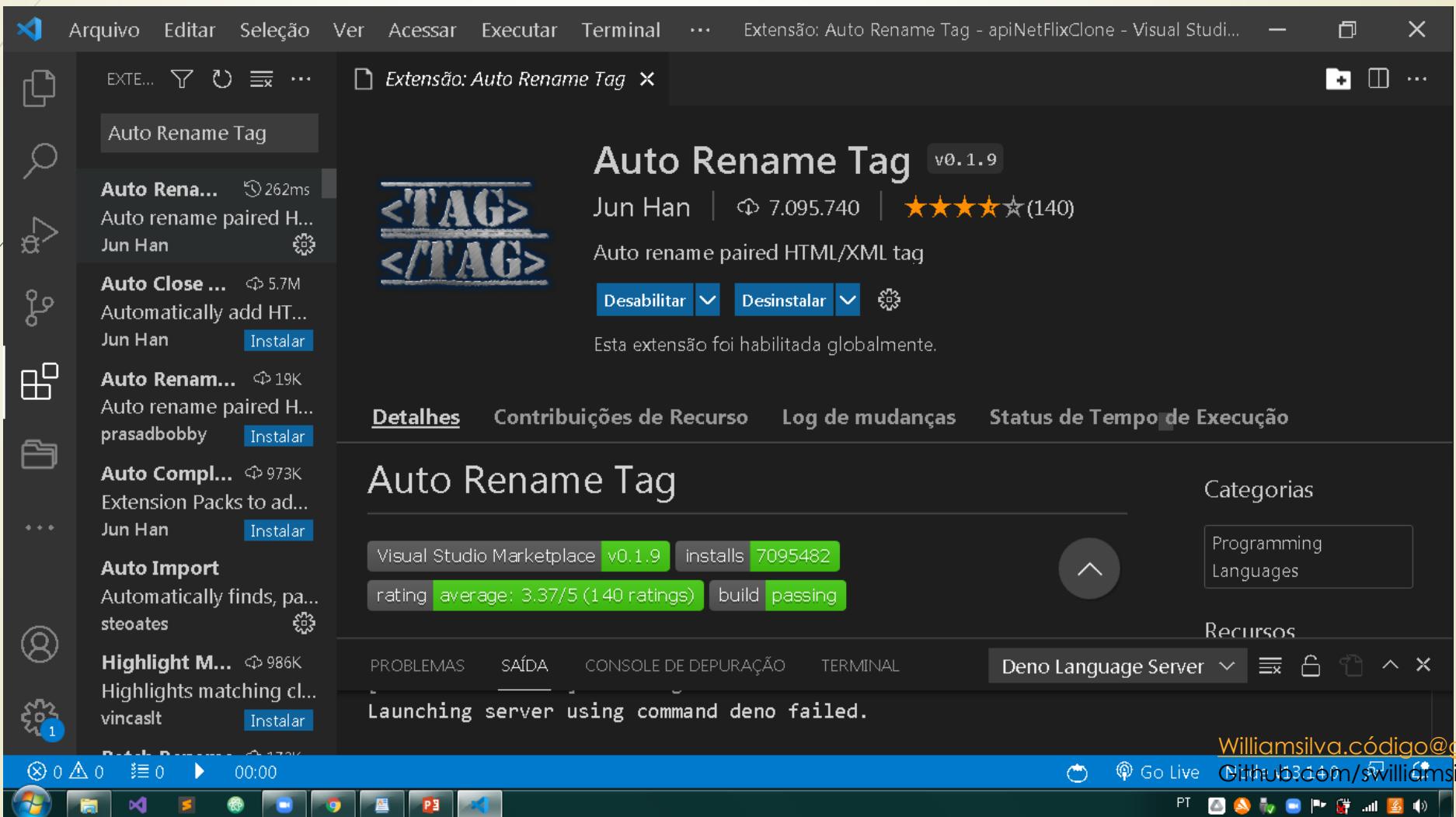
► Outra extensão interessante **do VSCode é AutoRename Tag**,

Ela é útil para quando você está editando uma TAG e você esquece de editar a tag de fechamento ou você precisa executar em dois lugares, essa extensão ajuda a gente a não ter esse problema.

Uma vez que você abre a tag de inicio ele já fecha a tag final.

Auto Rename Tag

- Confira a tela de instalação no VSCode

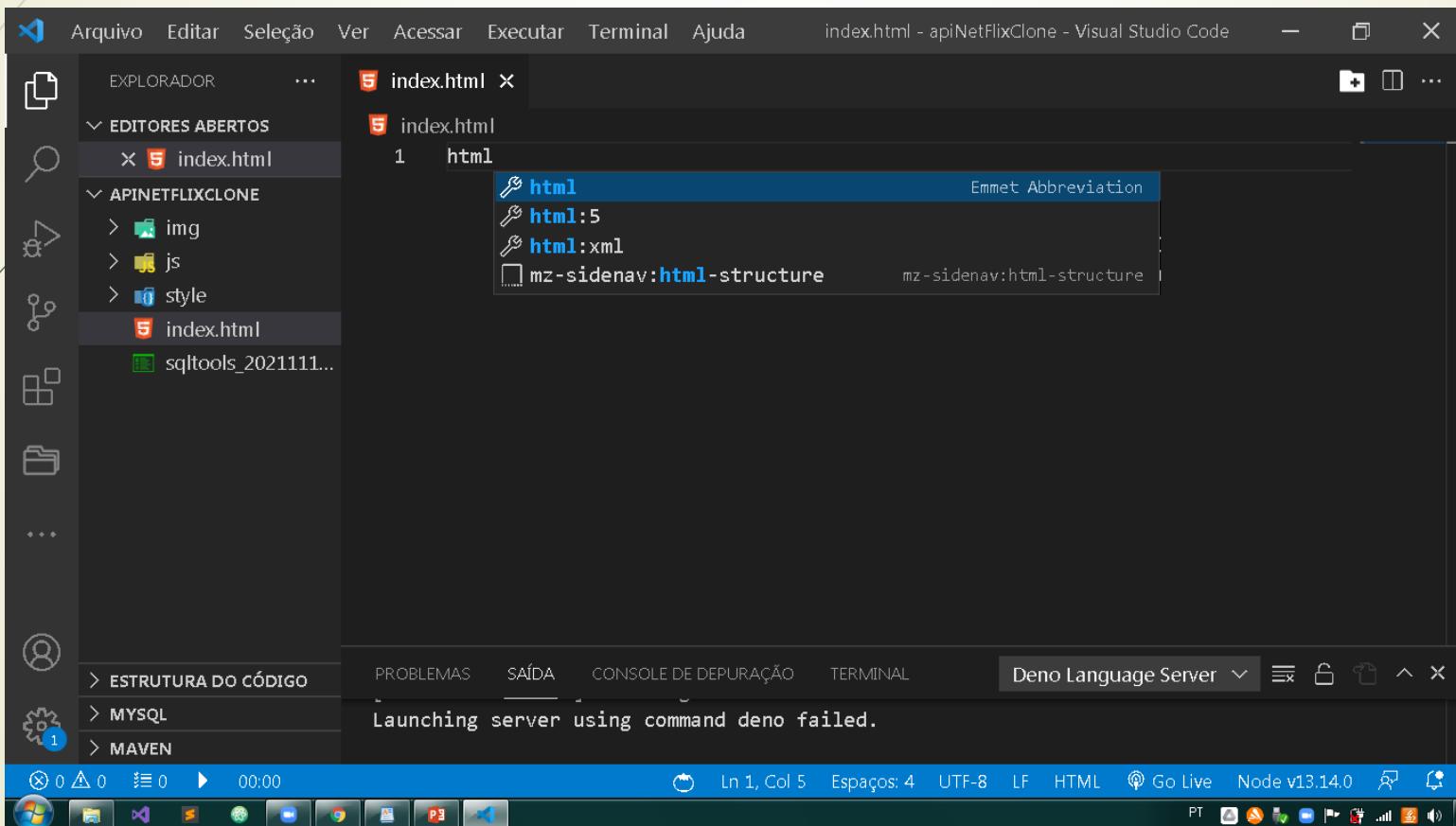


Usando o emmet

- ▶ Criando uma API interface NetFlix
- ▶ Começando a digitar o código no VS CODE uma ferramenta bastante útil é o **emmet** que completa o código para você facilitando a digitação e muito.
- ▶ Como se fosse atalhos. Por exemplo: A partir do momento que eu digito html ele já posiciona as opções para eu escolher e clicando nela, ela já trás o código.
- ▶ Quando eu clico em html5 ele já aparece a estrutura padrão do código html pronta.

Usando o emmt

► Veja na prática o recurso



Criando o Index.html

► Digite html ou ! no vscode

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** Arquivo, Editar, Seleção, Ver, Acessar, Executar, Terminal, Ajuda.
- Title Bar:** index.html - apiNetFlixClone - Visual Studio Code.
- Explorer View:** Shows the file structure:
 - EDITORES ABERTOS: index.html
 - APINETFLIXCLONE: img, js, style, index.html (selected), sqltools_2021111...
- Code Editor:** Displays the HTML code for index.html. The title "Document" is selected.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
  </head>
  <body>
  </body>
</html>
```
- Bottom Status Bar:** Shows the current file path (Ln 6, Col 54 (12 selecionado)), encoding (UTF-8), and other status information.
- Terminal:** Shows an error message: [Error - 09:44:46] Starting client failed and Launching server using command deno failed.

Estrutura básica do html

► Código HTML

Lembrando que o código html é dividido em duas partes:

<head> que é a cabeça, a parte pensante do código. Ele guarda mais informações de configurações do que vai acontecer ali.

Uma das configurações é o **<met charset="utf-8">** para poder acessar e aceitar caracteres acentuados.

E tem também o

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Para quando você começar a ver alguma coisa de mídia, e de quando você começar a trabalhar com ambiente mobile, onde você vai ver esses detalhes de escala, para pegar a altura e a largura de um componente. Muito útil para se estudar no futuro.

Titulo do documento

- Altere o **conteúdo** entre as tags

```
<title> Document</title>
```

O titulo da página que no nosso caso a gente vai chamar de

```
<title> NETFLIX CLONE </title>
```

Isto é o que aparece na barrinha do nosso site.

A segunda metade do HTML

- Criando o corpo da aplicação.

E temos o **<body>** ele é o corpo de nossa aplicação, onde fica todas as coisas em si, em geral.

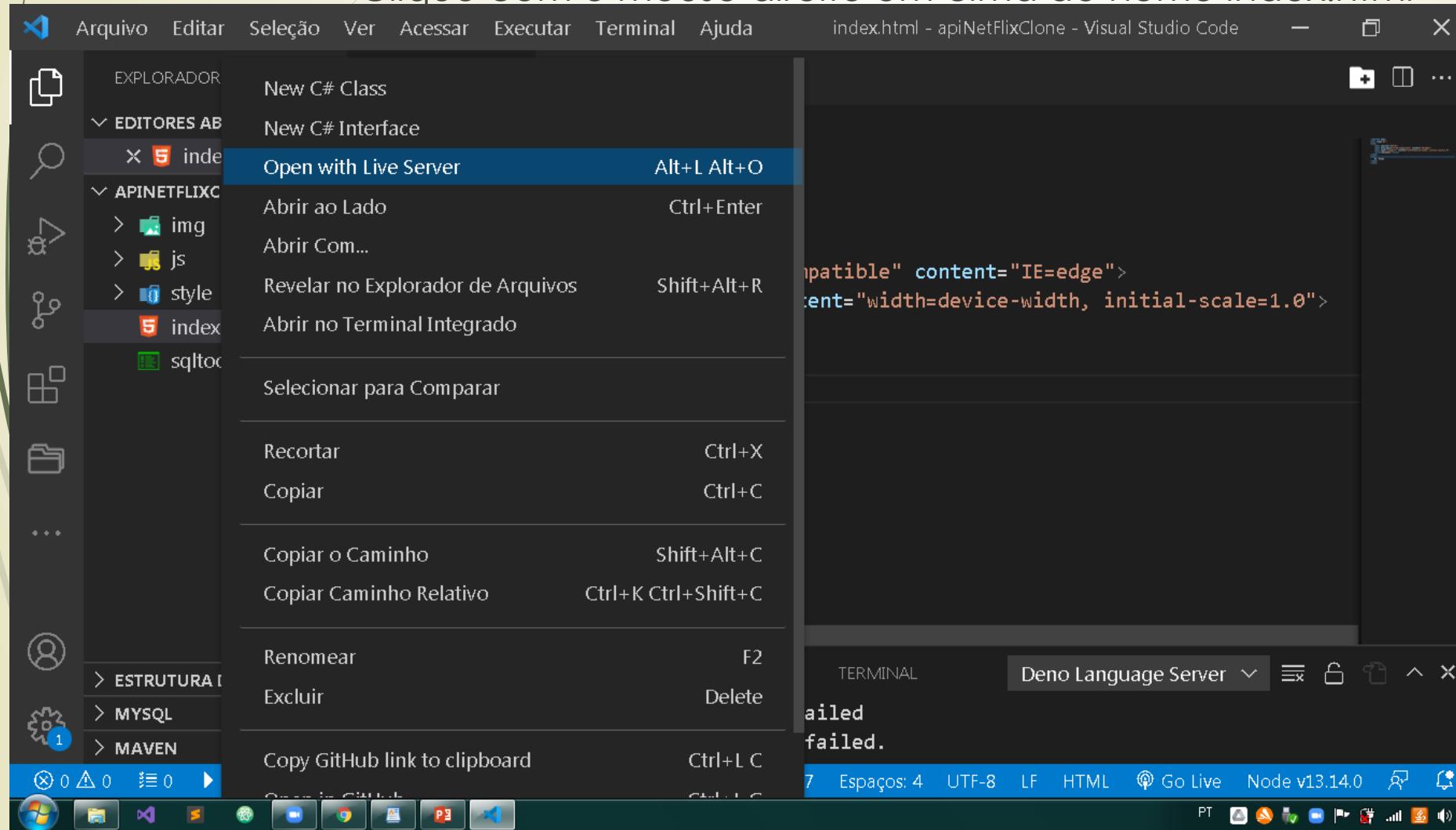
Então o que você digitar aqui entre as tags vai refletir direto lá na página.

Vamos **abrir o life server** clicando com o mouse direito em cima do **index.html** para garantir que está tudo bem com o ambiente.

Ou seja toda alteração que eu fizer no código já aparece em uma página do browser.

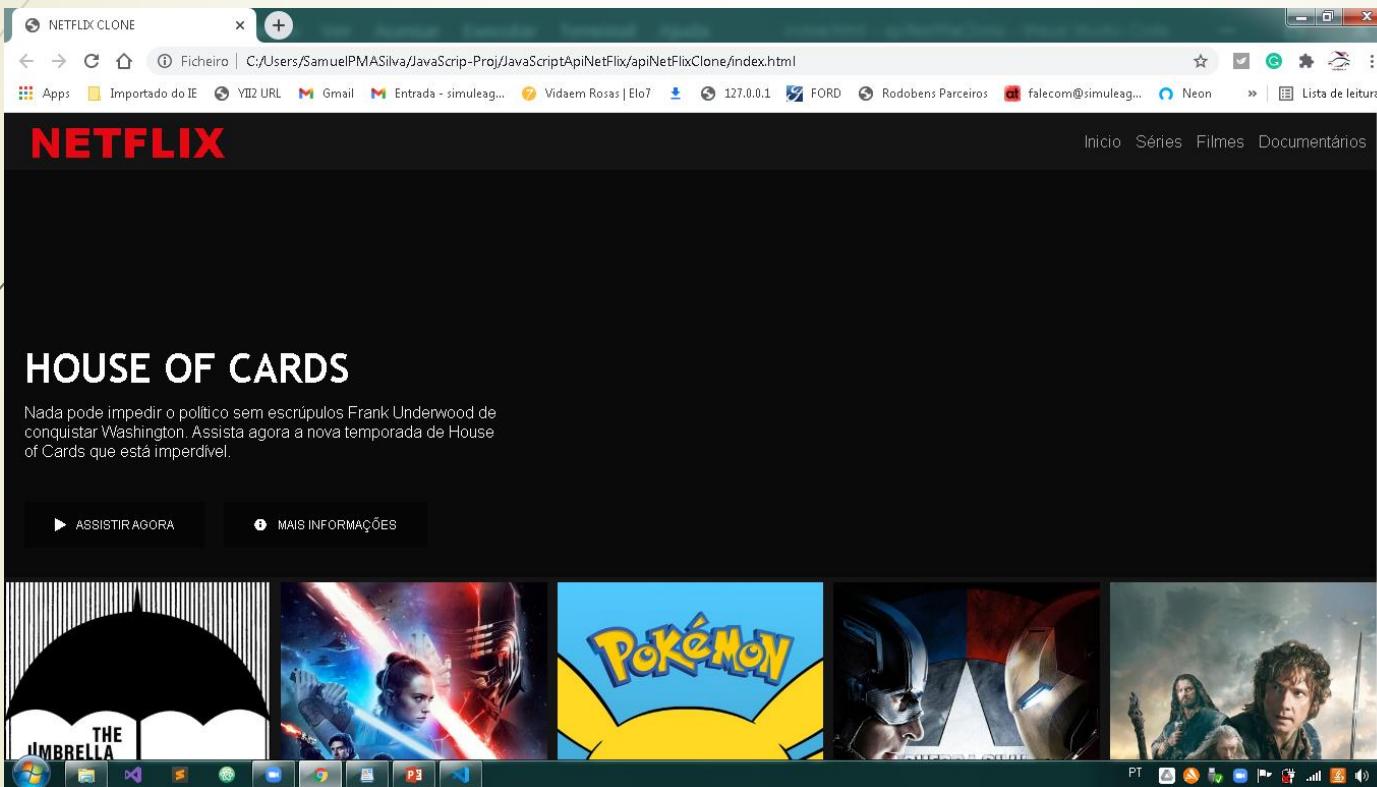
Abrindo o index.html no Browser

► Clique com o mouse direito em cima do nome index.html



Pode executar através do chrome

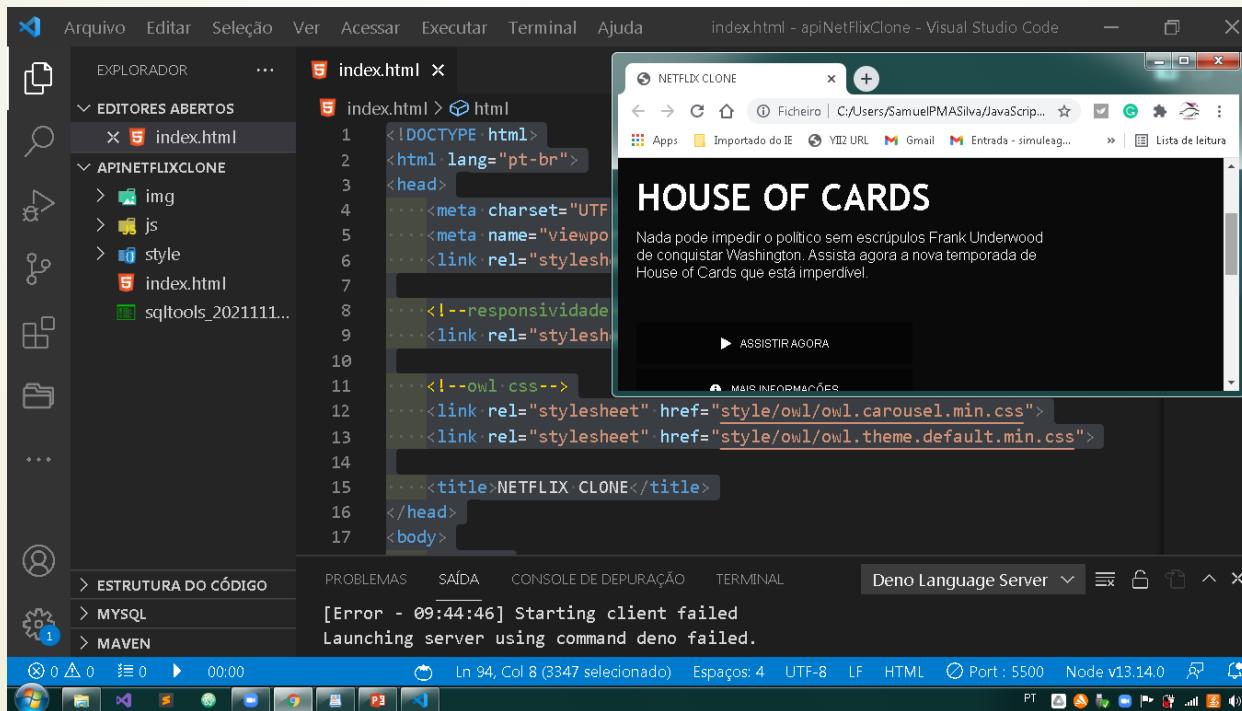
► No Browser vai aparecer assim.



Usando mais de uma tela

► Para você editar e inspecionar ao mesmo tempo a criação

Minimize a sua tela deixando tanto a tela do navegador quanto a do editor, isso facilita bastante a visão para o desenvolvedor.



Tag Semântica

- ▶ Tag que ajudam a organizar o código

Então vamos estudar o **nosso MENU** que vai dar uma cara para o nosso site.

A primeira coisa a fazer é definir uma **tag semântica**, então você vai ver que tudo em html é feita com **<div>** isto porque tudo que renderizamos na tela pensamos em **formato de caixa**,

E tem tag que também tem DIV mas que são chamadas tag semânticas que são mais para você entender do que para quem estiver olhando para o sistema.

Exemplo de tag semântica:

`<header></header>`

`<footer></footer>`

`<section></section>`

`<article></article>`, todas são div e representam caixas aqui.

Os primeiros passos

► Iniciando a aplicação

A primeira coisa a fazer é definir o cabeçalho.

Dentro dele vamos definir o nosso menu de navegação ou seja nossa **navbar**. Além da navbar temos um texto que serve como **logotipo**.

E tudo isso vai estar dentro de uma mesma caixinha.

Essa **caixinha** vamos chamar de **container**;

Vamos dar uma classe para ele chamada container porque vamos fazer com isso alguma adições posteriormente no html

```
<div class="container">  
  <h2> NETFLIX</h2>
```

e vamos usar mais **tag semântica** chamada de **<nav>** e ela vem de ação mesmo e geralmente é uma tag que a gente usa na hora de criar menus de navegação.

Já vamos colocar também alguns links que serão os links que iremos acessar dentro nosso menu.

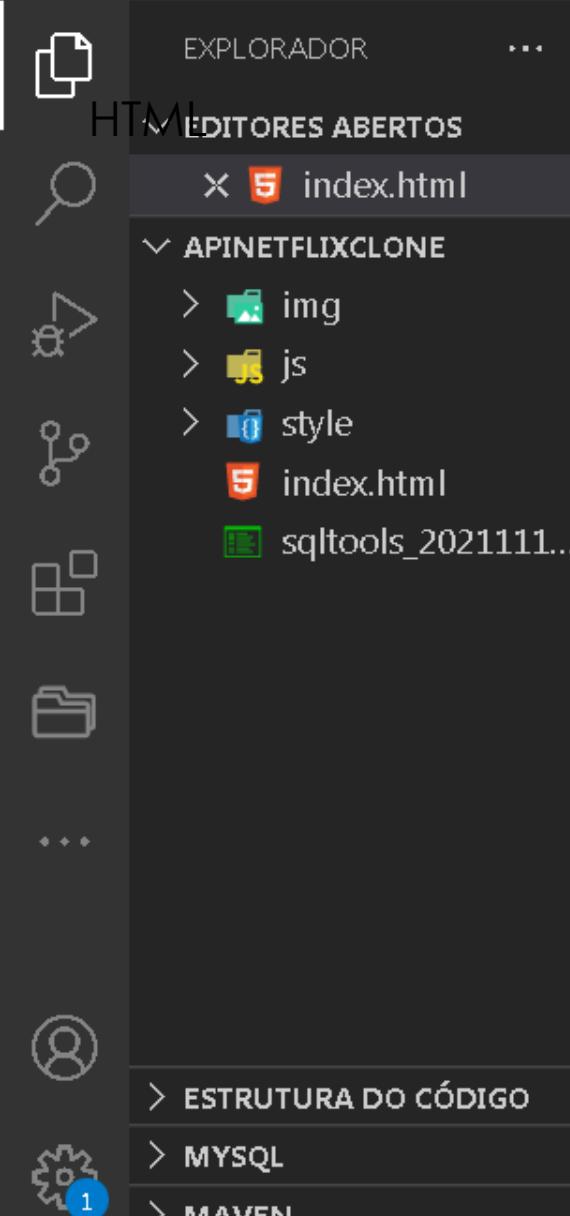
Veja no código como ficou.

Conhecer bem a IDE é fundamental

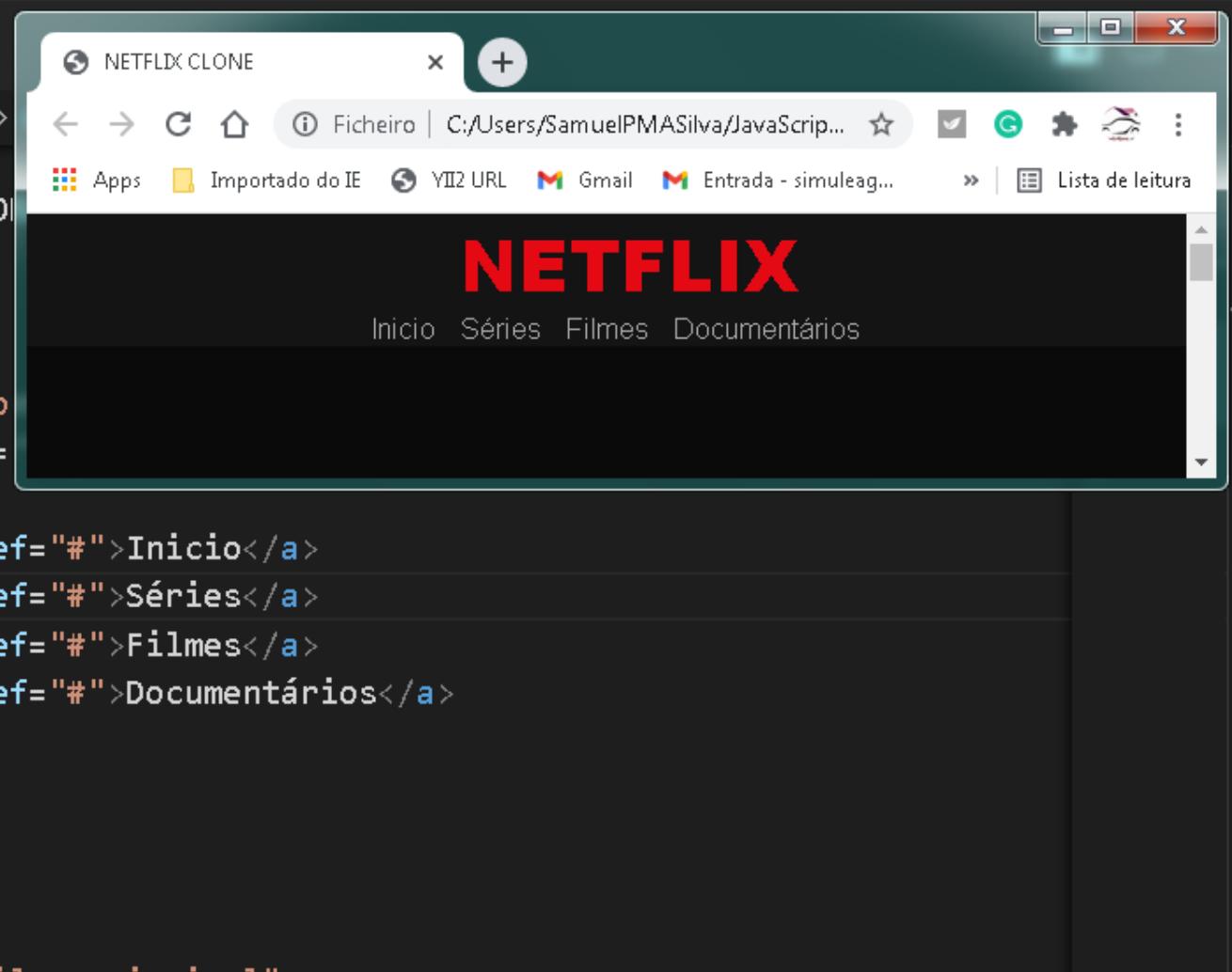
- ▶ Conheça as tag de atalho do VsCode

“**Alt+Shift no VSCode** permite você duplicar a linha, selecione a linha segure Alt Shift e use a seta para baixo para duplicar uma linha.”

Se você salvar Ctrl S e atualizar no Browse já pode ver o menu e o resultado.
Aqui nesse código já está pronto.



```
5 index.html x
5 index.html > html > body >
14
15     <title>NETFLIX CLO
16     </head>
17     <body>
18         <header>
19             <div class="co
20                 <h2 class=
21                 <nav>
22                     <a href=
23                     <a href=
24                     <a href=
25                     <a href=
26             </nav>
27         </div>
28     </header>
29
30     <main>
```



[Error - 09:44:46] Starting client failed
Launching server using command deno failed.

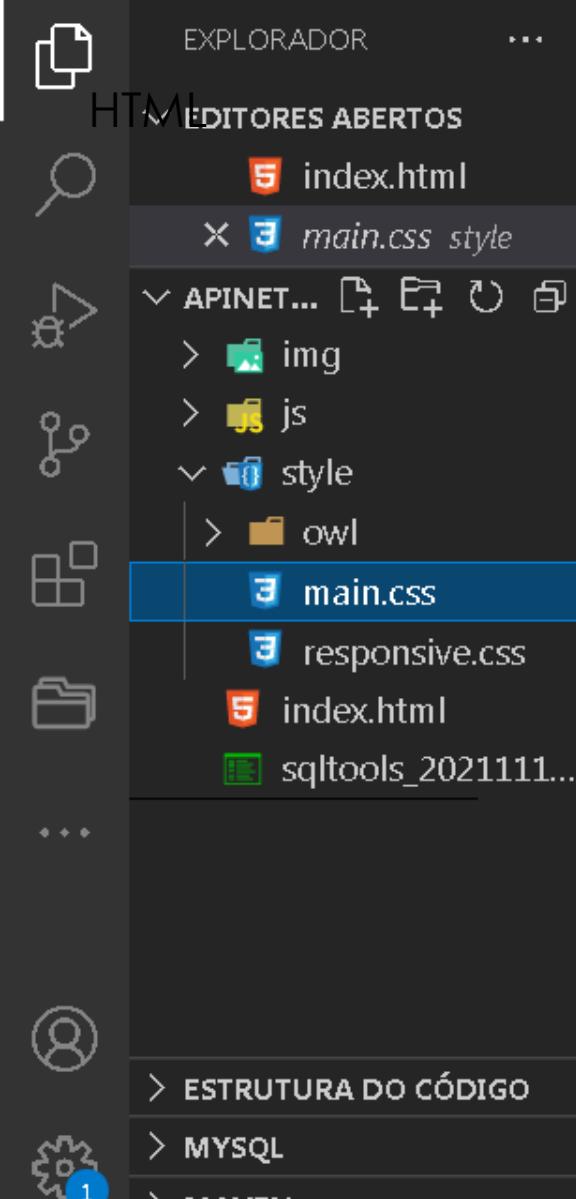
HTML tem como auxiliar o CSS3

- Criando uma página de estilos

O html não cuida da formatação em si, a **formatação** fica a cargo do **CSS3** que fica no arquivo **Styles** que vamos ver mais a frente.

O html é responsável muito mais pela estrutura do seu site do que para a sua formatação, embora de para formatar pelo html.

Vamos visitar nossa **folha de estilos** aqui **chamada de main.css**



5 index.html 3 main.css X

```
style > 3 main.css > :root
35 header nav a{
36     text-decoration:none;
37     color:#AAA;
38     margin-right: 10px;
39 }
40
41
42 header nav a:hover{
43     color:#fff;
44 }
45
46 /*filme principal*/
47 .filme-principal{
48     font-size:16px;
49     background: linear-gradient(rgba(0,0,0,.50),rgba(0,0,0,.50)100%), url(
50
51     height:400px;
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

[Error - 09:44:46] Starting client failed

Launching server using command deno failed.

Deno Language Server ▾ ^ X

Configurando estilos

- Tudo que é arte é configurado no CSS3

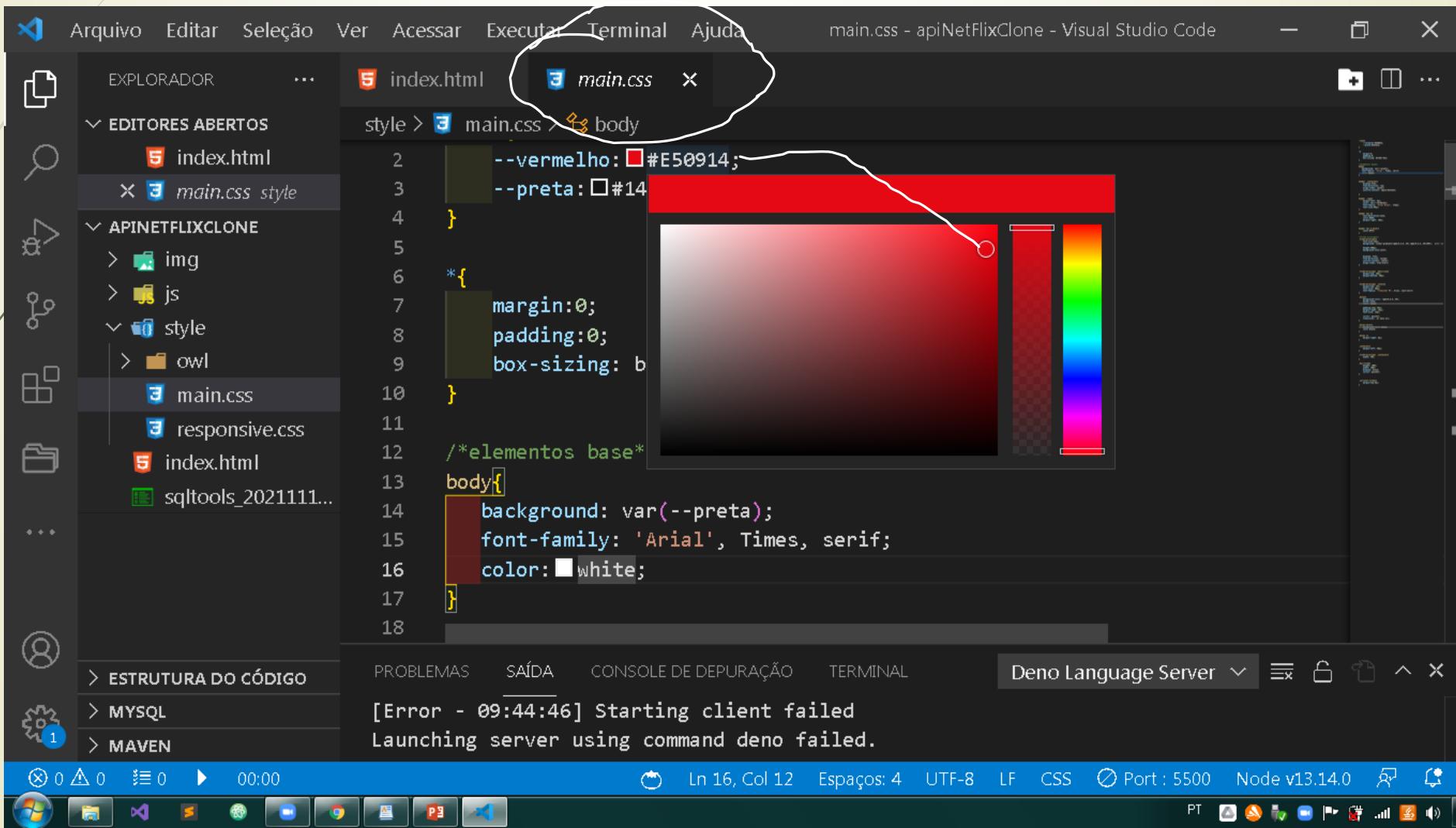
Dentro do CSS é que fazemos toda a nossa configuração de estilos, nossa formatação.

Por exemplo: **background** que **altera a cor** de fundo da tela para **preta**.

```
/*elementos base*/  
body{  
    background: var(--preta);  
    font-family: 'Arial', Times, serif;  
    color:white;  
}
```

CORES, MARGENS, FONTES, TAMANHO

► A IDE te ajuda muito trazendo as opções.



MUITO MAIS FOLHAS DE ESTILOS

- ▶ Crie mais de uma folha de estilos

Você pode ter em HTML varias folhas de estilos.

Mas precisa **informar isso** no programa para que ele possa localizar e para isso **na <head>** nos vamos digitar link e informar onde esta o arquivo de estilos.

Você pode ter vários arquivos de estilos conforme mostra o código a seguir.

```
<link rel="stylesheet" href="style/main.css">
```

stylesheet = Planilha de estilos

style = pasta onde se encontra o arquivo de estilos

main.css = arquivo de estilos.

O link também oferece uma série de arquivos para você referenciar. No nosso exemplo o arquivo html vai linkar com o arquivo css.

EXPLORADOR ...

HTML EDITORES ABERTOS

- index.html
- main.css style

APINET...    

- > img
- > js
- ✓ style
 - > owl
 - main.css
 - responsive.css
- index.html
- sqltools_2021111...

ESTRUTURA DO CÓDIGO

MYSQL

MAVEN

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Deno Language Server

[Error - 09:44:46] Starting client failed
Launching server using command deno failed.

0 0 0 0 0 0 0:00

Ln 23, Col 16 Espaços: 4 UTF-8 LF HTML

Port 3000 Node v15.14.0

Github.com/swilliamsilva

PT



index.html x main.css

index.html > html > body > header > div.container > nav

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="style/main.css">
7
8      <!--responsividade-->
9      <link rel="stylesheet" href="style/responsive.css">
10
11     <!--owl css-->
12     <link rel="stylesheet" href="style/owl/owl.carousel.min.css">
13     <link rel="stylesheet" href="style/owl/owl.theme.default.min.css">
14
15     <title>NETFLIX CLONE</title>
16 </head>
17 <body>
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Deno Language Server

[Error - 09:44:46] Starting client failed
Launching server using command deno failed.

Ln 23, Col 16 Espaços: 4 UTF-8 LF HTML

Port 3000 Node v15.14.0

Github.com/swilliamsilva

PT



Valores pré definidos e a definir.

► Fazendo a formatação

Cada navegador joga um valor de margin e padding e esse valor pode não ser padronizado. O que pode acontecer é que se você constrói um site em um navegador ele pode estar diferente em outro navegador. E uma das partes importantes do css é criar regras de formatação que deixa o seu site padrão independente do navegador.

Então isso é importante não só para dar carinha ao seu site mas também para forçar e manter um padrão de exibição

Usando um Reset

- Dar uma limpeza nas configurações.

RESET CSS

Muita coisa sobre a manutenção de padrão de exibição você resolve colocando um ponto zero. Mas vamos usar uma **técnica** muito comum na internet que é o **reset zero**.

A única coisa que você vai precisar é de um * ele se refere a todo documento que esta sendo renderizado naquele momento .

Então colocamos a margin por padrão ela vai ser zero, o padding e seus elementos vai ser zero também, eles já retiram qualquer espaçamento diferente que não deveriam ter aqui e outra coisa que a gente vai colocar aqui é o nosso **box-sizing: border-box**; uma vez que você declarou que um elemento tem 400px de largura e você começa a por outros conteúdos dentro dele forem maior que 400px.

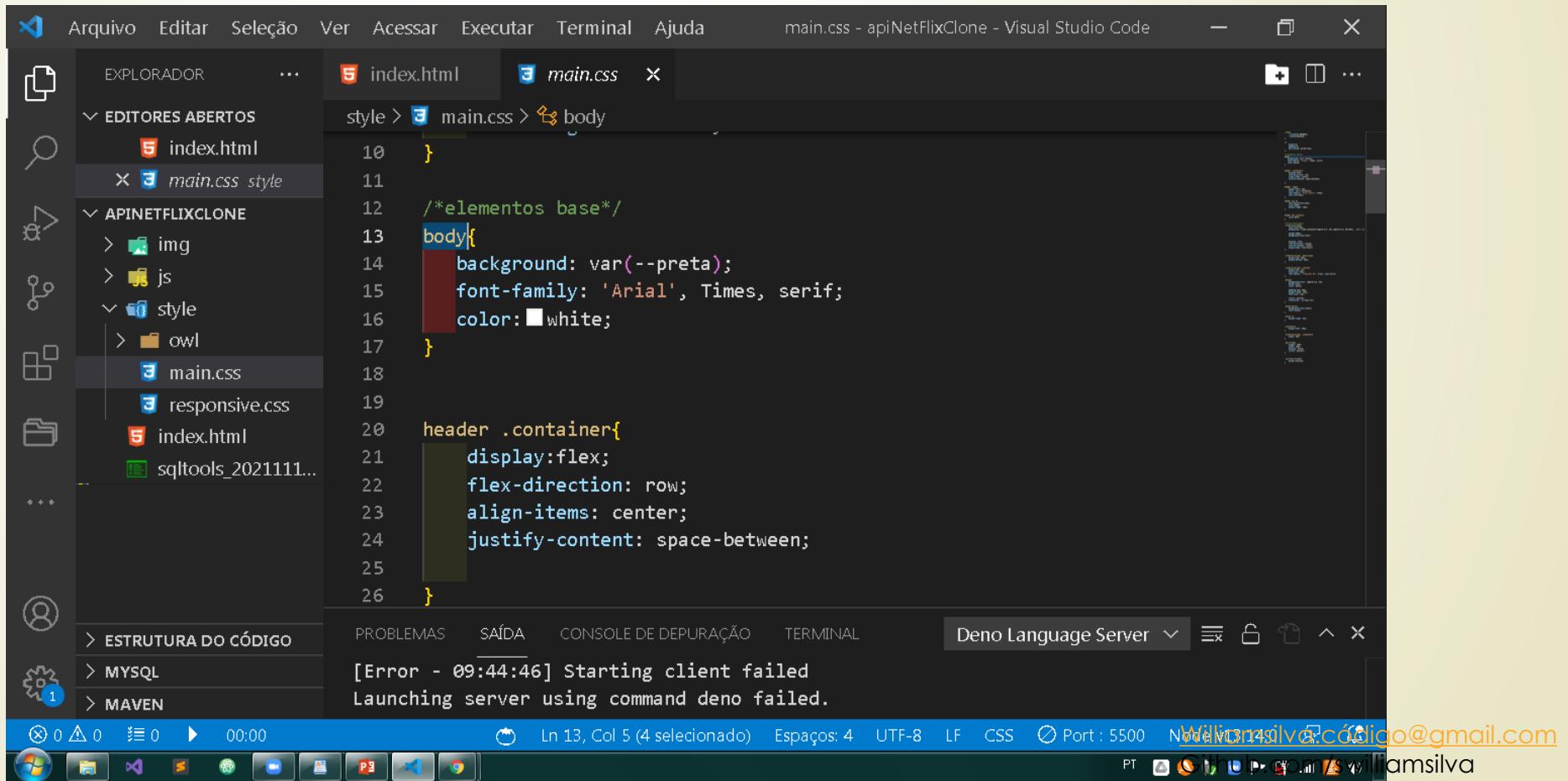
Eles irão começar a empurrar a caixa e para garantir que isso não aconteça usamos o **box-sizing: border-box**. A caixa é o container que foi definido na header.

- Veja o exemplo no código

Referencia pelo nome da tag

- Fazendo um link entre html e css

Toda vez que eu **me referencia** a uma **tag no css** eu coloco o **nome** da **tag**.



The screenshot shows the Visual Studio Code interface with the following details:

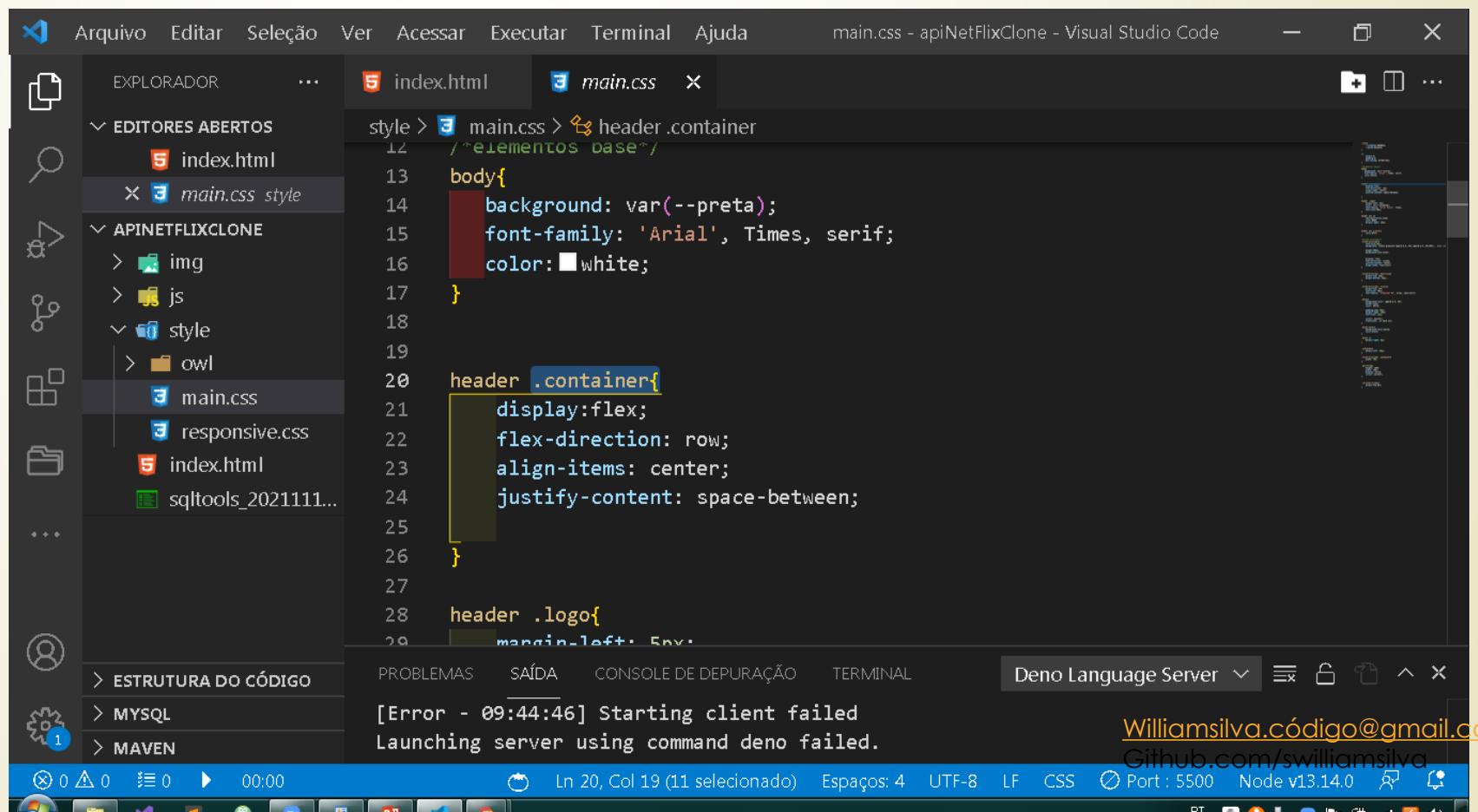
- File Explorer:** Shows files like index.html, main.css, img, js, style, owl, responsive.css, and index.html.
- Editor:** The main.css file is open, showing the following CSS code:

```
body{  
background: var(--preta);  
font-family: 'Arial', Times, serif;  
color: white;  
}  
  
header .container{  
display:flex;  
flex-direction: row;  
align-items: center;  
justify-content: space-between;  
}
```
- Bottom Status Bar:** Shows the current file is main.css - apiNetFlixClone - Visual Studio Code, with status indicators for problems, output, and terminal.
- Bottom Right:** Shows the user's email address: William.silva.codigo@gmail.com.

Fazendo referencia a uma classe

- Fazendo links entre o html e o css

Toda vez que **vou me referenciar a uma class** no css eu coloco o nome da **class depois de um ponto.**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like index.html, main.css, img, js, style, owl, responsive.css, and index.html.
- Editor:** The main.css file is open, displaying the following CSS code:

```
style > main.css > header .container
12 /*ELEMENTOS BASE*/
13 body{
14   background: var(--preta);
15   font-family: 'Arial', Times, serif;
16   color: white;
17 }

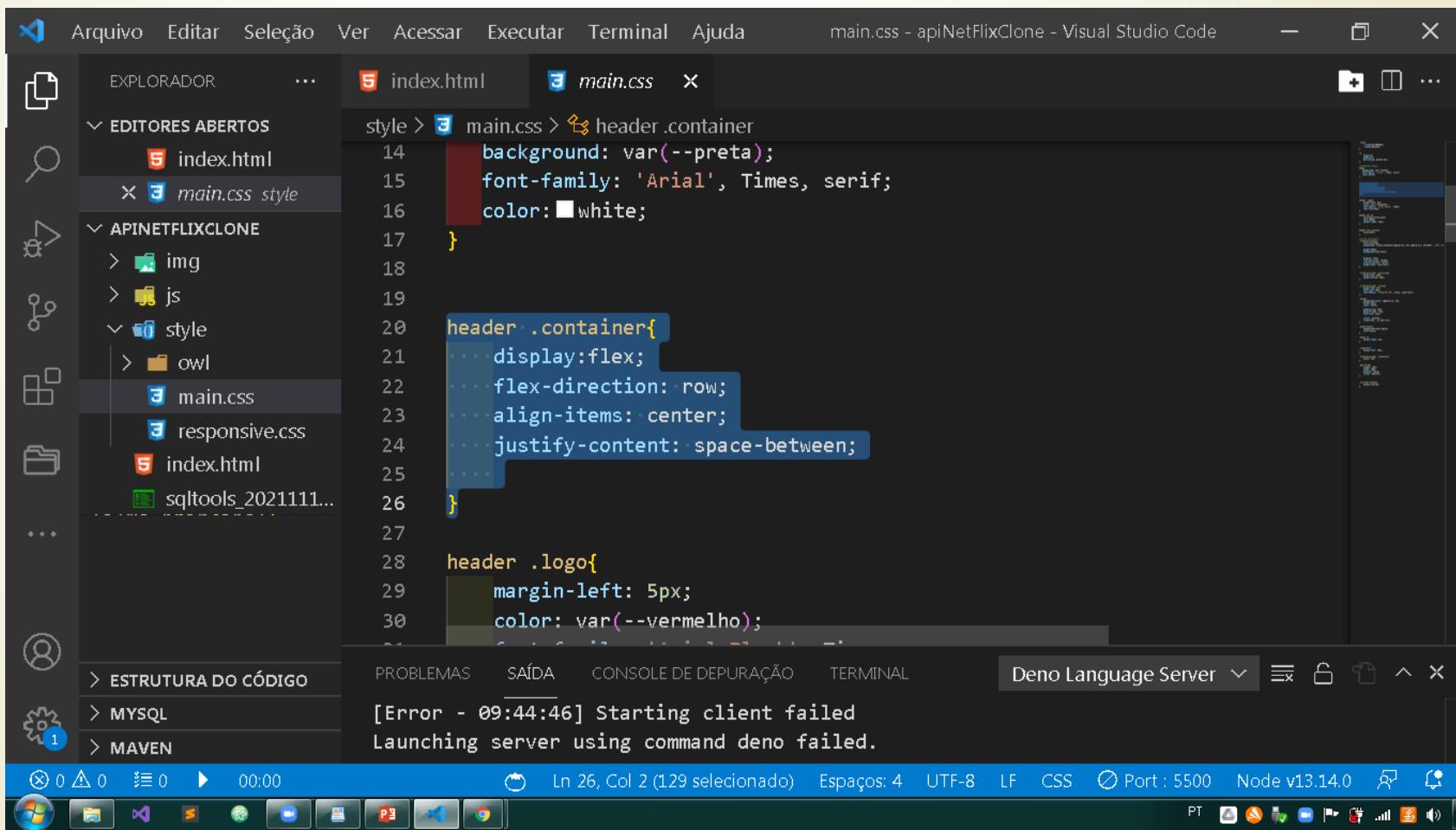
20 header .container{
21   display:flex;
22   flex-direction: row;
23   align-items: center;
24   justify-content: space-between;
25 }

28 header .logo{
29   margin-left: 5px;
```
- Bottom Status Bar:** Shows file navigation icons, a timer (00:00), and terminal status: "Ln 20, Col 19 (11 selecionado) Espaços: 4 UTF-8 LF CSS".
- Bottom Right:** Includes email (Williamsilva.codigo@gmail.com) and GitHub links (Github.com/swilliamsilva).

JavaScript

- Criando uma API interface NetFlix

Exemplo de uma caixa:



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "APINETFLIXCLONE" folder: index.html, main.css, img, js, style (with owl and main.css), responsive.css, index.html, and sqltools_20211111...
- Editor:** The "main.css" file is open, displaying CSS code for a header container.

```
style > main.css > header .container
14 background: var(--preta);
15 font-family: 'Arial', Times, serif;
16 color: white;
17 }

header .container{
18   display:flex;
19   flex-direction: row;
20   align-items: center;
21   justify-content: space-between;
22 }
23
24
25
26 }

header .logo{
27   margin-left: 5px;
28   color: var(--vermelho);
29 }
```
- Terminal:** Shows an error message: "[Error - 09:44:46] Starting client failed" and "Launching server using command deno failed."
- Bottom Bar:** Includes icons for file operations, a progress bar, and system status.

HTML e JavaScript

► Criando uma API interface NetFlix

É possível usar variáveis dentro do CSS3, para declarar uma variável dentro do CSS, nos definimos ela dentro da raiz do documento o elemento root, então podemos guardar nelas as cores que vamos utilizar, por que é interessante declarar dessa maneira e não utilizar uma uma. Porque vamos utilizar essas cores em vários lugares e se for preciso por algum motivo trocar essa cor vamos ter que precisar pesquisar pela folha de estilo inteira para sair trocando. Então uma boa prática é usar em uma variável.

EXPLORADOR ...

HTML EDITORES ABERTOS

index.html main.css style

APINETFLIXCLONE

img js

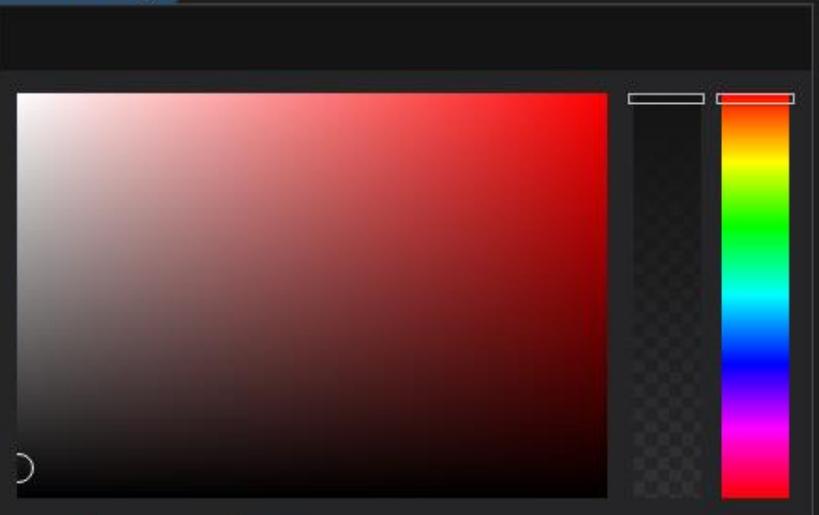
style

owl main.css responsive.css index.html sqltools_2021111...

index.html main.css

style > main.css > :root

```
:root{  
    --vermelho: #E50914;  
    --preta: #141414;  
}  
  
*{  
    margin:0;  
    padding:0;  
    box-sizing:  
}  
  
body{  
    background: var(--preta);  
    font-family: 'Arial', Times, serif;  
    color: white;  
}
```



ESTRUTURA DO CÓDIGO

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Deno Language Server

[Error - 09:44:46] Starting client failed
Launching server using command deno failed.

0 0 0 0 0:00

In 4, Col 2 (53 selecionado) Espaços: 4 UTF-8 LF CSS Port : 5500 Node v13.14.0

PT

HTML e JavaScript

► Criando uma API interface NetFlix

Como eu sei que o site tem uma cor de fundo definida eu posso declarar um **background:**, posso passar também diversas propriedades.

Observe na próxima tela que declaramos em uma variável **var**

Outra coisa que declaramos são as fontes do nosso projeto, família de fontes que vou utilizar. Você poderia usar fontes do google fontes por exemplo.

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

► O conselho é deixar essas sofisticações para depois para diminuir o nível de preocupação. Trabalhe com coisas simples por enquanto e depois na refatoração você pode sofisticar.

HTML e JavaScript

► Criando uma API interface Netflix

The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** Arquivo, Editar, Seleção, Ver, Acessar, Executar, Terminal, Ajuda.
- Title Bar:** main.css - apiNetFlixClone - Visual Studio Code.
- Explorer:** Shows "EDITORES ABERTOS" with "index.html" and "main.css" selected. It also lists a folder "APINETFLIXCLONE" containing "img", "js", and "style" subfolders, with "main.css" and "responsive.css" inside "style".
- Editor:** The "main.css" file is open, showing CSS code for styling the body and header container. The "background" property for the body is currently being typed as "background: var(--preta);".
- Bottom Status Bar:** Deno Language Server, PROBLEMAS, SAÍDA, CONSOLE DE DEPURAÇÃO, TERMINAL, [Error - 09:44:46] Starting client failed, Launching server using command deno failed.

HTML e JavaScript

► Criando uma API interface NetFlix

Declarando o Logo dentro de uma classe eu posso acertar detalhes como renderizar sempre pela esquerda e os menus sempre pelo lado direito.

É sempre bom ter uma referência visual de um sistema pronto.

Se o seu sistema é muito novo e você não tem nada parecido crie uma capa em uma ferramenta como figma ou adobe então depois disso você vai usar a sua referencia visual para poder codificar.

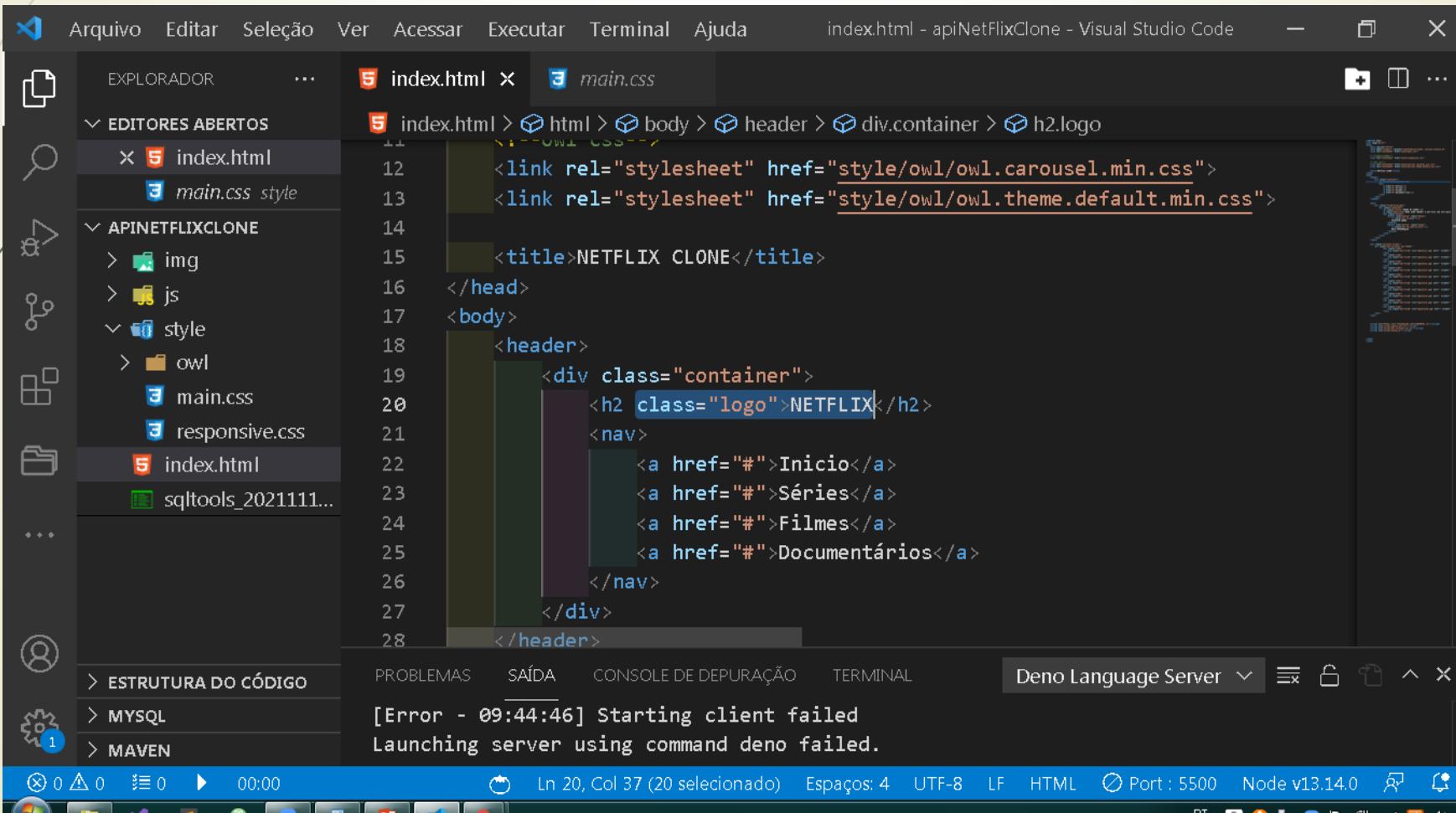
<https://www.figma.com/figjam/>

<https://www.adobe.com/br/products/xd.html>

HTML e JavaScript

- Criando uma API interface Netflix

No arquivo **index.html** eu declaro a **classe logo**



The screenshot shows the Visual Studio Code interface with the following details:

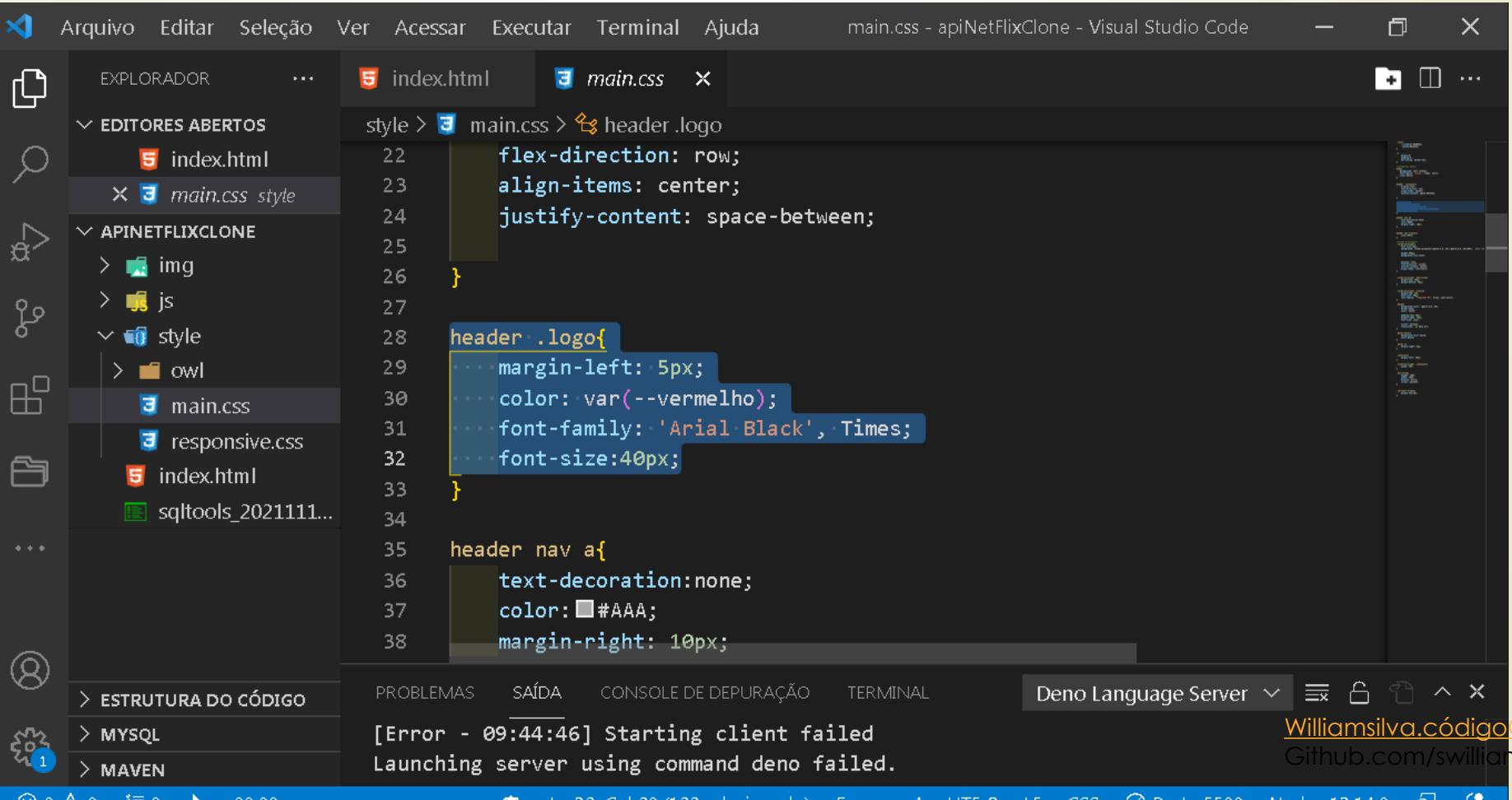
- File Explorer:** Shows the project structure with files like `index.html`, `main.css`, `img`, `js`, and `style/owl/main.css`.
- Editor:** The `index.html` file is open, displaying HTML code. A specific line of code is highlighted:

```
<h2 class="logo">NETFLIX</h2>
```
- Terminal:** At the bottom, the terminal shows an error message:

```
[Error - 09:44:46] Starting client failed  
Launching server using command deno failed.
```

HTML e JavaScript

- Criando uma API interface NetFlix
no arquivo **main.css** eu declaro a **class .logo**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like index.html, main.css, img, js, style, owl, responsive.css, index.html, and sqltools_20211111... under the APINETFLIXCLONE folder.
- Editor:** The main.css file is open, displaying the following CSS code:

```
style > main.css > header .logo
22   flex-direction: row;
23   align-items: center;
24   justify-content: space-between;
25 }
26
27
28 header .logo{
29   margin-left: 5px;
30   color: var(--vermelho);
31   font-family: 'Arial Black', Times;
32   font-size: 40px;
33 }
34
35 header nav a{
36   text-decoration: none;
37   color: #AAA;
38   margin-right: 10px;
```
- Terminal:** Shows an error message: [Error - 09:44:46] Starting client failed Launching server using command deno failed.
- Status Bar:** Shows the file is In 32 Col 20 (122 selected), Espacos: 4, UTF-8, LF, CSS, Port: 5500, Node v13.14.0.
- Bottom Right:** Includes the author's email (Williamsilva.codigo@gmail.com) and GitHub link (Github.com/swilliamsilva).

HTML e JavaScript

► Criando uma API interface NetFlix

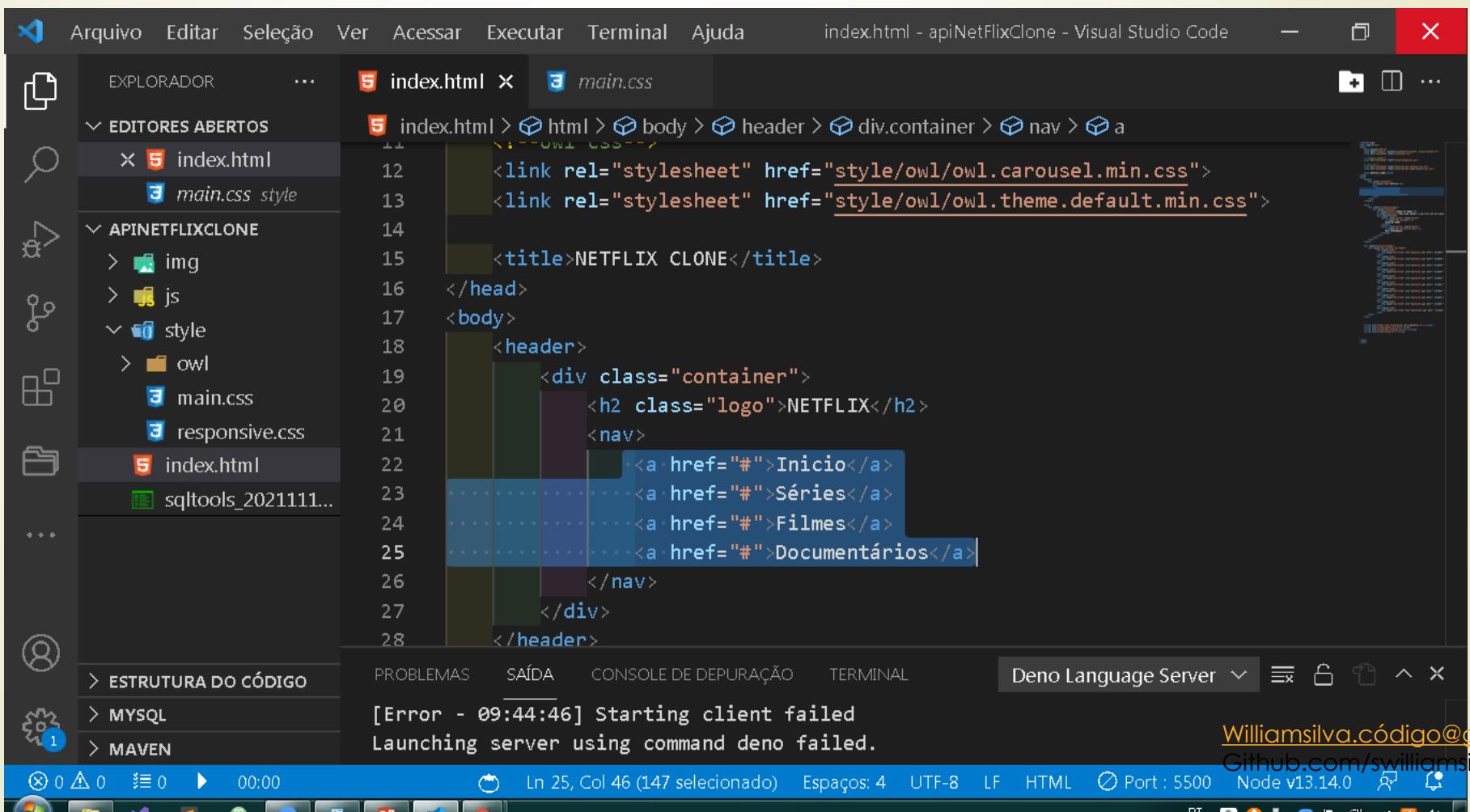
Seu o meu fundo é escuro o ideal é que minhas fontes seja claras para poder constratar, porque assim eu consigo visualizar elas de uma maneira melhor.

Para isso vamos acessar a <header> e dentro dele o elemento <navbar> e vamos acessar os nossos links

HTML e JavaScript

► Criando uma API interface Netflix

No index.html



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "EDITORES ABERTOS".
 - index.html
 - main.css
 - APINETFLIXCLONE
 - img
 - js
 - style
 - owl
 - main.css
 - responsive.css
 - index.html
 - sqltools_2021111...
- Code Editor (Center):** Displays the content of index.html.

```
index.html > html > body > header > div.container > nav > a
<link rel="stylesheet" href="style/owl/owl.carousel.min.css">
<link rel="stylesheet" href="style/owl/owl.theme.default.min.css">

<title>NETFLIX CLONE</title>
</head>
<body>
  <header>
    <div class="container">
      <h2 class="logo">NETFLIX</h2>
      <nav>
        <a href="#">Inicio</a>
        <a href="#">Séries</a>
        <a href="#">Filmes</a>
        <a href="#">Documentários</a>
      </nav>
    </div>
  </header>
```
- Bottom Status Bar:** Shows file status (0 errors, 0 warnings), line and column information (Ln 25, Col 46), and terminal output.

```
[Error - 09:44:46] Starting client failed
Launching server using command deno failed.
```
- Bottom Right:** Includes the GitHub URL (williamsilva.codigo@gmail.com) and the Node.js version (v13.14.0).

JavaScript

- Criando uma API interface Netflix

No main.css fazemos a decoração da nav bar.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "EDITORES ABERTOS". The "main.css" file is selected in the "style" folder.
- Editor:** Displays the CSS code for the "main.css" file, specifically targeting the navigation bar.

```
style > main.css > header nav a
30   color: var(--vermelho);
31   font-family: 'Arial Black', Times;
32   font-size:40px;
33 }
34
35 header nav a{
36   text-decoration:none;
37   color: #AAA;
38   margin-right: 10px;
39 }
40
41 header nav a:hover{
42   color: #fff;
43 }
44
45 /*filme principal*/
```
- Terminal:** Shows an error message: "[Error - 09:44:46] Starting client failed" and "Launching server using command deno failed."

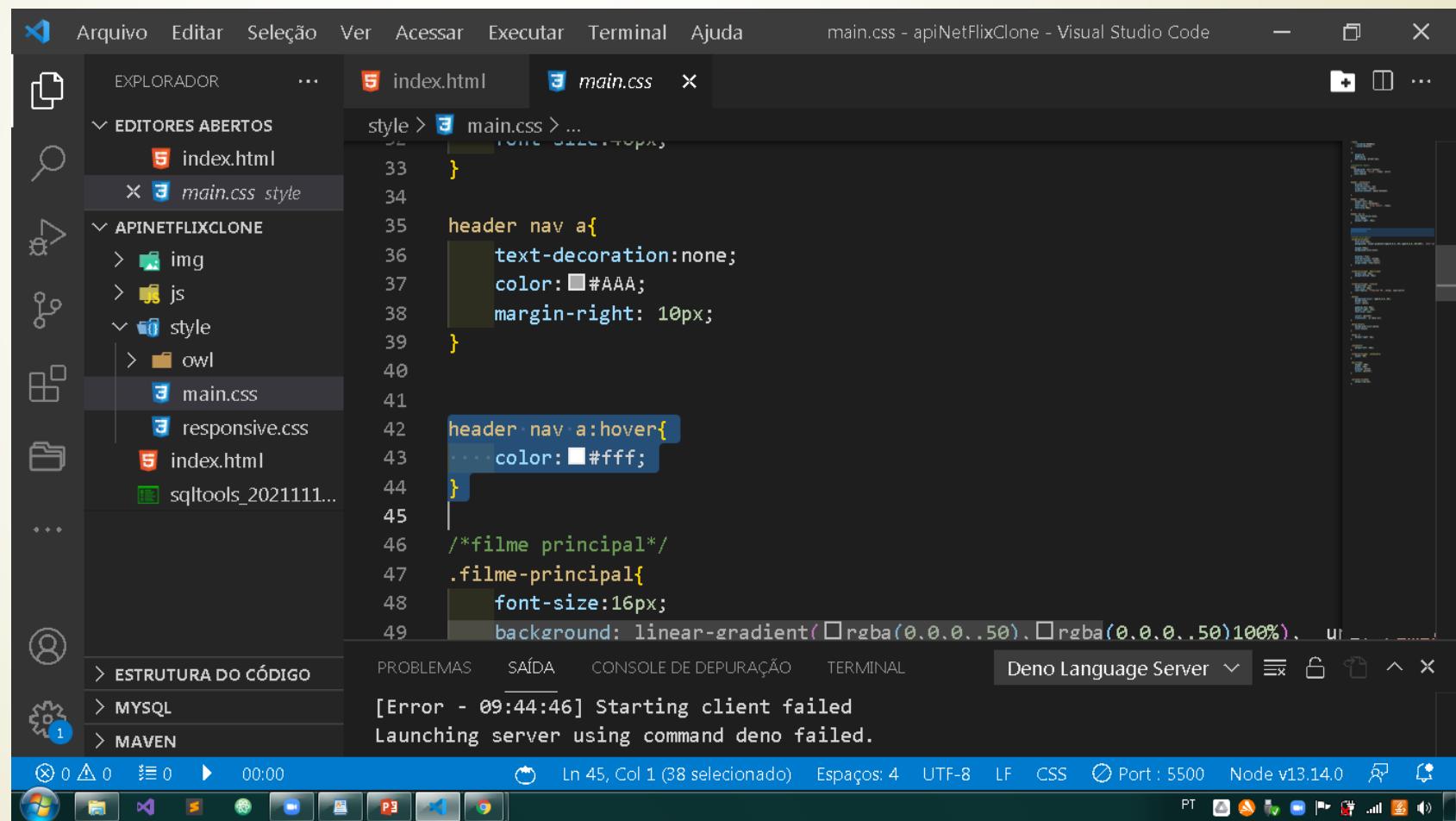
HTML e JavaScript

► Criando uma API interface NetFlix

Colocando o margin ele já guarda um distanciamento entre as palavras e quando passarmos o mouse sobre a letra gostaríamos que elas mudassem a cor para chamar a atenção e para isso pegamos um evento específico que é quando o mouse estiver em cima da palavra do nosso elemento de navegação

JavaScript

- ▶ Criando uma API interface NetFlix
usando o hover para mudar a cor quando passar o mouse sobre a palavra.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "EDITORES ABERTOS" (Open Editors) section: index.html, main.css (selected), and responsive.css. In the "APINETFLIXCLONE" folder, there are img, js, style (with main.css and owl), and sqltools_2021111...
- Code Editor:** The main.css file contains the following CSS code:

```
header nav a{  
    text-decoration:none;  
    color:#AAA;  
    margin-right: 10px;  
}  
  
header nav a:hover{  
    color:#fff;  
}  
  
/*filme principal*/  
.filme-principal{  
    font-size:16px;  
    background: linear-gradient(rgba(0,0,0,.50),rgba(0,0,0,.50)100%);  
    color:white;  
    padding: 10px;  
    border-radius: 10px;  
    text-align: center;  
    width: fit-content;  
    margin: auto;  
}
```
- Terminal:** Shows an error message: "[Error - 09:44:46] Starting client failed".
- Status Bar:** Displays file status (0 changes), line 45, column 1 (38 selected), and other system information like Node v13.14.0.

HTML e JavaScript

► Criando uma API interface NetFlix

Agora vamos posicionar esses elementos na tela, lembra que criamos uma caixa que é o nosso container.

Para isso vamos pegar o nosso header container e vamos falar que ele é um display do tipo , isso vai fazer a gente a utilizar um flex box que é uma maneira de você alinhar os elementos na tela, Toda vez que você faz isso ele tem uma tendência de jogar tudo para a linha, usando uma propriedade chamada flex-direction: row, se eu não declarar ela ele vai executar do mesmo jeito.

Outra coisa é alinhar os elementos mais para o centro e para fazer isso vamos usar o align-items: center; ou seja, os elementos ficaram centralizados porem eu preciso ter um espaço entre a tag h2 e a caixa de seletor nav.

Para isso usamos a propriedade justify-content: space-between ele vai trabalhar alinhado horizontal

HTML

Java Script

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda main.css - apiNetFlixClone - Visual Studio Code

EXPLORADOR ... index.html main.css X

EDITORES ABERTOS

index.html main.css style

APINETFLIXCLONE

img js

style

owl main.css responsive.css index.html

sqltools_2021111...

style > main.css > header .container

```
14 background: var(--preta);
15 font-family: 'Arial', Times, serif;
16 color: white;
17 }
18
19
20 header .container{
21   display:flex;
22   flex-direction: row;
23   align-items: center;
24   justify-content: space-between;
25 }
26
27
28 header .logo{
29   margin-left: 5px;
30   color: var(--vermelho);
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL Deno Language Server

[Error - 09:44:46] Starting client failed
Launching server using command deno failed.

0 0 0 0 0:00 Ln 20, Col 1 (122 selecionado) Espaços: 4 UTF-8 LF CSS Port: 5500 Node v13.14.0

PT



EXPLORADOR

...

index.html

main.css

X

+

□

...



HTML

EDITORES ABERTOS

index.html

X main.css style



...



APINETFLIXCLONE

> img

> js

> style

> owl

main.css

responsive.css

index.html

sqltools_2021111...

style > main.css > header .container

background: var(--preta);
font-family: 'Arial', Times, serif;
color: white;
}

header .container{

display:flex;
flex-direction: row;
align-items: center;
justify-content: space-between;

}

header .logo{

margin-left: 5px;
color: var(--vermelho);

}

> ESTRUTURA DO CÓDIGO

> MYSQL

> MAVEN

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

Deno Language Server

▼

☰

🔒

✖

^

✖

[Error - 09:44:46] Starting client failed

Launching server using command deno failed.

0 ▲ 0 ⚙ 0 00:00



Ln 20, Col 1 (122 selecionado)

Espaços: 4

UTF-8 LF

CSS

Port 55000 Node v15.14.0
Github.com/swilliamsilva

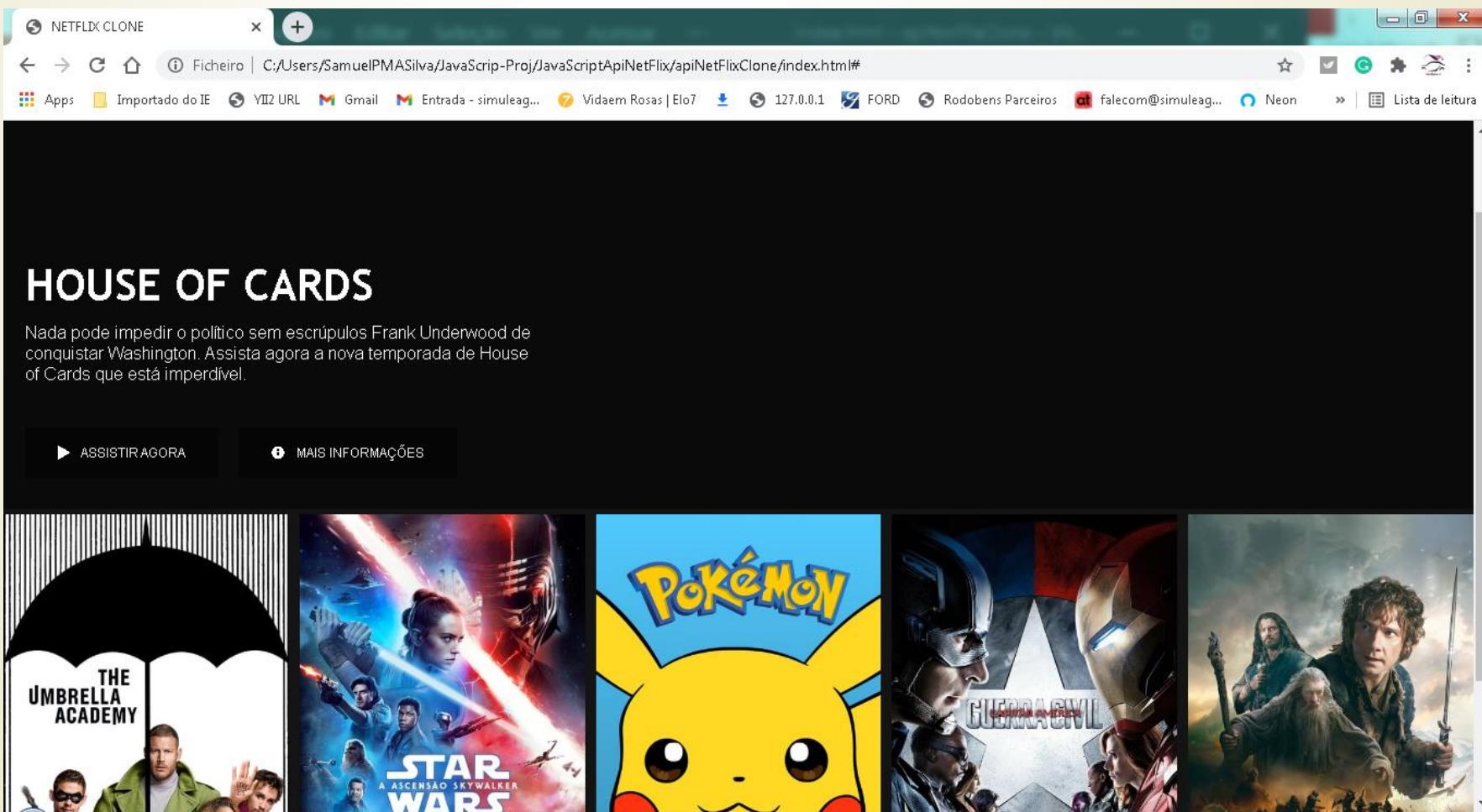
PT



HTML e JavaScript

► Construindo o corpo da aplicação

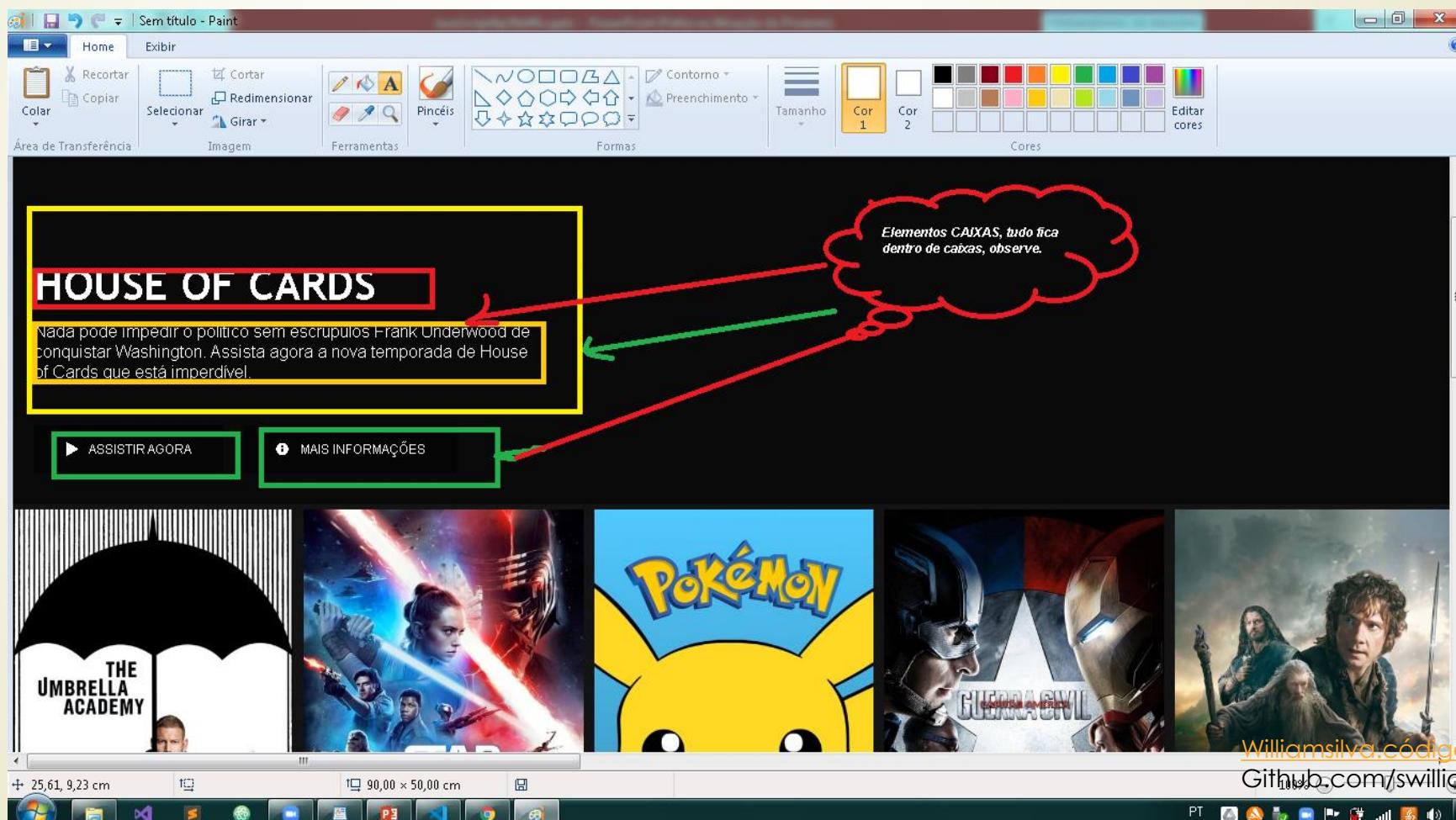
Agora vamos fazer o miolo do site que é esta tela onde temos um cartaz.



HTML e JavaScript

- Construindo o corpo da aplicação

Caixas contem as informações que você precisa para trabalhar



Williamsilva.código@gmail.com

Github.com/swilliamsilva

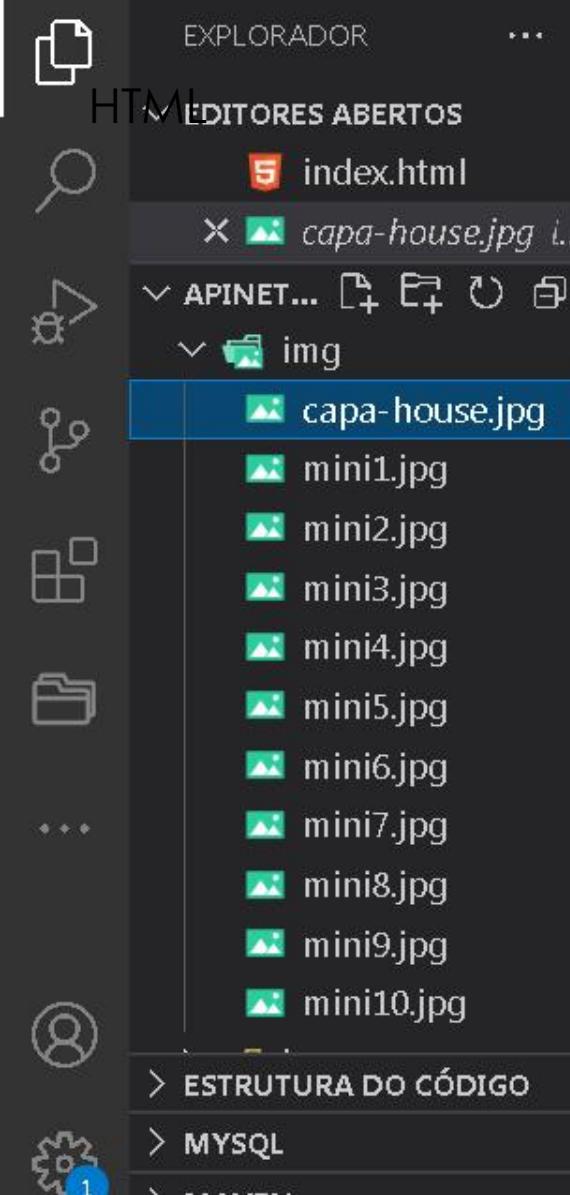
HTML e JavaScript

- Construindo o corpo da aplicação

As imagens você pode pegar do site:

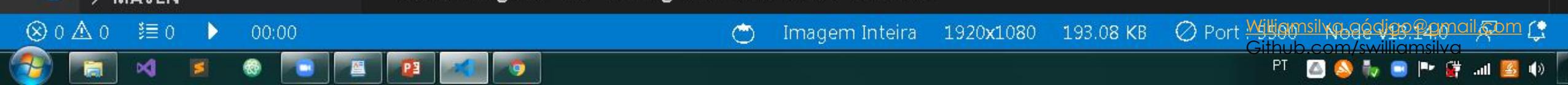
<https://www.themoviedb.org/>

Escolha as imagens que lhe agrada para por no site e copie para a **pasta img**



PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

[Error - 09:44:46] Starting client failed
Launching server using command `demo failed`



HTML e JavaScript

► Construindo o corpo da aplicação

O miolo do nosso site vai ficar dentro de uma **tag semântica** chamada `<main>`

Que o sistema vai entender que é a classe principal.

Nós vamos trabalhar nas caixas que estão dentro delas que são tag s filhas chamadas `<div>`

`<div class="filme-principal">`

Também vamos colocar uma classe chamada container, uma classe wrapper, que são `<div>` que fazemos para encaixar tudo ali dentro. Se precisar alterar alguma coisa você vai mexer apenas nessa classe wrapper.

EXPLORADOR ...

HTML EDITORES ABERTOS

index.html capa-house.jpg

APINETFLIXCLONE

img

- capa-house.jpg
- mini1.jpg
- mini2.jpg
- mini3.jpg
- mini4.jpg
- mini5.jpg
- mini6.jpg
- mini7.jpg
- mini8.jpg
- mini9.jpg
- mini10.jpg

ESTRUTURA DO CÓDIGO

MYSQL

MAVEN

index.html > html > body > main > div.filme-principal

```
27   </div>
28   </header>
29
30 <main>
31   <div class="filme-principal">
32     <div class="container">
33       <h3 class="titulo">HOUSE OF CARDS</h3>
34       <p class="descricao">Nada pode impedir o político sem escrúpu</p>
35       <div class="botoes">
36         <button role="button" class="botao">
37           <i class="fas fa-play"></i>
38           ASSISTIR AGORA
39         </button>
40         <button role="button" class="botao">
41           <i class="fas fa-info-circle"></i>
42           MAIS INFORMAÇÕES
43         </button>
44       </div>
45     </div>
46   </main>
47
48
```

Ln 31, Col 38 (30 selecionado) Espacos: 4 UTF-8 LF HTML Port WilliamSilva@ádico...@gmail.com Github.com/swilliamsilva

PT

HTML e JavaScript

► Construindo o corpo da aplicação

As tag herdam todas as configurações que o elemento pai tem.

O pai do <h3> é o <div> e o pai do <div> é o elemento principal o <main>

E todos estão no <body> e tudo que eu fizer no <body> vai influenciar nos filhos.

Outra coisa que temos na descrição do filme é uma tag chamada <p> de paragrafo. E vamos referenciar no css com o nome de **decricao. Não use caracteres acentuados ou cedilha nos nomes de class.**

JavaScript

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda index.html - apiNetFlixClone - Visual Studio Code

EXPLORADOR ...

EDITORES ABERTOS

- index.html
- main.css

APINETFLIXCLONE

- mini7.jpg
- mini8.jpg
- mini9.jpg
- mini10.jpg
- js
- style

 - owl
 - main.css
 - responsive.css

- index.html
- sqltools_2021111...

ESTRUTURA DO CÓDIGO

```
index.html > html > body > main > div.filme-principal > div.container > p.descricao
27 </div>
28 </header>
29
30 <main>
31   <div class="filme-principal">
32     <div class="container">
33       <h3 class="titulo">HOUSE OF CARDS</h3>
34       <p class="descricao">Nada pode impedir o político sem escrúpu...
35       <div class="botoes">
36         <button role="button" class="botao">
37           <i class="fas fa-play"></i>
38           ASSISTIR AGORA
39         </button>
40         <button role="button" class="botao">
41           <i class="fas fa-info-circle"></i>
42           MAIS INFORMAÇÕES
43         </button>
44       </div>
45     </div>
46   </div>
```

WilliamSilva.codigo@gmail.com
Github.com/swilliamsilva

HTML e JavaScript

► Construindo o corpo da aplicação

Toda vez que você quiser fazer um comentário no css, basta colocar uma barra para direita e depois um asterisco, insere seu comentário e depois fecha a barra.

HTML e JavaScript

- ▶ Construindo o corpo da aplicação

No main.css vamos referenciar e tratar a imagem de centro da aplicação usando na tag background a url.

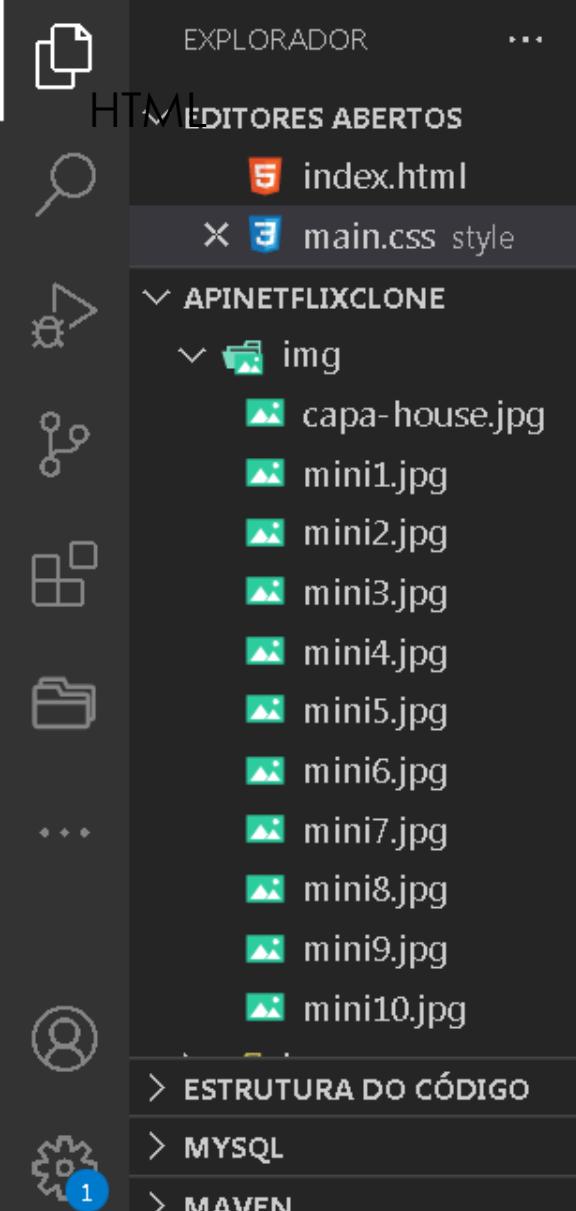
- ▶ **background: linear-gradient(rgba(0,0,0,.50),rgba(0,0,0,.50)100%), url('/img/capa-house.jpg');**

- ▶ O linear-gradiente vai criar um gradiente de cor na imagem aqui de cor escura.

- ▶ Para especificar o tamanho da imagem dentro da caixa vamos usar a tag backbround abaixo:

background-size:cover;

O cover vai pegar a área de cobertura de nossa caixa.



```
style > 3 main.css > .filme-principal
40
41
42     header nav a:hover{
43         color: #fff;
44     }
45
46     /*filme principal*/
47     .filme-principal{
48         font-size:16px;
49         background: url('/img/capa-house.jpg');
50
51         height:400px;
52         background-size:cover;
53
54
55         display: flex;
56         flex-direction: column;
57         justify-content: center;
58         align-items: flex-start;
59     }
60
61
```

HTML e JavaScript

► Construindo o corpo da aplicação

Agora vamos acrescentar alguns botões para ir um pouco além do proposto.

O primeiro botão vamos chamar de ASSISTIR AGORA

E um outro botão com o nome de MAIS INFORMAÇÕES

Dentro de uma classe padrão que vamos chamar de botão mesmo.

HTML e JavaScript

- ▶ Construindo o corpo da aplicação
`<button role="button" class="botao">`
- ▶ A função desse role é atender as necessidades de renderização em outros equipamentos que não sejam desktop.

JavaScript

- Construindo o corpo da aplicação

O display-flex sempre joga para o padrão linha.

E ele sempre vem acompanhado do flex-direction: aqui com a propriedade coluna. Isso para poder normalizar ele de uma maneira vertical.

Nosso display-flex sempre segue a direção do flex-direction.

Assim vamos centralizar no meio da coluna usando a tag.

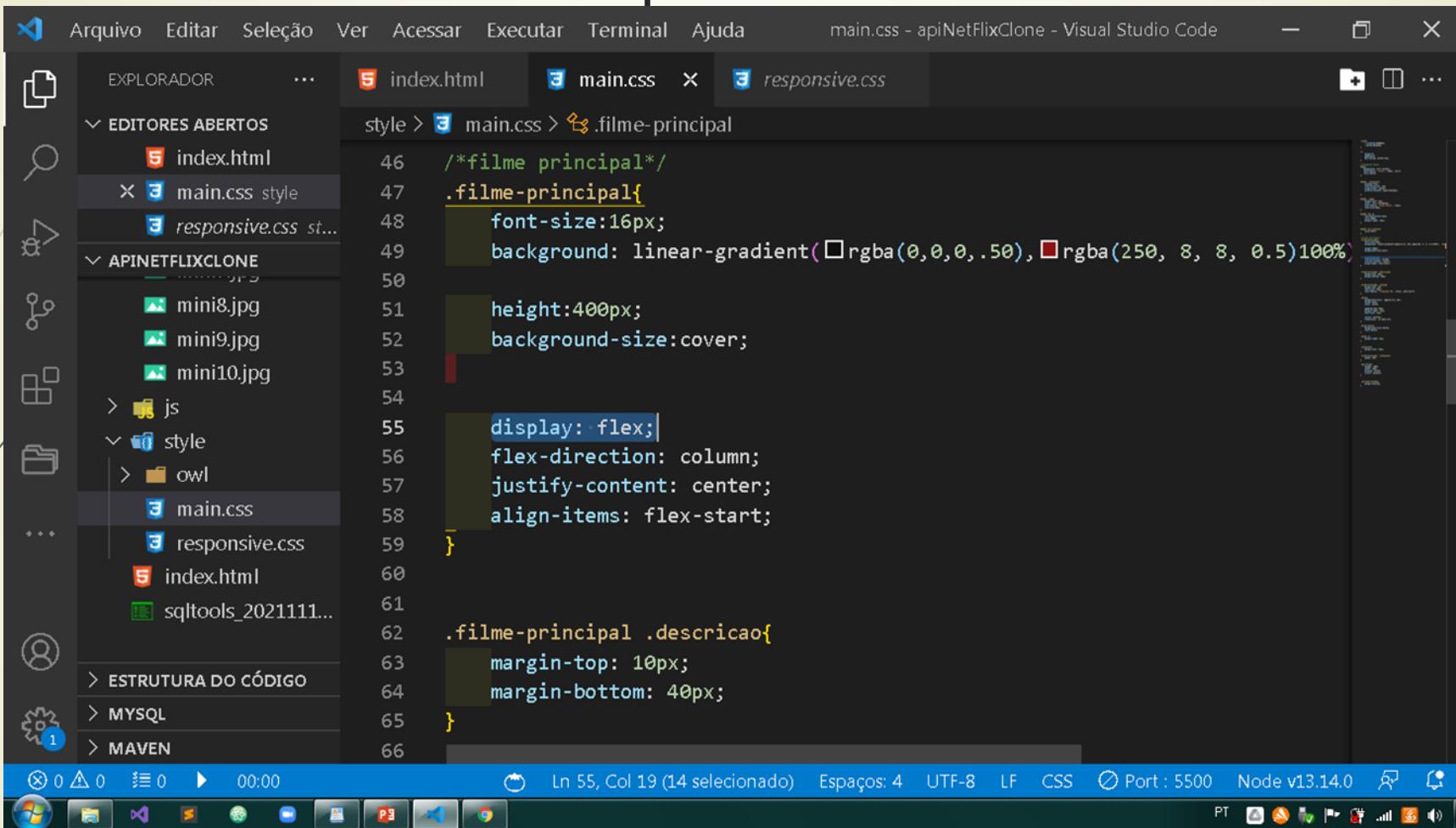
justify-content: center;

- Para que os elementos começem sempre da esquerda usamos o:

align-items: flex-start;

HTML

HTML e JavaScript



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "EDITORES ABERTOS".
 - index.html
 - main.css (selected)
 - responsive.css
 - APINETFLIXCLONE
 - mini8.jpg
 - mini9.jpg
 - mini10.jpg
 - js
 - style
 - owl
 - main.css
 - responsive.css
 - index.html
 - sqltools_2021111...
- Code Editor (Center):** Displays the CSS code for ".filme-principal".

```
style > main.css > .filme-principal
46  /*filme principal*/
47  .filme-principal{
48    font-size:16px;
49    background: linear-gradient(rgba(0,0,0,.50), rgba(250, 8, 8, 0.5)100%);
50
51    height:400px;
52    background-size:cover;
53
54
55    display: flex;
56    flex-direction: column;
57    justify-content: center;
58    align-items: flex-start;
59  }
60
61
62  .filme-principal .descricao{
63    margin-top: 10px;
64    margin-bottom: 40px;
65  }
```
- Bottom Status Bar:** Shows file status (0 changes), time (00:00), line (Ln 55, Col 19 (14 selecionado)), and other settings like spaces (4), encoding (UTF-8), and port (Port : 5500 Node v13.14.0).

HTML e JavaScript

► Construindo o corpo da aplicação

Toda vez que você cria uma <div> ela sempre vem por padrão com display-block.

Alguma propriedades do css mesmo que você não declare elas veem por padrão então por essa razão seu site pode ter comportamento estranho por essa razão é importante declarar. Como o flex-direction.

HTML e JavaScript

- Construindo o corpo da aplicação

HTML e JavaScript

- ▶ Construindo o corpo da aplicação
O botão.

Ele já vem com um estilo pré definido, já vem com uma cor, tem um RGB e elementos que a gente não definiu. Isso porque ele já vem por padrão definido.

HTML e JavaScript

- Construindo o corpo da aplicação

Font Awesome

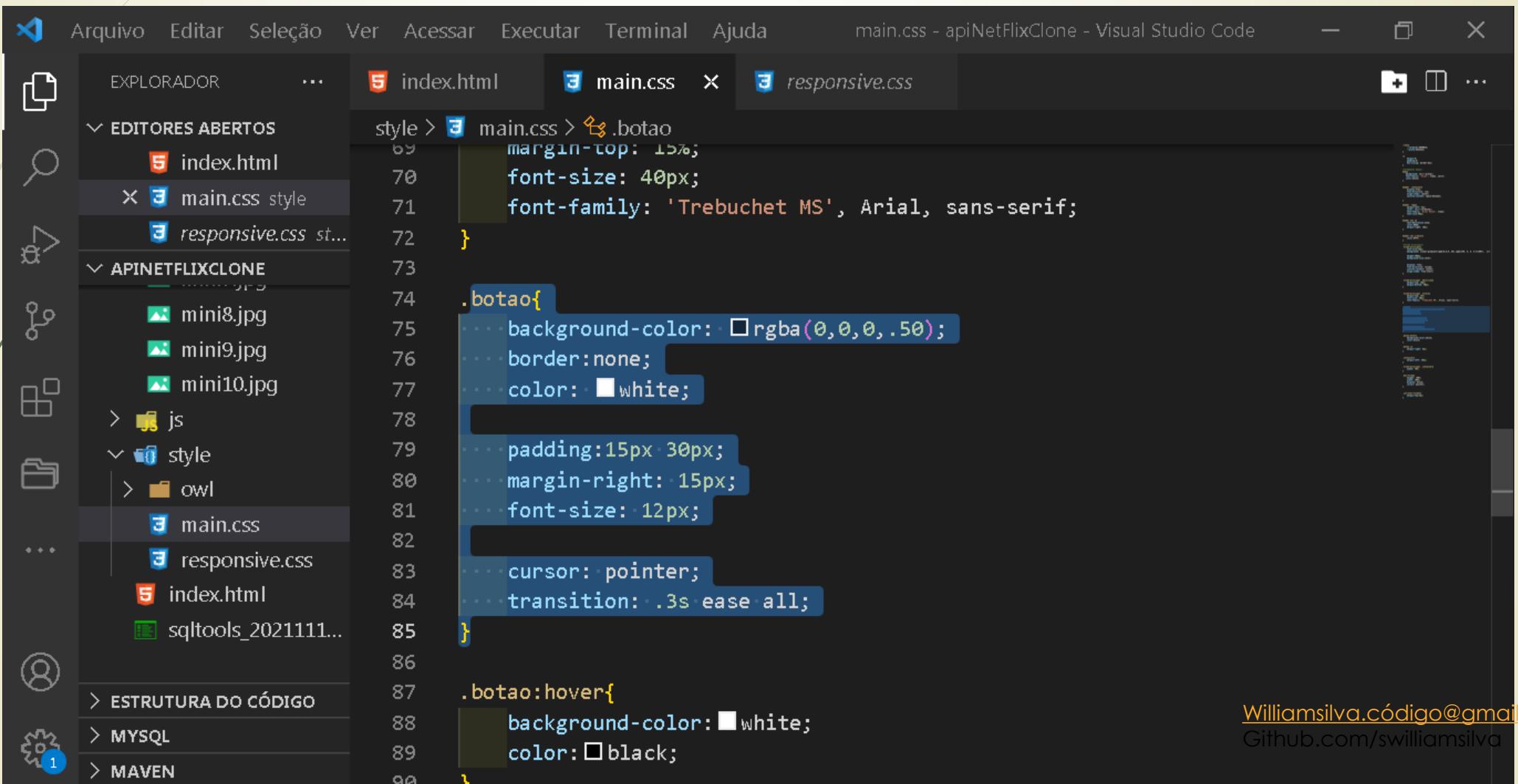
<https://fontawesome.com/v5.15/icons?d=gallery&p=2>

Eles são fontes e ícones já pré definidos que você pode aplicar no seu trabalho conferindo muito mais qualidade.

Ele permite você criar ícones dentro do HTML sem você ter essas imagens.

JavaScript

► Formatando o botão no main.css



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** Arquivo, Editar, Seleção, Ver, Acessar, Executar, Terminal, Ajuda.
- Title Bar:** main.css - apiNetFlixClone - Visual Studio Code
- Explorer:** Shows the project structure under "EDITORES ABERTOS":
 - index.html
 - main.css style
 - responsive.css st...
 - APINETFLIXCLONE
 - mini8.jpg
 - mini9.jpg
 - mini10.jpg
 - js
 - style
 - owl
 - main.css
 - responsive.css
 - index.html
 - sqltools_2021111...
- Code Editor:** The "main.css" tab is active, displaying the following CSS code:

```
style > main.css > .botao
69 margin-top: 15px;
70 font-size: 40px;
71 font-family: 'Trebuchet MS', Arial, sans-serif;
72 }

73 .botao{
74   background-color: #rgba(0,0,0,.50);
75   border:none;
76   color: white;
77   padding:15px 30px;
78   margin-right: 15px;
79   font-size: 12px;
80   cursor: pointer;
81   transition: .3s ease all;
82 }

83 .botao:hover{
84   background-color: white;
85   color: black;
```
- Bottom Right:** Includes the author's email (Williamsilva.codigo@gmail.com) and GitHub link (Github.com/swilliamsilva).

HTML e JavaScript

- Formatando a cor no padrão RGBA

background-color: rgba(153, 111, 111, 0.5);

R = Vermelho

G= Verde

B=Blue

A= Transparência que é a camada alfa onde você informa em percentual.

Pode declarar por exemplo 50%

HTML e JavaScript

► Formatando o Botão

padding:15px 30px;

A padding lida com o espaço interno do botão.

Quando eu informo somente um valor, por exemplo:

padding: 15px; ele se refere as quatro posições.

Quando eu passo dois valores no caso 15px e 30px ele sempre vai olhar no sentido horário, sendo que o primeiro valor para cima e em baixo e o segundo valor será para os dois lados, direita e esquerda.

Se eu passar 4 valores eu estarei passando um para cima, para direita, baixo e esquerda.

Formatando o botão

- ▶ Formatando o botão

margin-right: 15px;

Define a margem direita, poderia ser margin-left, é o que vai definir o espaço entre os botões

margin-top: 15px por exemplo empurra tudo para baixo.

Sempre é mais vantajoso trabalhar com percentual. Ele vai fazer um calculo para você de como ele vai posicionar os elementos em tela, independente da altura ou da largura. Isso é importante em um site responsivo para trabalhar com mobile.

margin-top: 15%

HTML e JavaScript

- Redefinindo o titulo principal

font-family: 'Trebuchet MS', Arial, sans-serif;

Normalmente ele envia como sugestão mas você pode trabalhar com qualquer outra fonte.

O navegador vai buscar sempre o nome da fonte que estiver entre aspas porém se não encontrar ele vai usar as alternativas similares como Arial, sans-serif.

font-size: 12px;

Define o tamanho da fonte.

HTML e JavaScript

- UX, experiência com o usuário

cursor: pointer;

Quando você passa o mouse sobre o botão ele muda de cor.

É importante mudar o cursor para um ponteiro para quando o usuário passar o mouse sobre o botão ele mostrar que é uma parte clicável do botão.

transition: .3s ease all;

Ele cuida da parte da transição de uma animação ou outra e sua configuração é a seguinte

.3s = 3 segundos, é um tempo para sair de uma animação e provocar outra.

Ease all = tipo da animação, suavizada,

HTML e JavaScript

- Utilizando ícones na aplicação usando o Font Awesome

Basta acessar o site:

- <https://fontawesome.com/>

Clique em START FOR FREE, ele vai pedir para você se cadastrar, e quando você colocar o seu email ele vai te mandar um link para você só copiar e colar na sua página html e você já vai poder utilizar esses ícones.

- É um link para um arquivo tipo JS, JavaScript e nesse caso vamos estar referenciando um link externo que esta na nuvem.

HTML e JavaScript

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Kit: 52f6d687a7 | Font Awesome". The page displays a dark-themed interface for managing a Font Awesome kit. At the top, there's a navigation bar with links for "How to Use", "Settings", and "Uploaded Icons". A prominent green callout box in the center says "Welcome to your new Font Awesome Account + Kit!". It provides instructions: "Your kit and all its icon goodness is ready to use right now! To get started, follow the steps below, or you can jump in and adjust your kit settings to add the sites you'll use it on." Step 1, "Add Your Kit's Code to a Project", is highlighted with a large number "1". Below it is a code snippet:

```
<script src="https://kit.fontawesome.com/52f6d687a7.js" crossorigin="anonymous"></script>
```

. To the right of the code is a blue button labeled "Copy Kit Code!". At the bottom, there's a note: "Copy your kit's code into the <head> of each template or page in your project where you want to use Font Awesome via this kit. If you've got other versions or flavors of Font Awesome already in use, make sure to remove them, mmkay?". To the right of this note is a button labeled "Show Me Exactly Where to Put This Kit Code". The browser's toolbar at the bottom includes icons for file operations, search, and other common functions.



EXPLORADOR



5 index.html ×



main.css



responsive.css



HTM ED

EDITORES ABERTOS

- X  index.html
E  main.css style
E  responsive.css st...



5 index.html > html > body > script

```
75
76     <div class="item">
77         
79     <div class="item">
80         
82 </div>
83
84
85
86 <script src="https://kit.fontawesome.com/2c36e9b7b1.js"></script>
87 <script src="js/owl/jquery.min.js"></script>
88 <script src="js/owl/owl.carousel.min.js"></script>
89 <script src="js/owl/setup.js"></script>
90
91
92
93 </body>
94 </html>
```

⊗ 0 △ 0 ≡ 0 ► 00:00



Ln 86, Col 70 (65 seleccionado)

Espaços: 4

UTF-8

LF HT

ML

Port: 5500 Node.js 14.0 Github.com/swilliamsilva



HTML e JavaScript

- ▶ Link externo para font awesome

```
<script src="https://kit.fontawesome.com/2c36e9b7b1.js"></script>
```

- ▶ Como vê é um link javascript

HTML e JavaScript

► Trabalhando com font awesome

Dentro do site na seção de ícones, tem uma galeria de ícones, tem alguns ícones pagos e tem os free. Pesquise por exemplo um botão de play.

Toda vez que você clica em um ícone ele já te da a opção COPIAR HTML então é só copiar e colar dentro do nosso site.

Dentro do nosso botão.

```
<i class="fas fa-play"></i>
```


HTML e JavaScript

► A classe Wrapper

O que diferencia os bons desenvolvedores dos medianos é que os medianos se contentam em copiar o código sem saber para que.

Por isso é importante a teoria.

Um bom desenvolvedor entende a fundo como uma coisa funciona.

HTML e JavaScript

► A classe wrapper

Muitas coisas que você faz com o CSS não tem uma propriedade pronta.

Não é um código que você digita é mais conceitual, são truques que você faz com CSS.

Wrapper significa envelopar ou envelope

Que é justamente o objetivo de fazer uma classe wrapper que é o que chamamos container.

Existem desenvolvedores que fazem distinção entre wrapper e containers.

O objetivo dessa classe é envelopar tudo que está dentro dela.

Éla é uma <div> que guarda vários elementos dentro dela.

HTML e JavaScript

► A classe wrapper

O primeiro motivo para se criar uma classe assim é que fica mais fácil de manipular os elementos filhos ao invés de manipular um elemento filho de cada vez, você só faz com elementos pai. Pois os filhos são diretamente influenciados pelos pais. Se o pai recebe um determinado tratamento, obviamente os filhos recebem o mesmo tratamento.

São classes que envelopam as subclasses para você trabalhar com os elementos como um todo.

Se você for no bootstrap você vai ver que ele trabalha com classes containers oferecendo vários modelos.

<https://getbootstrap.com/docs/5.1/layout/containers/>

Tudo que estiver envelopado dentro de um container vai andar junto com ele.

HTML e JavaScript

► Construindo o carrossel.

Existe **um plug-ins** JQuery que se chama **owl.carousel.js**

<https://owlcarousel2.github.io/OwlCarousel2/>

Ainda é bem utilizado em muitos projetos herdados e por isso é importante conhecer. É apenas uma ferramenta.

Ele é um repositório que vai facilitar a sua vida para criar caroucel.

Clique em Download e baixe um zip e descompacte, você verá uma série de arquivos. Isto é uma série de demonstrações para você estudar e utilizar.

owlcarousel2.github.io/OwlCarousel2/

HTML owl.carousel.js

Home Demos Docs Download

Owl Carousel 2

Touch enabled jQuery plugin that lets you create a beautiful responsive carousel slider.

Download Github

2.3.4



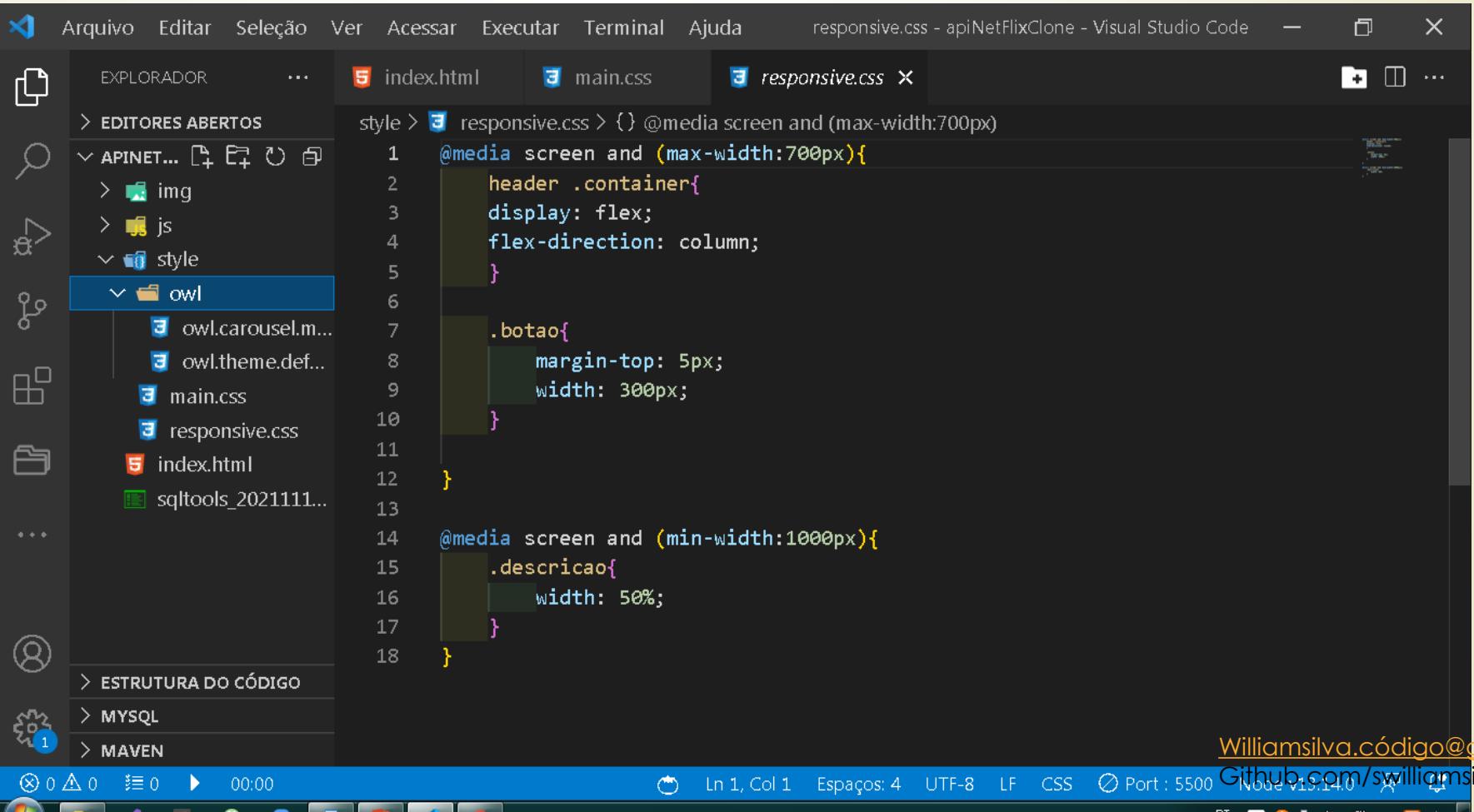
Demo

Williamsilva.codigo@gmail.com
Github.com/swilliamsilva

HTML e JavaScript

► Construindo o carrossel.

Dentro da **pasta Styles** crie uma pasta chamada de **owl**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with a folder named "owl" inside "style". Other files visible include "index.html", "main.css", and "responsive.css".
- Code Editor:** The "responsive.css" file is open, displaying the following CSS code:

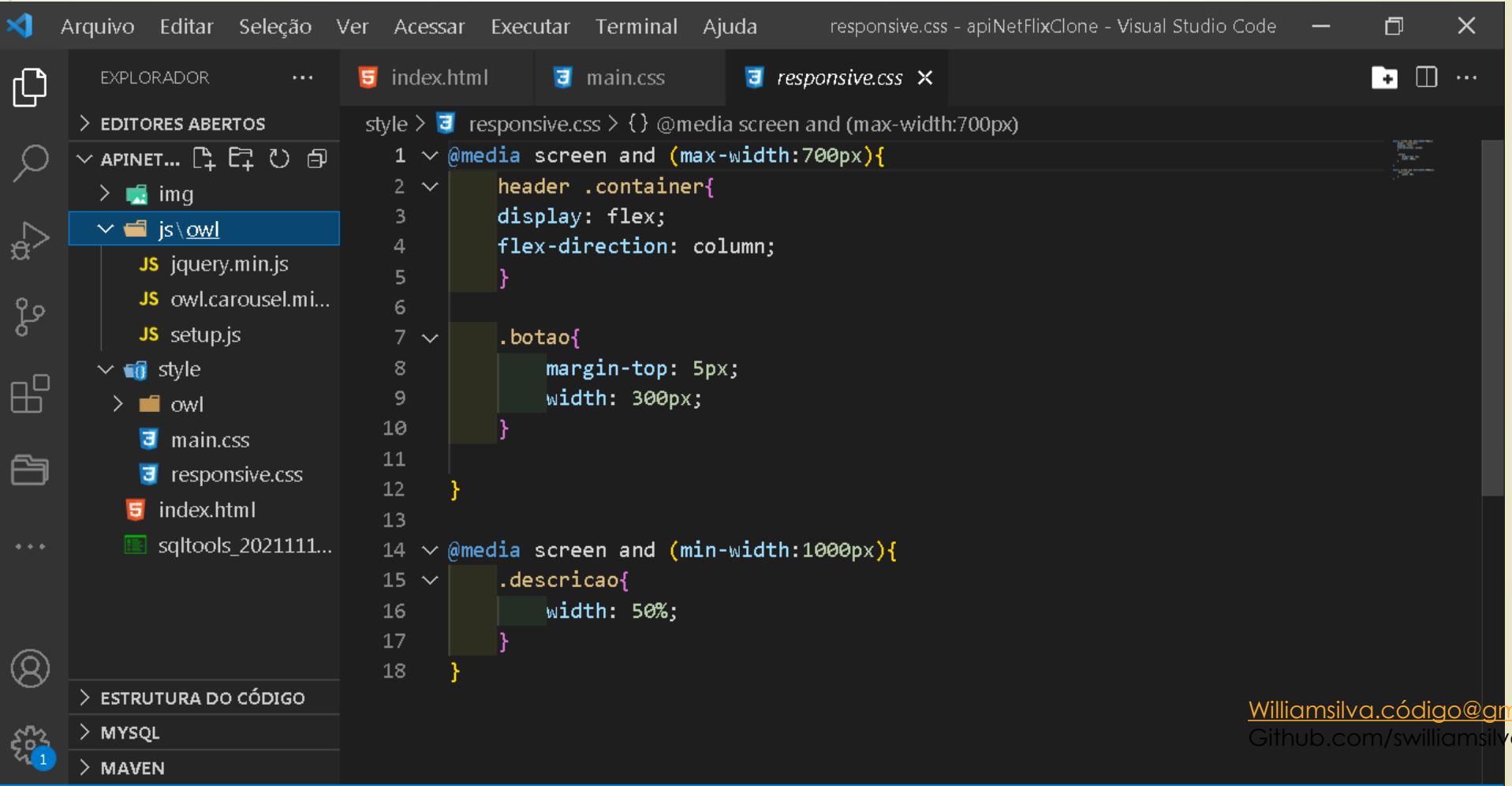
```
style > responsive.css > {} @media screen and (max-width:700px){  
1  @media screen and (max-width:700px){  
2      header .container{  
3          display: flex;  
4          flex-direction: column;  
5      }  
6  
7      .botao{  
8          margin-top: 5px;  
9          width: 300px;  
10     }  
11 }  
12  
13  
14 @media screen and (min-width:1000px){  
15     .descricao{  
16         width: 50%;  
17     }  
18 }
```

The status bar at the bottom shows the following information: Line 1, Column 1, Spaces: 4, UTF-8, LF, CSS, Port: 5500, Node.js v13.14.0.

JavaScript

► Construindo o carrossel.

E dentro da **pasta js** crie também uma pasta chamada **owl**



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure:
 - APINET...
 - img
 - js\owl (selected)
 - jquery.min.js
 - owl.carousel.mi...
 - setup.js
- Editors:** Three files are open:
 - index.html
 - main.css
 - responsive.css
- Code Editor:** The responsive.css file contains the following CSS code:

```
style > responsive.css > {} @media screen and (max-width:700px){  
1  @media screen and (max-width:700px){  
2    header .container{  
3      display: flex;  
4      flex-direction: column;  
5    }  
6  
7    .botao{  
8      margin-top: 5px;  
9      width: 300px;  
10 }  
11 }  
12 }  
13  
14 @media screen and (min-width:1000px){  
15   .descricao{  
16     width: 50%;  
17   }  
18 }
```

HTML e JavaScript

► **Construindo o carrossel.**

Dentro da pasta onde você extraiu os arquivos jQuery tem uma pasta chamada docs e dentro de docs você tem uma pasta chamada **assets** e dentro dela os algumas outras pastas.

Dentro da pasta vendor temos uma coleção de arquivos e o que nos interessa é o jquery em si. Que vamos copiar e colar de OWL

jquery.min.js

Dentro de docs vamos dentro de assets e vemos varias pasta mas o que interessa é a pasta owl e dentro dela:

owl.carousel.min.js

Assim pegamos esses dois arquivos e copiamos para dentro do nosso projeto.

Construindo o Carrossel

► Construindo o carrossel.

Da mesma maneira que a gente pegou **nosso JS** o **nosso CSS** também vai estar **dentro da pasta owl**

Dentro dessa pasta temos uma pasta de distribuição **chamada dist** e dentro dela também tem **o assets** e daqui vamos precisar **do owl** e do **tema então** vamos pegar e copiar para dentro do nosso projeto os arquivos.

owl.carousel.min.css

e

owl.theme.default.min.css

Ficando como esta no próximo slide

EXPLORADOR

HTML EDITORES ABERTOS

- APINETFLIXCLONE
 - img
 - js\owl
 - style
 - owl
 - owl.carousel.min.css
 - owl.theme.default.min.css
 - main.css
 - responsive.css
 - index.html
 - sqltools_20211112093828_2464.log

ESTRUTURA DO CÓDIGO

MYSQL

MAVEN

0 ▲ 0 ⚙ 0 ▶ 00:00

Ln 1, Col 1 Espaços: 4 UTF-8 LF {} JavaScript Port: 5500 Node v13.14.0

PT

index.html main.css jquery.min.js

js > owl > JS jquery.min.js > ...

```
length,s=!n;o<a;o++)(r!=t(e[o],o))!==s&&i.push(e[o]);return i
,map:function(e,t,n){var r,i,o=0,s=[];if(C(e))for(r=e.length;
o<r;o++)null!=(i=t(e[o],o,n))&&s.push(i);else for(o in e)
null!=(i=t(e[o],o,n))&&s.push(i);return a.apply([],s)},guid:1,
support:h),"function"==typeof Symbol&&(w.fn[Symbol.iterator]
=n[Symbol.iterator]),w.each("Boolean Number String Function
Array Date RegExp Object Error Symbol".split(" "),function(e,
t){l"[object "+t+"]"]的文化.toLowerCase()});function C(e){var
t=!!e&&"length"in e&&e.length,n=x(e);return!g(e)&&!y(e)&&
("array"==n||0==t)||"number"==typeof t&&t>0&&t-1 in e});var
E=function(e){var t,n,r,i,o,a,s,u,l,c,f,p,d,h,g,y,v,m,x,
b="sizzle"+1*new Date,w=e.document,T=0,C=0,E=ae(),k=ae(),S=ae
(),D=function(e,t){return e==t&&(f=!0),0},N={};
hasOwnProperty,A=[],j=A.pop,q=A.push,L=A.push,H=A.slice,
O=function(e,t){for(var n=0,r=e.length;n<r;n++)if(e[n]==t)
return n;return-1},P="checked|selected|async|autofocus|
autoplay|controls|defer|disabled|hidden|ismap|loop|multiple|
open|readonly|required|scoped",M="[\u0020\t\r\n\f]",R="(?:
\\\\\\\\.|[\u0020-\u00a0])+",I="\u0020["+M+"*("+R+")(?:"+M+"*(
[^$|!~]?)+"+M+"*?(?:\\\\\\\\.|[^\\\\\\\\])*)'|\"((?:\\\\\\\\.|[^\\\\\\\\])*)\\\"|(\"+R+\"))|\"+M+"*\\\"]",W=":(\"+R+\")(?:\\\\(((?:"
```

HTML e JavaScript

► **Construindo o carrossel.**

Essa pasta que você baixou do jquery somente serve para você copiar estes arquivos que foram solicitados. Então você apaga-la. Não vai precisar de mais nada dela.

HTML e JavaScript

► Construindo o carrossel.

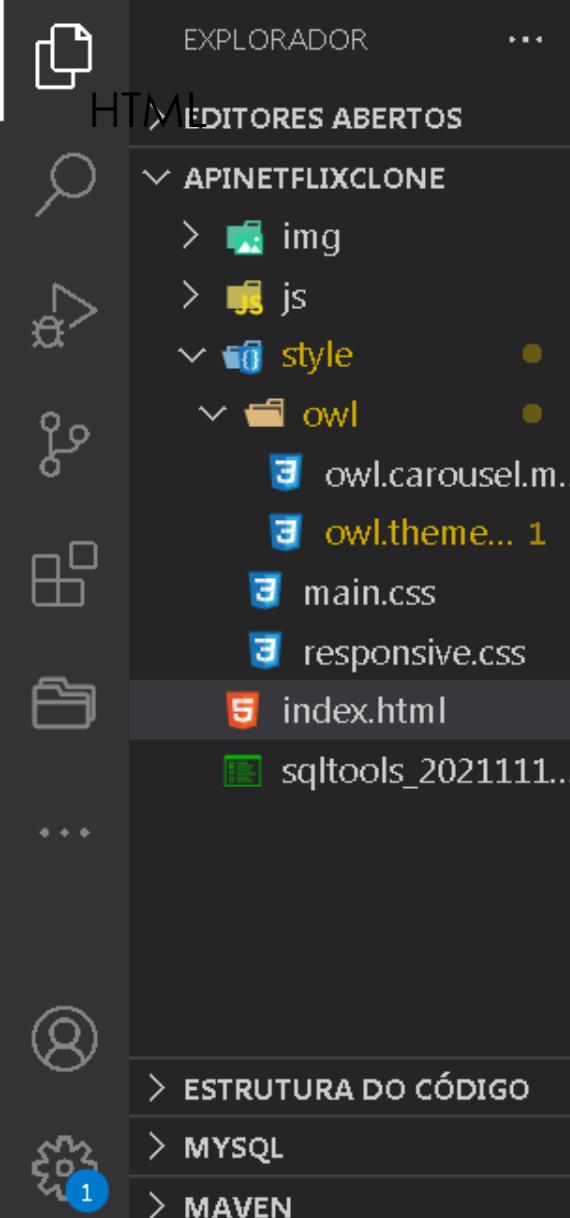
O próximo passo é criar os links para os arquivos que copiamos
E para isso vamos ao topo do programa.

Inserindo essa linha logo após um comentário no index.html

`<!--owl css-->` Assim que se comenta no html

`<link rel="stylesheet" href="style/owl/owl.carousel.min.css">`

`<link rel="stylesheet" href="style/owl/owl.theme.default.min.css">`



```
index.html × main.css owl.theme.default.min.css 1

index.html > html > head
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="style/main.css">
7
8      <!--responsividade-->
9      <link rel="stylesheet" href="style/responsive.css">
10
11     <!--owl.css-->
12     <link rel="stylesheet" href="style/owl/owl.carousel.min.css">
13     <link rel="stylesheet" href="style/owl/owl.theme.default.min.css">
14
15     <title>NETFLIX CLONE</title>
16 </head>
17 <body>
18     <header>
19         <div class="container">
20             <h2 class="logo">NETFLIX</h2>
21             <nav>
```

HTML e JavaScript

► **Construindo o carrossel.**

No final do html vamos colocar o javascript isto porque o sistema executa na ordem sequencial e se você tentar carregar o javascript antes da pagina ter carregado pode acontecer algum erro

HTML e JavaScript

► **Construindo o carrossel.**

Feito isso o **owl** já esta instalado.

Depois disso você pode baixar e instalar as imagens para o caroucel na pasta img como esta no projeto.

Aqui salvamos umas 10 imagens.



EXPLORADOR

...

index.html

mini1.jpg X

main.css



...



HTML

EDITORES ABERTOS



APINET... □+ E+ C+ B+

img

capa-house.jpg

mini1.jpg

mini2.jpg

mini3.jpg

mini4.jpg

mini5.jpg

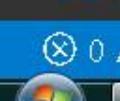
mini6.jpg

mini7.jpg

mini8.jpg

mini9.jpg

mini10.jpg



img > mini1.jpg



HTML e JavaScript

► Construindo o carrossel.

Dentro do **site do owl** tem vários modelos para escolher na **pasta Demos** vamos escolher um para montar o nosso trabalho.

Vamos pegar um Basic.

Ele já nos fornece o código para JavaScript e para o HTML

Demos

Basic demos:

Basic

Responsive

Center

Merge

Auto Width

Url Hash Navigation

Events

stagePadding

Right To Left

NETFLIX CLONE tmdb - Resultados da busca Yah The Movie Database (TMDB) oin caroucel 2 - Resultados da b Basic Demo | Owl Carousel | 2.3.1

owlcarousel2.github.io/OwlCarousel2/demos/basic.html

Apps Importado do IE YII2 URL Gmail Entrada - simuleag... Vidaem Rosas | Elo7 127.0.0.1 FORD Rodobens Parceiros falecom@simuleag... Neon Lista de leitura

HTML

```
        items:3
    },
    1000:{
        items:5
    }
}
})
```

html

```
<div class="owl-carousel owl-theme">
    <div class="item"><h4>1</h4></div>
    <div class="item"><h4>2</h4></div>
    <div class="item"><h4>3</h4></div>
    <div class="item"><h4>4</h4></div>
    <div class="item"><h4>5</h4></div>
    <div class="item"><h4>6</h4></div>
    <div class="item"><h4>7</h4></div>
    <div class="item"><h4>8</h4></div>
    <div class="item"><h4>9</h4></div>
    <div class="item"><h4>10</h4></div>
    <div class="item"><h4>11</h4></div>
    <div class="item"><h4>12</h4></div>
</div>
```



HTML e JavaScript

► **Construindo o carrossel.**

Com base na documentação do owl vamos construir o caroucel.
conforme alteramos o código fazendo as implementações.

```
<div class="carrossel-filmes">  
    <div class="owl-carousel owl-theme">  
        <div class="item">  
              
        </div>  
        <div class="item">  
              
        </div>  
        <div class="item".....segue
```

index.html x mini1.jpg main.css

HTML index.html > html > body > div.carrosel-filmes > div.owl-carousel.owl-theme > div.item

```
46     </div>
47   </main>
48
49   <div class="carrosel-filmes">
50     <div class="owl-carousel owl-theme">
51       <div class="item">
52         
53       </div>
54       <div class="item">
55         
56       </div>
57       <div class="item">
58         
59       </div>
60       <div class="item">
61         
62       </div>
63       <div class="item">
64         
65       </div>
66       <div class="item">
```

0 0 0:00:00 ▶ 00:00 ⏴ Ln 57, Col 34 (386 selecionado) Espacos: 4 UTF-8 LF HTML ⌂ Port WilliamSilva.gódico@gmail.com Github.com/swilliamsilva PT

HTML e JavaScript

► Construindo o carrossel.

então fazemos a codificação exatamente como solicitado pelo owl

```
<div class="owl-carousel owl-theme">
  <div class="item"><h4>1</h4></div>
  <div class="item"><h4>2</h4></div>
  <div class="item"><h4>3</h4></div>
  <div class="item"><h4>4</h4></div>
  <div class="item"><h4>5</h4></div>
  <div class="item"><h4>6</h4></div>
  <div class="item"><h4>7</h4></div>
  <div class="item"><h4>8</h4></div>
  <div class="item"><h4>9</h4></div>
  <div class="item"><h4>10</h4></div>
  <div class="item"><h4>11</h4></div>
  <div class="item"><h4>12</h4></div>
</div>
```

HTML e JavaScript

► **Construindo o carrossel.**

E você precisa referenciar também o owl dentro do js

Para isso criamos o arquivo setup.js dentro da pasta js e nele copiamos o código conforme oferecido pelo owl

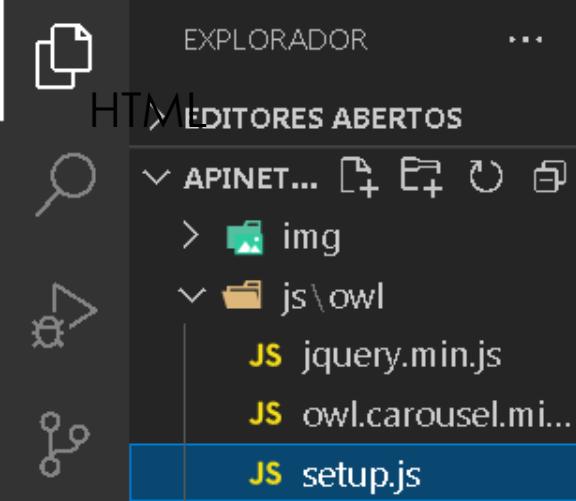
HTML

Setup

```
$('.owl-carousel').owlCarousel({
    loop:true,
    margin:10,
    nav:true,
    responsive:{
        0:{
            items:1
        },
        600:{
            items:3
        },
        1000:{
            items:5
        }
    }
})
```

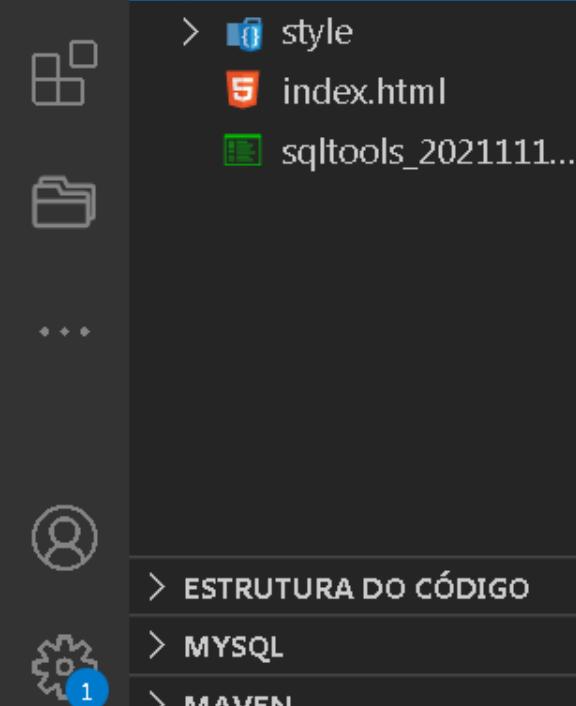
html





```
5 index.html    JS setup.js    X 3 main.css

js > owl > JS setup.js
1   $('.owl-carousel').owlCarousel({
2       loop:true,
3       margin:10,
4       nav:false,
5       responsive:{
6           0:{
7               items:1
8           },
9           600:{
10              items:3
11          },
12          1000:{
13              items:5
14          }
15      }
16  })
```



- > ESTRUTURA DO CÓDIGO
- > MYSQL
- > MAVEN

HTML e JavaScript

► Construindo o carrossel.

Então dentro do **main.css** criamos uma classe onde colocamos as configurações dessas imagens.

A ideia é especificar uma altura e uma largura para controlar essas imagens.



EXPLORADOR

...

index.html

main.css

owl.carousel.min.js



EDITORES ABERTOS



APINETFLIXCLONE



js\owl

jquery.min.js

owl.carousel.mi...

setup.js



style



owl

main.css

responsive.css



index.html

sqltools_2021111...



...



> ESTRUTURA DO CÓDIGO

> MYSQL

> MAVEN



style > main.css > .box-filme

93 margin-right: 8px;

94 }

95

96

97 .container{

98 margin-left: 20px;

99 }

100

101 .filme-principal .container{

102 width: 70%;

103 }

104

105 .box-filme{

106 height: 100%;

107 width: 100%;

108 display: block;

109 cursor: pointer;

110 }

111

112

113 .carrossel-filmes{

114



Ln 110, Col 2 (89 selecionado)

Espaços: 4

UTF-8

LF

CSS



Port Williamsilva.godígo@gmail.com

Github.com/swilliamsilva

PT



HTML e JavaScript

► **Construindo o carrossel.**

Para que tudo funcione corretamente não se pode esquecer de linkar o setup.js ao arquivo html



EXPLORADOR

◀ □ ▶

5 index.html X

3 main.css

JS owl.carousel.min.js



> EDITORES ABERTOS



> ESTRUTURA DO CÓDIGO

> MySQL

> MAVEN

⊗ 0 ⚠ 0 ≡ 0 ► 00:00

1

Ln 89, Col 44 (39 seleccionado)

Espaços: 4

UTF-8

LF HTM

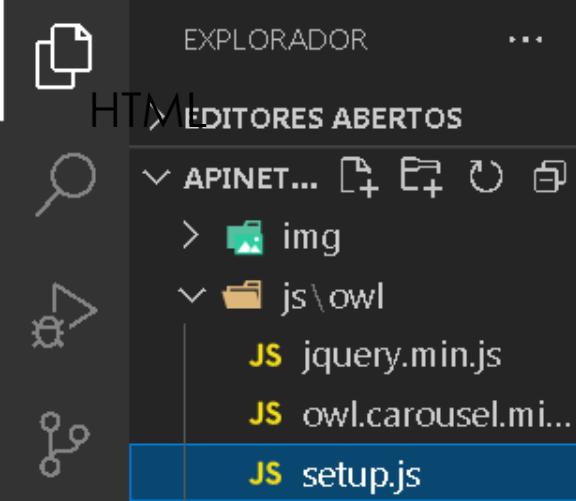
Por

: 5500 Node v13.14.0

HTML e JavaScript

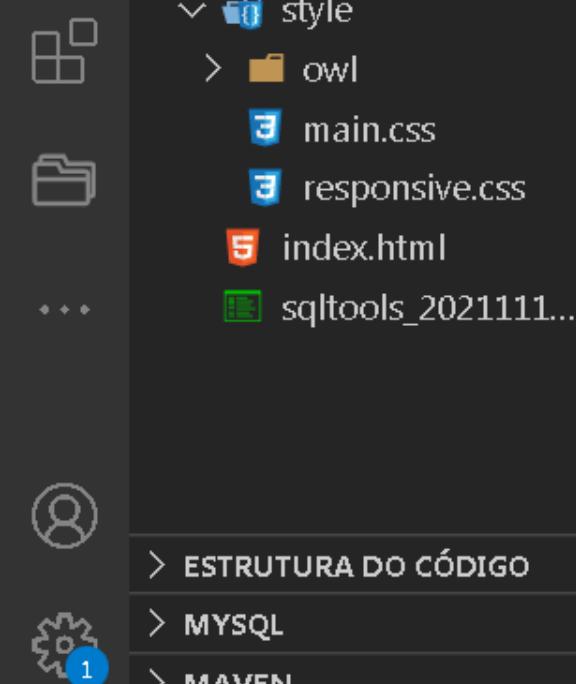
► **Construindo o carrossel.**

Vamos analisar o arquivo setup.js



```
5 index.html    JS setup.js    X 3 main.css

js > owl > JS setup.js > ...
1 1 $('.owl-carousel').owlCarousel({
2   loop:true,
3   margin:10,
4   nav:false,
5   responsive:{ 
6     0:{ 
7       items:1
8     },
9     600:{ 
10       items:3
11     },
12     1000:{ 
13       items:5
14     }
15   }
16 })
```



00:00

Ln 1, Col 1 Espaços: 4 UTF-8 LF {} JavaScript ⚡ Port : 5500 Node v13.14.0 🔍 🌐

HTML e JavaScript

► Construindo o carrossel.

loop:true, aqui você informa que deseja o efeito de loop que o movimento entre os elementos do caroucel.

Se esta true você consegue navegar de forma infinita.

margin:10, ele coloca uma margem entre um quadrado e outro

nav:false, cria aquelas bolinhas de navegação se estiver true.

responsive:{

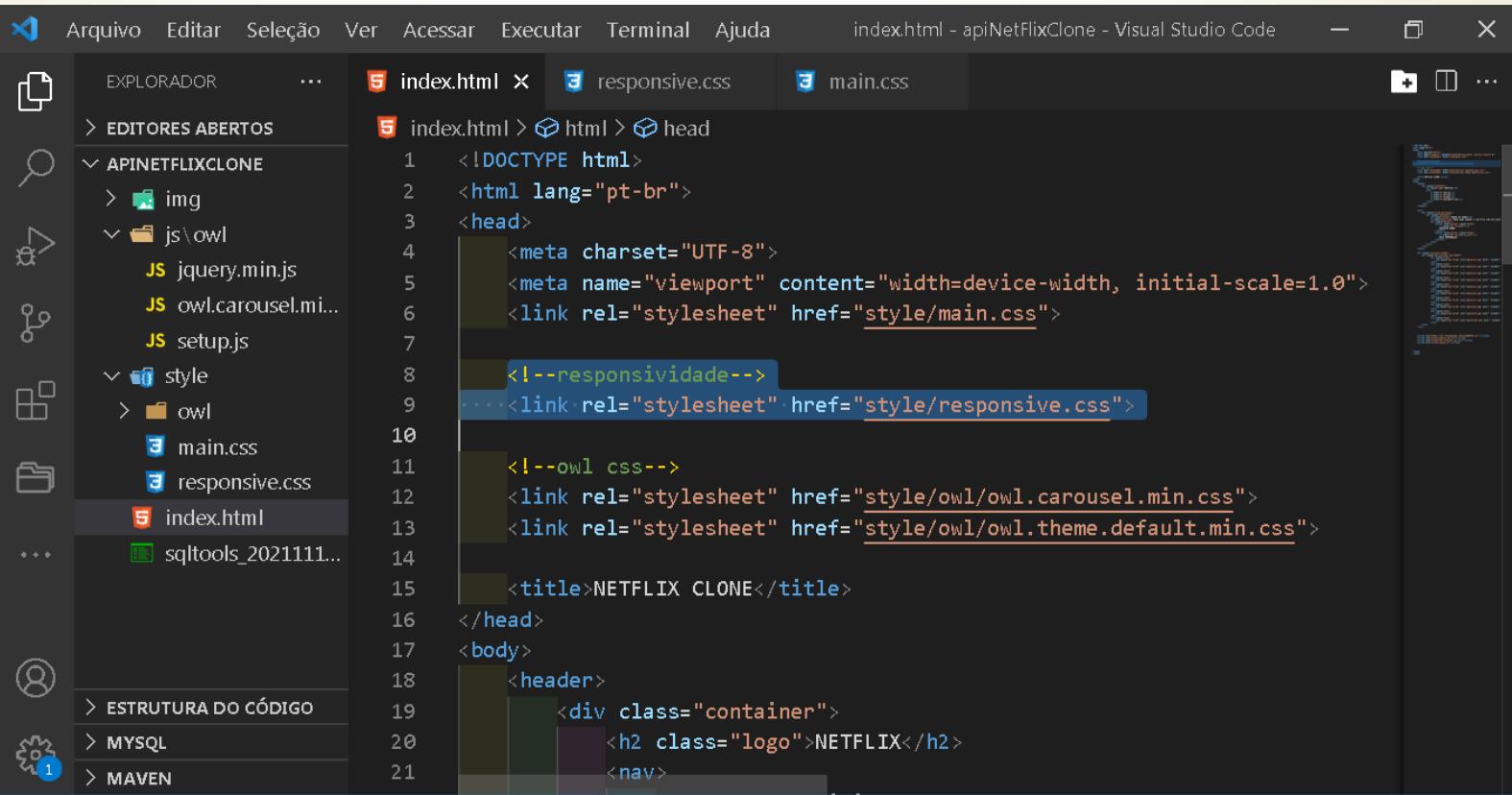
0:{

items:1

}, responsavidade é quando a tela se adapta a diversos equipamentos como desktop, mobile e outros.

JavaScript

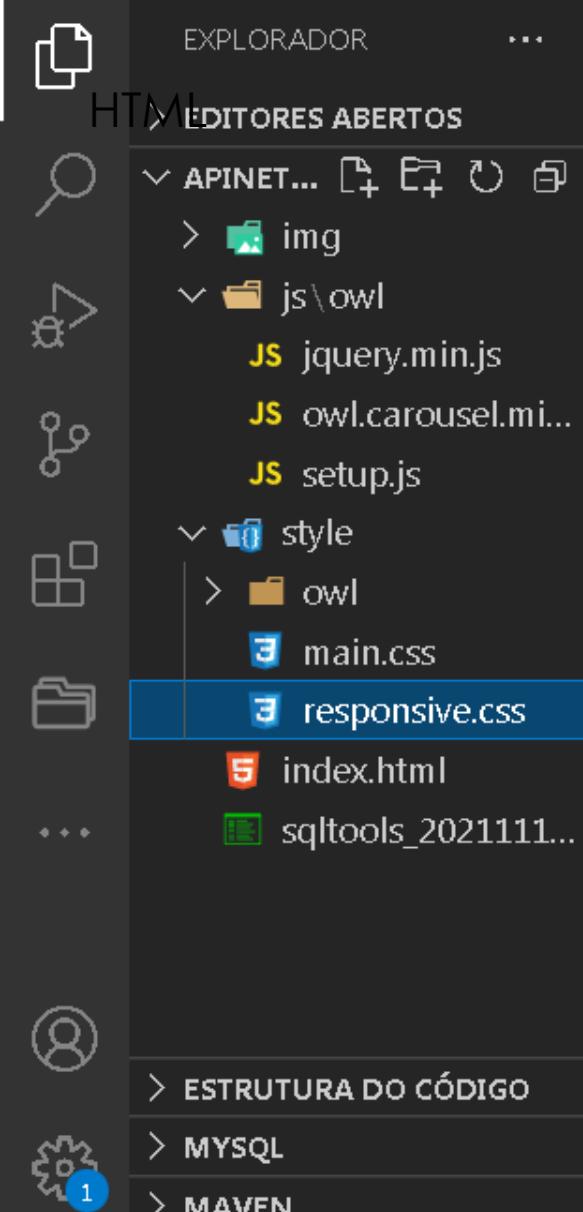
- **Midia query**
- Trata justamente do tamanho da tela do seu site então você pode aplicar estilos dependendo do tamanho da tela.
- Para isso criamos a pasta responsive.css e fazemos o link para o nosso html.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "APINETFLIXCLONE". It includes folders for "img", "js\owl" (containing "jquery.min.js", "owl.carousel.min.js", and "setup.js"), and "style" (containing "main.css" and "responsive.css"). The "index.html" file is also listed.
- Code Editor:** The "index.html" file is open. The code includes meta tags for charset and viewport, and links to "main.css" and "responsive.css". A comment block for responsiveness is present, followed by the link to "responsive.css".

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style/main.css">
    <!--responsividade-->
    ...<link rel="stylesheet" href="style/responsive.css">
    <!--owl css-->
    <link rel="stylesheet" href="style/owl/owl.carousel.min.css">
    <link rel="stylesheet" href="style/owl/owl.theme.default.min.css">
<title>NETFLIX CLONE</title>
</head>
<body>
    <header>
        <div class="container">
            <h2 class="logo">NETFLIX</h2>
            <nav>
```
- Terminal:** Not visible in the screenshot.



index.html responsive.css main.css

```
style > responsive.css > {} @media screen and (max-width:700px)
1  @media screen and (max-width:700px){
2    header .container{
3      display: flex;
4      flex-direction: column;
5    }
6
7    .botao{
8      margin-top: 5px;
9      width: 300px;
10   }
11 }
12
13
14 @media screen and (min-width:1000px){
15   .descricao{
16     width: 50%;
17   }
18 }
```

HTML e JavaScript

► Responsividade

```
@media screen and (max-width:700px){  
    header .container{  
        display: flex;  
        flex-direction: column;  
    }  
}
```

@media, quando você digita essa opção no vscode ele até aparece a opção de auto completar. Então ele exibe essa sintaxe

```
@media screen() {}
```

Já o argumento da função (**max-width:700px**) informa que minha tela terá um tamanho máximo de 700px, quando a tela for até esse valor ele executa os comando entre as {} organizando a tela de forma inteligente.

JavaScript

► Responsividade

Quando a tela fica muito pequena os botões ficam grudados e um em baixo do outro.

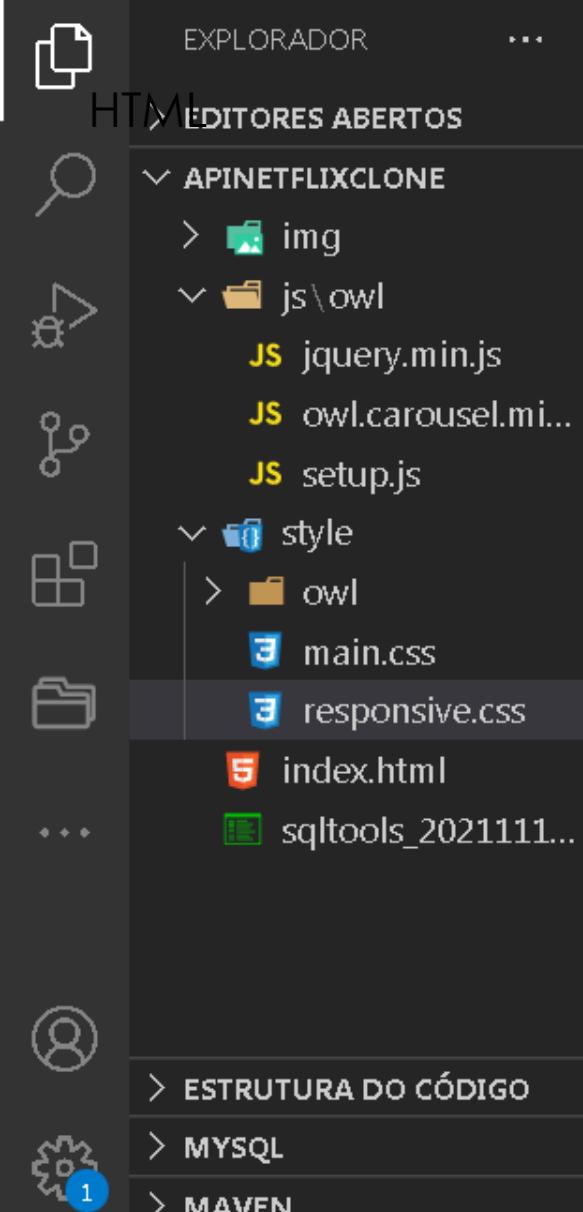
Para corrigir esse efeito na minha **classe botao** eu dou uma

.botao{

margin-top: 5px;

width: 300px; → para padronizar o tamanho dos botões.

}



index.html responsive.css main.css

style > responsive.css > {} @media screen and (max-width:700px) > .botao

```
1  @media screen and (max-width:700px){  
2      header .container{  
3          display: flex;  
4          flex-direction: column;  
5      }  
6  
7      .botao{  
8          margin-top: 5px;  
9          width: 300px;  
10     }  
11 }  
12  
13  
14 @media screen and (min-width:1000px){  
15     .descricao{  
16         width: 50%;  
17     }  
18 }
```

HTML e JavaScript

► Responsividade

Quando a tela abre muito e os elementos ficam espalhados fica feio

Então para isso podemos limitar a expansão com o código abaixo:

@media screen and (min-width:1000px){ → mínimo de largura

.descricao{

width: 50%;

}

Observe o resultado da responsividade no próximo slide.

JavaScriptApiNetflix.pptx - PowerPoint (Falha na Ativação do Produto)

FERRAMENTAS DE DESENHO

ARQUIVO PÁGINA INICIAL INSERIR DESIGN TRANSIÇÕES ANIMAÇÕES APRESENTAÇÃO DE SLIDES REVISÃO EXIBIÇÃO STORYBOARDING FORMATAR

Conta da Microsoft

HTML

121

122

123

124

125

126

127

Seleção

Mostrar Tudo Ocultar Tudo

Espaço Reservado para Conteúdo 2

Título 1

NETFLIX tmdb The N oin ca Basic

← → C Ficheiro | C:/Users/SamuelPMASilva/JavaScript... Apps Importado do IE YII2 URL Gmail Entrada - simuleag... » Lista de leitura

HOUSE OF CARDS

Nada pode impedir o político sem escrúulos Frank Underwood de conquistar Washington. Assista agora a nova temporada de House of Cards que está imperdível.

▶ ASSISTIR AGORA

MAIS INFORMAÇÕES

williamsilva.código@gmail.com

Github.com/swilliamsilva