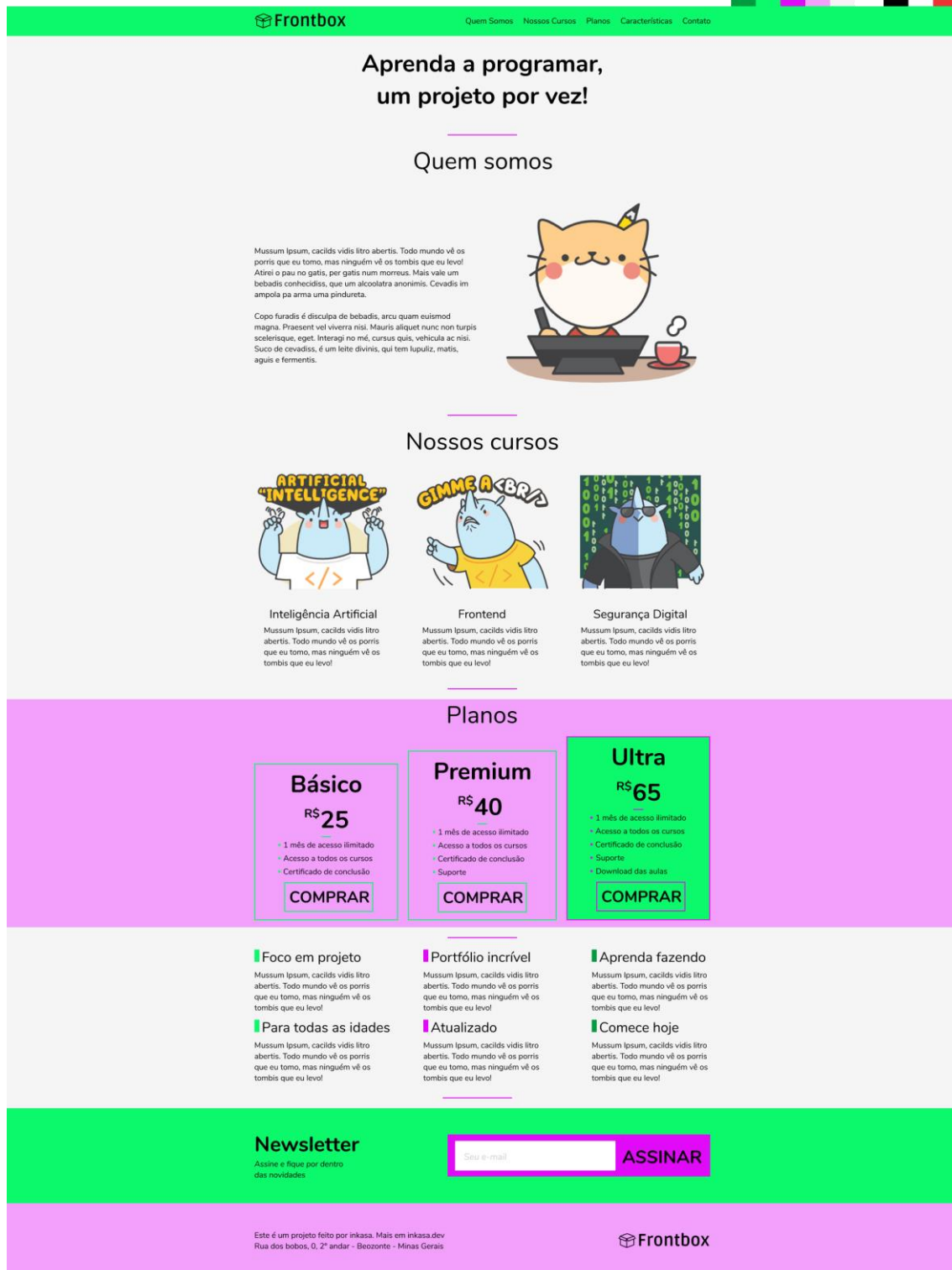


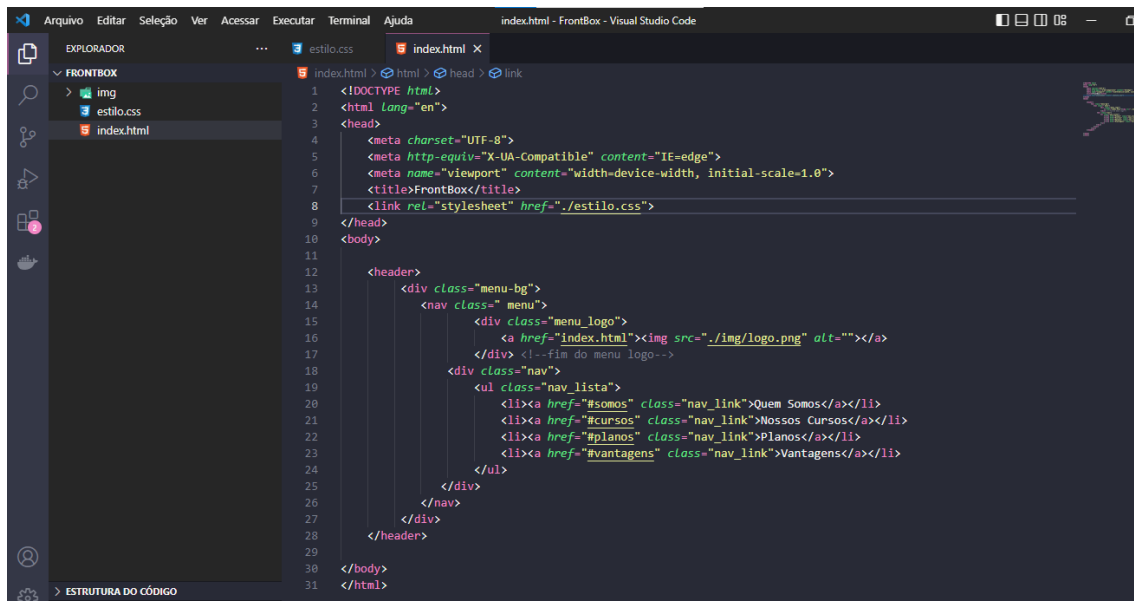
Projeto 1

Neste projeto vamos aprender como criar um site Responsivo.



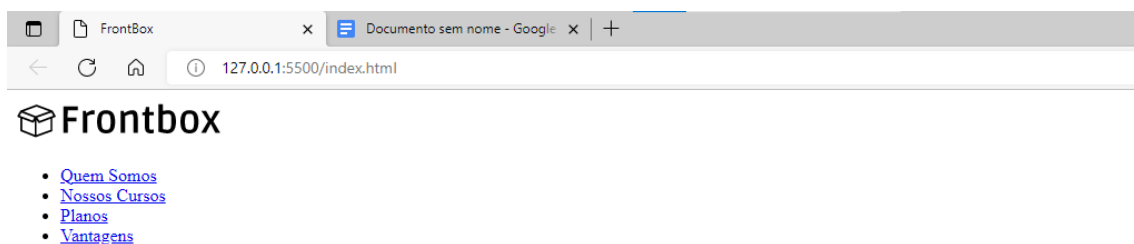
Aplicando tudo que aprendemos até agora: html, css e javascript. Esse é o protótipo do site que vamos desenvolver. Devemos ter a capacidade de imitar a página, usando nossos códigos.

Agora vamos colocar a mão na massa, meus queridos colegas, **vamos criar uma pasta e vamos dar-lhe o nome de FrontBox**. Dentro da pasta, vamos colocar imagem para nós poder usar a desenvolver o site.

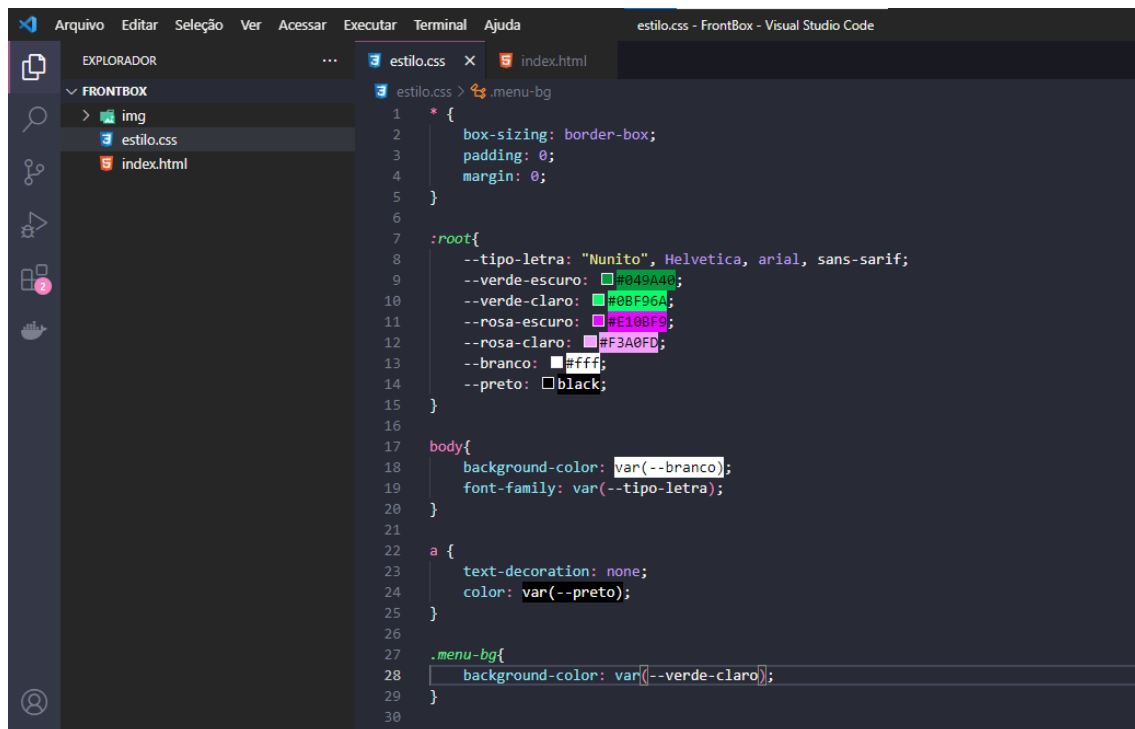


```
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>FrontBox</title>
8   <link rel="stylesheet" href="./estilo.css">
9 </head>
10 <body>
11
12   <header>
13     <div class="menu-bg">
14       <nav class="menu">
15         <div class="menu_logo">
16           <a href="index.html"></a>
17         </div> <!-- fim do menu logo -->
18         <div class="nav">
19           <ul class="nav_lista">
20             <li><a href="#somos" class="nav_link">Quem Somos</a></li>
21             <li><a href="#cursos" class="nav_link">Nossos Cursos</a></li>
22             <li><a href="#planos" class="nav_link">Planos</a></li>
23             <li><a href="#vantagens" class="nav_link">Vantagens</a></li>
24           </ul>
25         </div>
26       </nav>
27     </div>
28   </header>
29
30 </body>
31 </html>
```

Nesse momento já temos a nossa pasta no visual studio code. Já temos a nossa pasta imagem, ficheiro **index.html** e o **estilo.css**. No ficheiro html vamos criar a nossa estrutura básica para fazer o nosso menu do nosso site. Pelo que estamos a ver na imagem, estamos a usar as tags semântica de modo organizar o nosso projecto. No nosso menu html, temos o logo do nosso site e a nossa lista desordenada. cada disse que criamos está a receber uma classe, de modo eu poder mexer no css.É importante colocar classes ou id nas tags, para te facilitar quando tiveres a mexer no css de cada elemento do html.



Utilizando apenas html, o nosso site fica dessa forma. A única coisa que temos até ao momento é o logo da página e a nossa lista. Agora vamos mexer com css, de modo a dar estilo a nossa página.

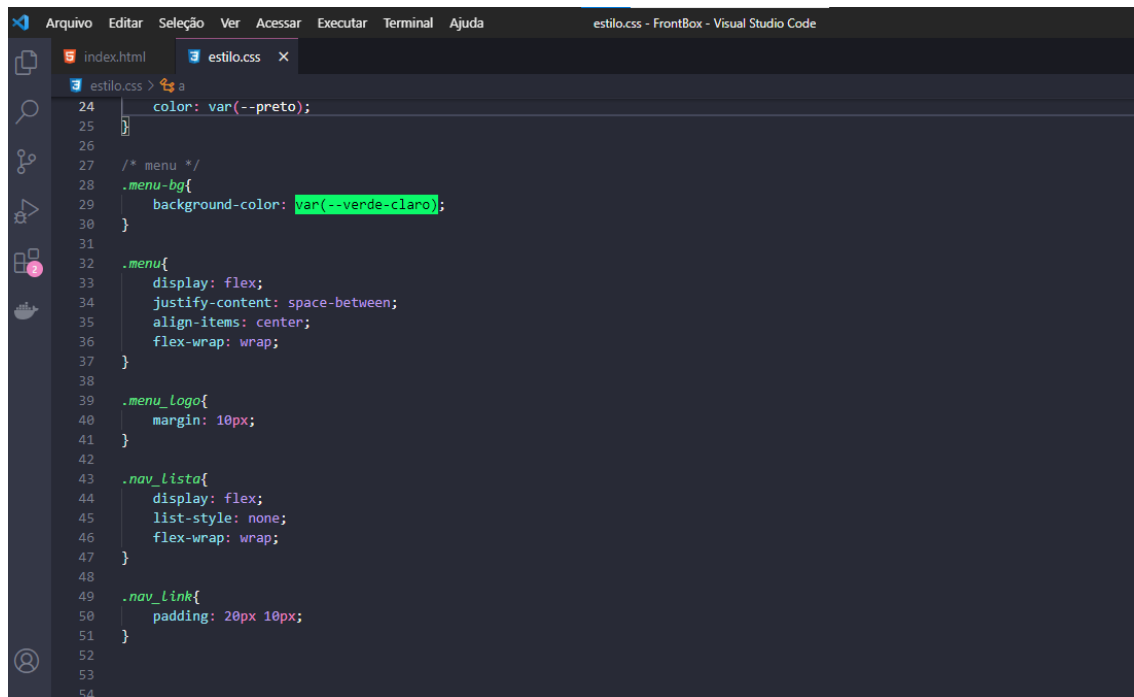


```
1 * {
2   box-sizing: border-box;
3   padding: 0;
4   margin: 0;
5 }
6
7 :root{
8   --tipo-letra: "Nunito", Helvetica, arial, sans-serif;
9   --verde-escuro: #049A46;
10  --verde-claro: #08F96A;
11  --rosa-escuro: #E188F9;
12  --rosa-claro: #F3A0FD;
13  --branco: #fff;
14  --preto: black;
15 }
16
17 body{
18   background-color: var(--branco);
19   font-family: var(--tipo-letra);
20 }
21
22 a {
23   text-decoration: none;
24   color: var(--preto);
25 }
26
27 .menu-bg{
28   background-color: var(--verde-claro);
29 }
30
```

Nesse momento estamos a mexer no css, assim como estão a ver na imagem. criamos um selector global, onde aplicamos um **box-sizing: border-box;** de modo a controlar a largura e altura do meu navegador da minha página. Tivemos que aplicar um **padding: 0; e margin: 0;** de modo eliminar todas as margem interna e externa. O **:root** é o mesmo que dizer **html**, é a raiz da nossa página. Na nossa raiz aplicamos o tipo de letra e as cores que vamos utilizar durante a construção do site. Aplicamos uma cor de fundo branca no corpo da nossa página. Isso quer dizer, que o fundo inteiro da nossa página é branco. Aplicamos uma cor verde no fundo da nossa página.

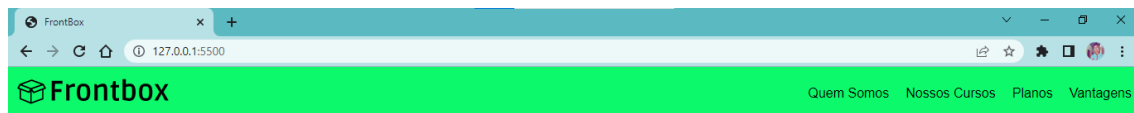


O nosso site está assim nesse momento. Vamos dar continuidade a nossa página.

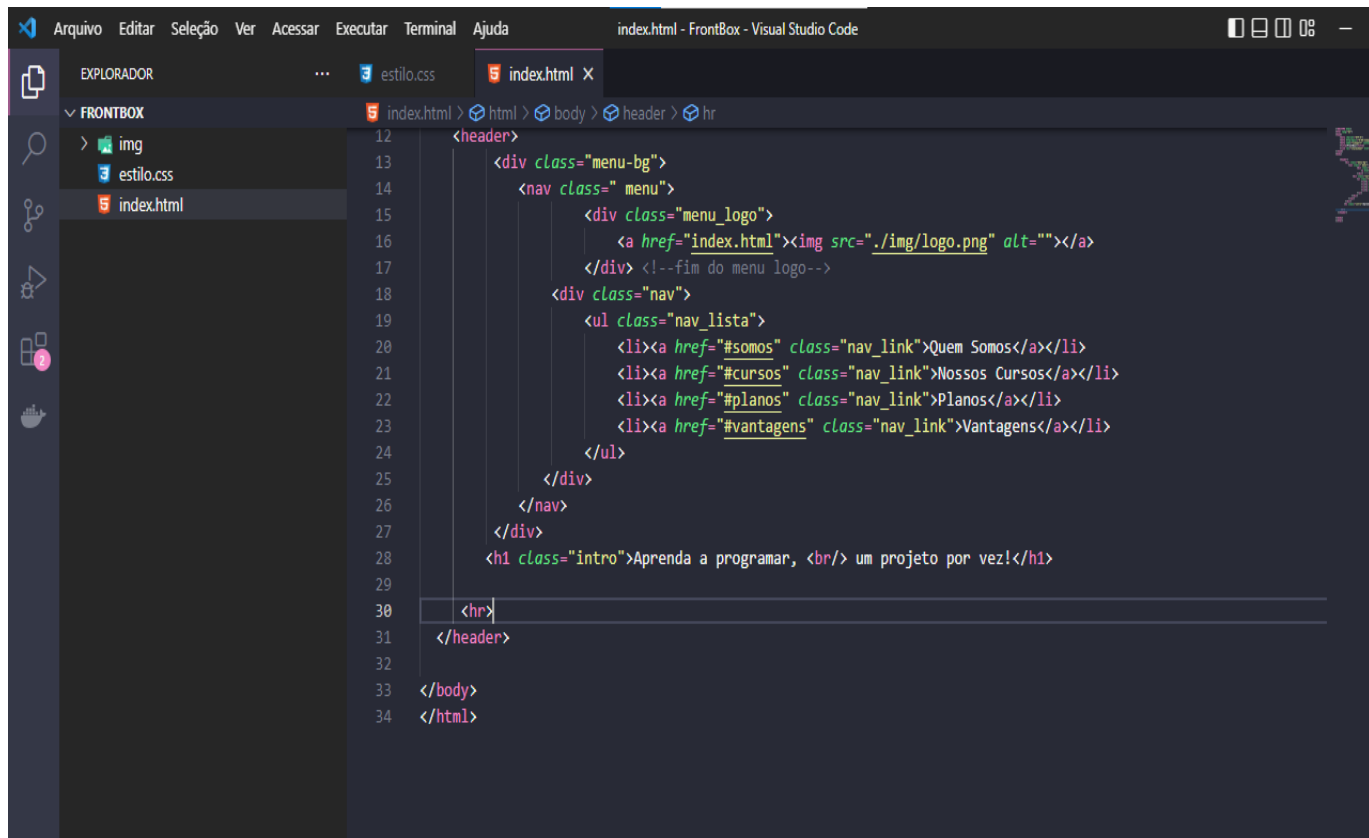


```
estilo.css
24     color: var(--preto);
25 }
26
27 /* menu */
28 .menu-bg{
29     background-color: var(--verde-claro);
30 }
31
32 .menu{
33     display: flex;
34     justify-content: space-between;
35     align-items: center;
36     flex-wrap: wrap;
37 }
38
39 .menu_Logo{
40     margin: 10px;
41 }
42
43 .nav_Lista{
44     display: flex;
45     list-style: none;
46     flex-wrap: wrap;
47 }
48
49 .nav_Link{
50     padding: 20px 10px;
51 }
52
53
54
```

Agora vamos utilizar o flexbox de modo a tornar o nosso site responsivo. No menu do nosso site tivemos que aplicar um `display: flex;` de modo a separar elemento logo do site e os menus da página. Para isso tivemos que aplicar `justify-content: space-between;` aplicamos também margem interna entre os itens da lista de 20px para parte **superior** e **inferior**. 10px para **esquerda** e **direita**. Nem todos os códigos serão, explicado, porque são códigos que já foram explicados na aula. Agora vamos ver como ficou o site na internet.

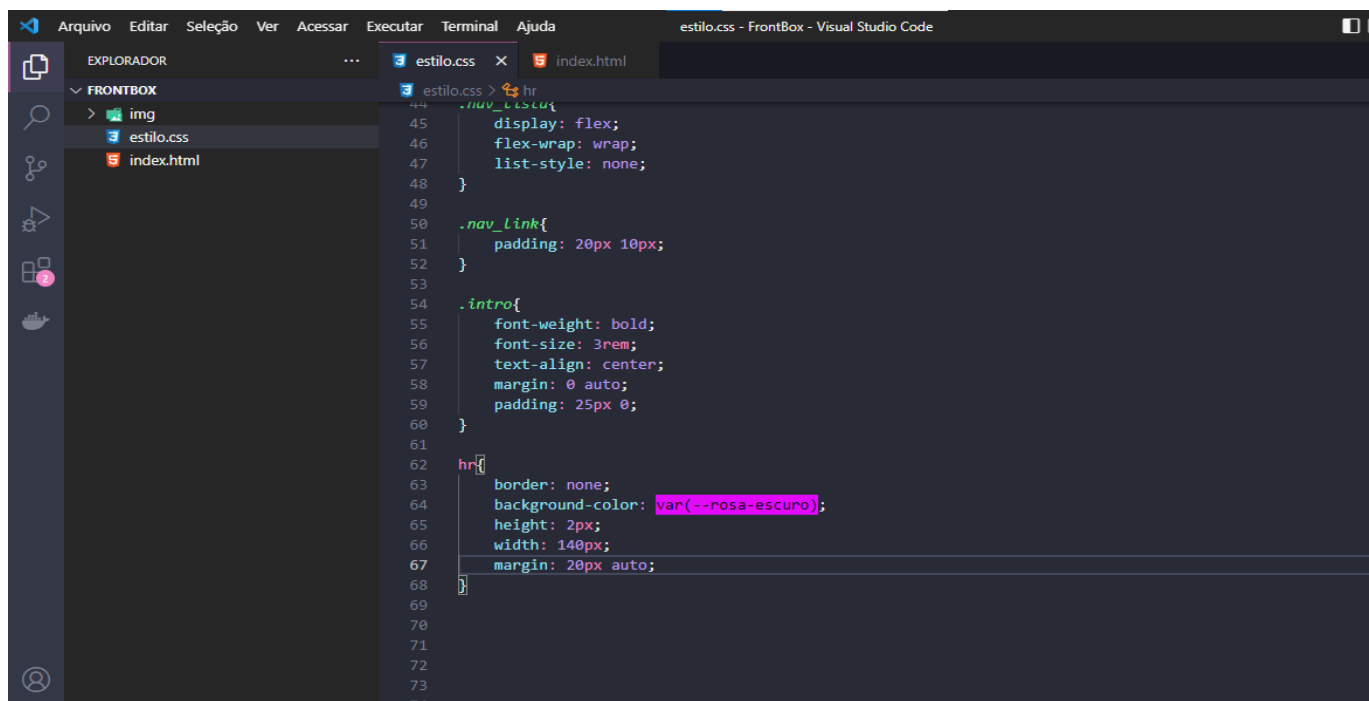


Agora vamos mexer no html, de modo a colocar mais conteúdo.



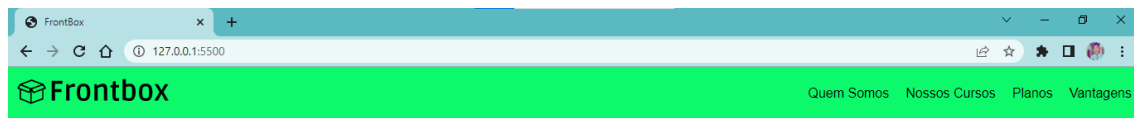
```
12 <header>
13   <div class="menu-bg">
14     <nav class="menu">
15       <div class="menu_logo">
16         <a href="index.html"></a>
17       </div> <!-- fim do menu logo -->
18       <div class="nav">
19         <ul class="nav_lista">
20           <li><a href="#somos" class="nav_link">Quem Somos</a></li>
21           <li><a href="#cursos" class="nav_link">Nossos Cursos</a></li>
22           <li><a href="#planos" class="nav_link">Planos</a></li>
23           <li><a href="#vantagens" class="nav_link">Vantagens</a></li>
24         </ul>
25       </div>
26     </nav>
27   </div>
28   <h1 class="intro">Aprenda a programar, <br/> um projeto por vez!</h1>
29
30   <hr>
31 </header>
32
33 </body>
34 </html>
```

Depois de terminar a div do menu, criamos um titulo da página **“Aprenda a programar, um projeto por vez!”** depois disso, criamos uma linha de baixo do titulo, usando a tag <hr>. Agora vamos levar isso no css.



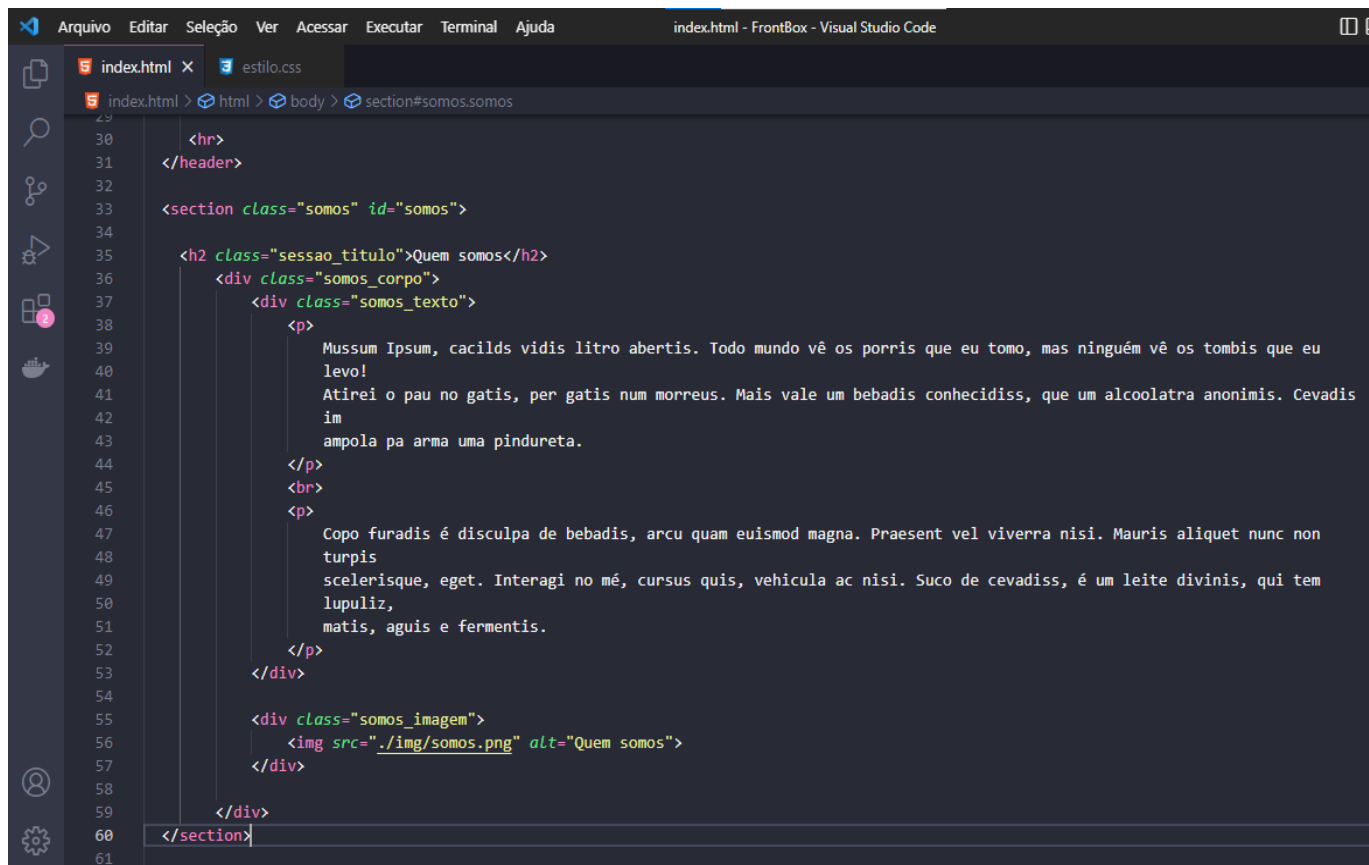
```
44 .nav_lista{
45   display: flex;
46   flex-wrap: wrap;
47   list-style: none;
48 }
49
50 .nav_link{
51   padding: 20px 10px;
52 }
53
54 .intro{
55   font-weight: bold;
56   font-size: 3rem;
57   text-align: center;
58   margin: 0 auto;
59   padding: 25px 0;
60 }
61
62 hr{
63   border: none;
64   background-color: var(--nosa-escuro);
65   height: 2px;
66   width: 140px;
67   margin: 20px auto;
68 }
69
70
71
72
73
74
```

O nosso título recebeu uma classe = “intro”, que vamos utilizar agora no CSS. O nosso título todo queremos colocar como negrito. Para isso tivemos que usar o **font-weight: bold;** aplicamos um tamanho de 3rem no título, e centralizamos o nosso texto no centro. Para colocar a linha, tivemos primeiro que dar uma cor de fundo, depois colocar um tamanho e uma altura. Agora vamos como ficou no navegador.

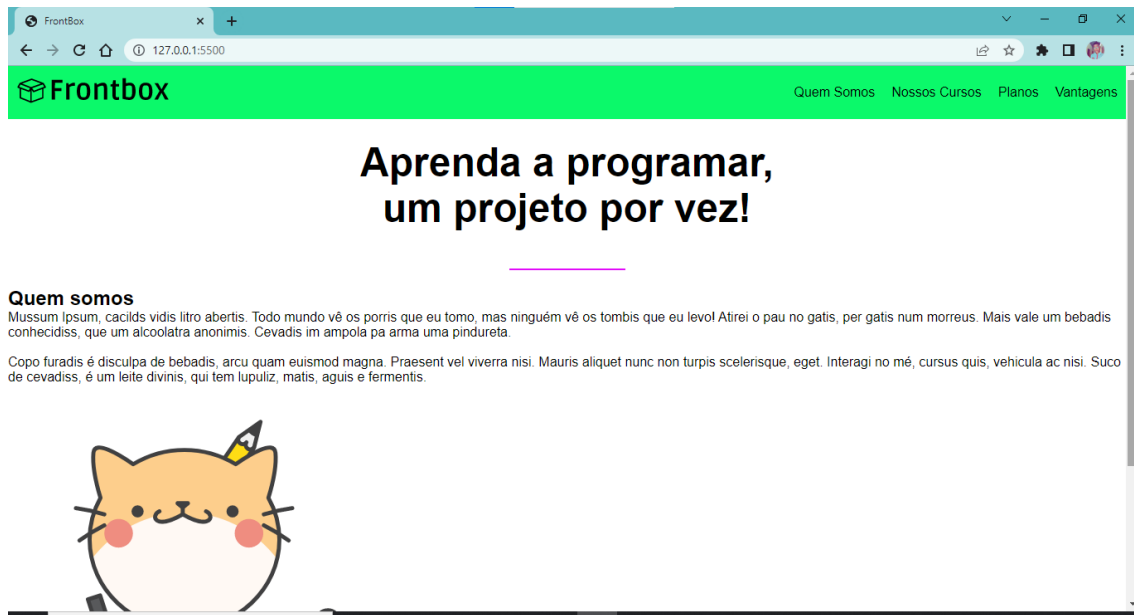


Aprenda a programar, um projeto por vez!

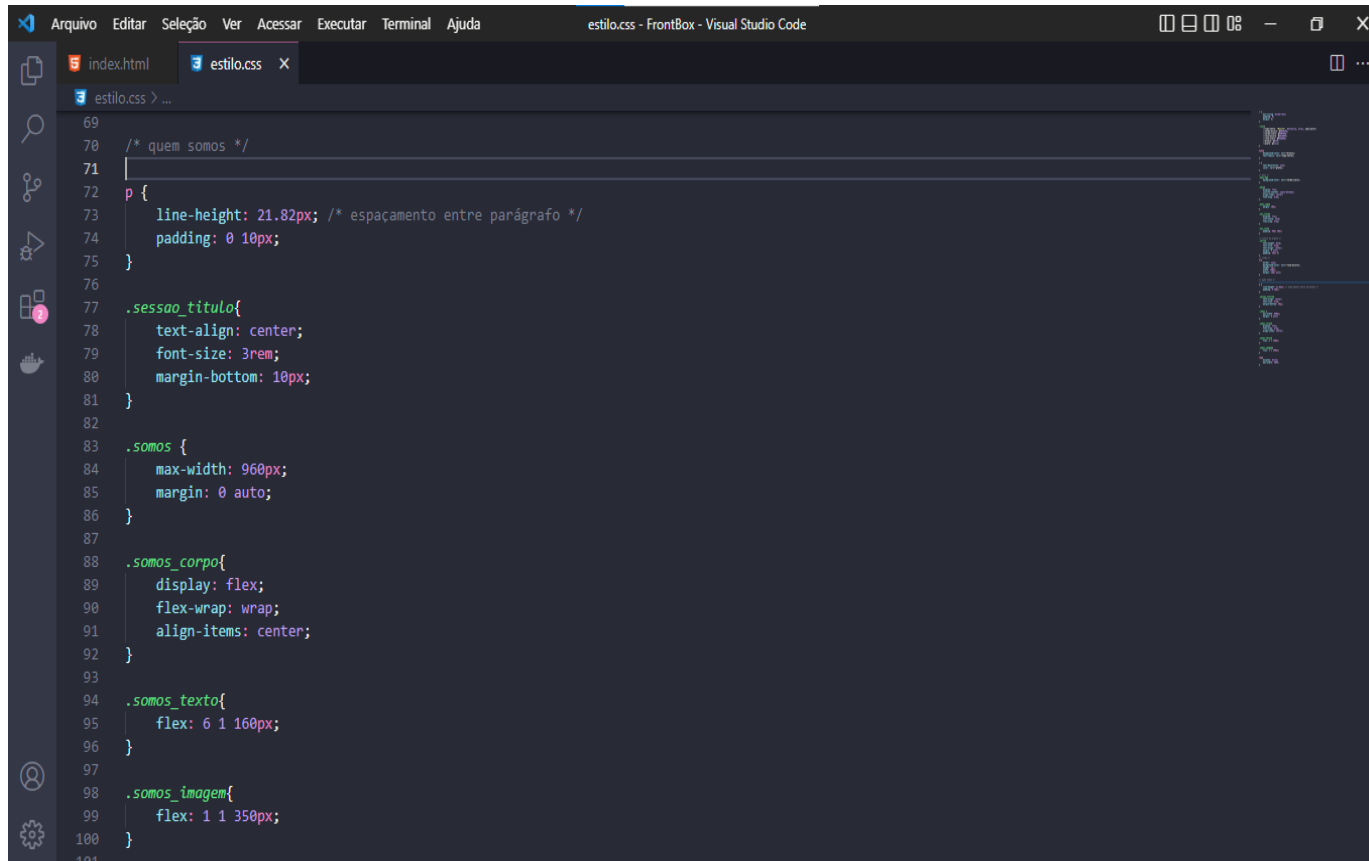
Pelo visto a nossa página está a ficar bonita. Uma forma de memorizar os códigos é praticar sempre. Vamos continuar a codificar a nossa página.



Acabamos de criar a sessão quem somos, criamos parágrafo e colocamos imagem. Agora vamos visualizar como ficou no navegador.

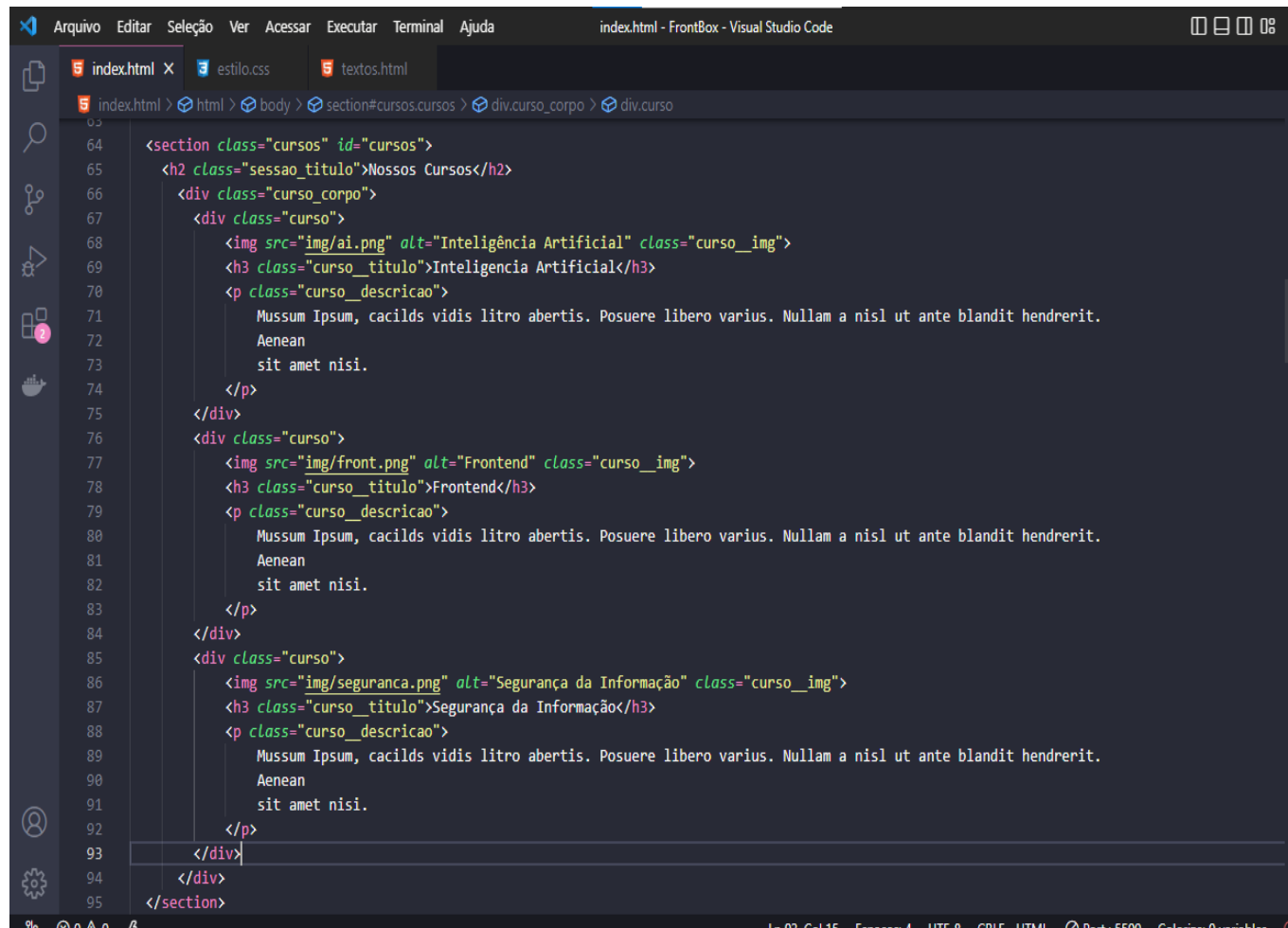


Nesse momento já temos o título, o parágrafo e a imagem. Agora devemos mexer no css, de modo organizar os elementos no site.



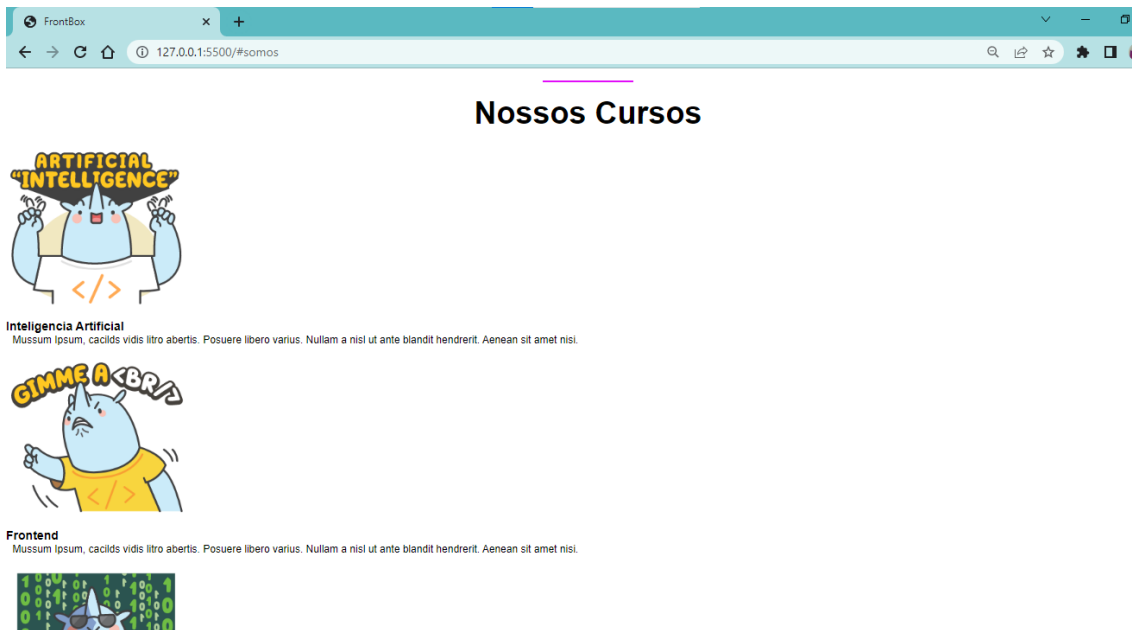
Até ao momento a única coisa de novo é **flex: 6 1 160px**; essa é uma propriedade do flexbox de modo a controlar o tamanho dos elementos filhos. O valor 6 pertence a propriedade **flex-grow:6**; **flex-shrink: 1**; e o **flex-basis: 160px**; O quando colocarmos **flex-grow: 6**, estamos a dizer que o texto deve ser, 6 vezes maior que os outros itens do elemento pai. Nesse caso a imagem. Então o tamanho que margem ocupa do texto é 6 vezes maior que a imagem. **Flex-shrink:1**; está a querer dizer que o texto deve reduzir o tamanho, para caber na margem principal que é do **somos_corpos**. **Flex-basis: 160px**; está a querer dizer que o tamanho inicial do texto é de 160px.

Vamos dar continuidade ao código do site que estamos a fazer.

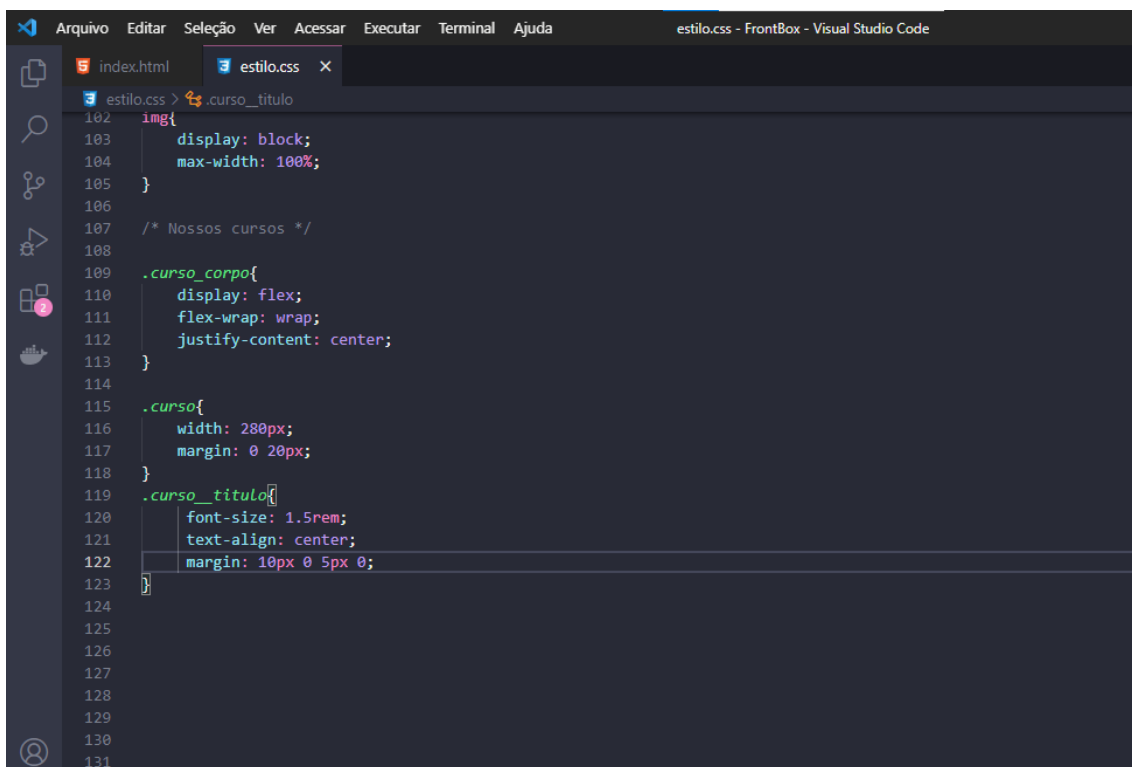
A screenshot of the Visual Studio Code editor interface. The top bar shows the menu (Arquivo, Editar, Seleção, Ver, Acessar, Executar, Terminal, Ajuda) and the active file 'index.html - FrontBox - Visual Studio Code'. The Explorer sidebar on the left shows the file structure with 'index.html', 'estilo.css', and 'textos.html'. The main editor area displays the HTML code for a section titled 'Nossos Cursos'. The code is as follows:

```
64 <section class="cursos" id="cursos">
65   <h2 class="sessao_titulo">Nossos Cursos</h2>
66   <div class="curso_corpo">
67     <div class="curso">
68       
69       <h3 class="curso_titulo">Inteligencia Artificial</h3>
70       <p class="curso_descricao">
71         Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit.
72         Aenean
73         sit amet nisi.
74       </p>
75     </div>
76     <div class="curso">
77       
78       <h3 class="curso_titulo">Frontend</h3>
79       <p class="curso_descricao">
80         Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit.
81         Aenean
82         sit amet nisi.
83       </p>
84     </div>
85     <div class="curso">
86       
87       <h3 class="curso_titulo">Segurança da Informação</h3>
88       <p class="curso_descricao">
89         Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit.
90         Aenean
91         sit amet nisi.
92       </p>
93     </div>
94   </div>
95 </section>
```

Nesse momento criamos uma nova sessão que tem como nome “**nossos cursos**”. Vamos poder visualizar agora no navegador.



Agora vamos mexer no css de modo a deixar o nosso curso centralizado e bem organizado.



No elemento pai, aplicamos um **display: flex**; de modo que os elementos filhos fiquem de forma de linha e não de coluna. Diminuímos também a largura dos cursos. E fizemos a separação entre os elementos. Vamos ver como ficou.



Nossos Cursos



Inteligencia Artificial

Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit. Aenean sit amet nisi.



Frontend

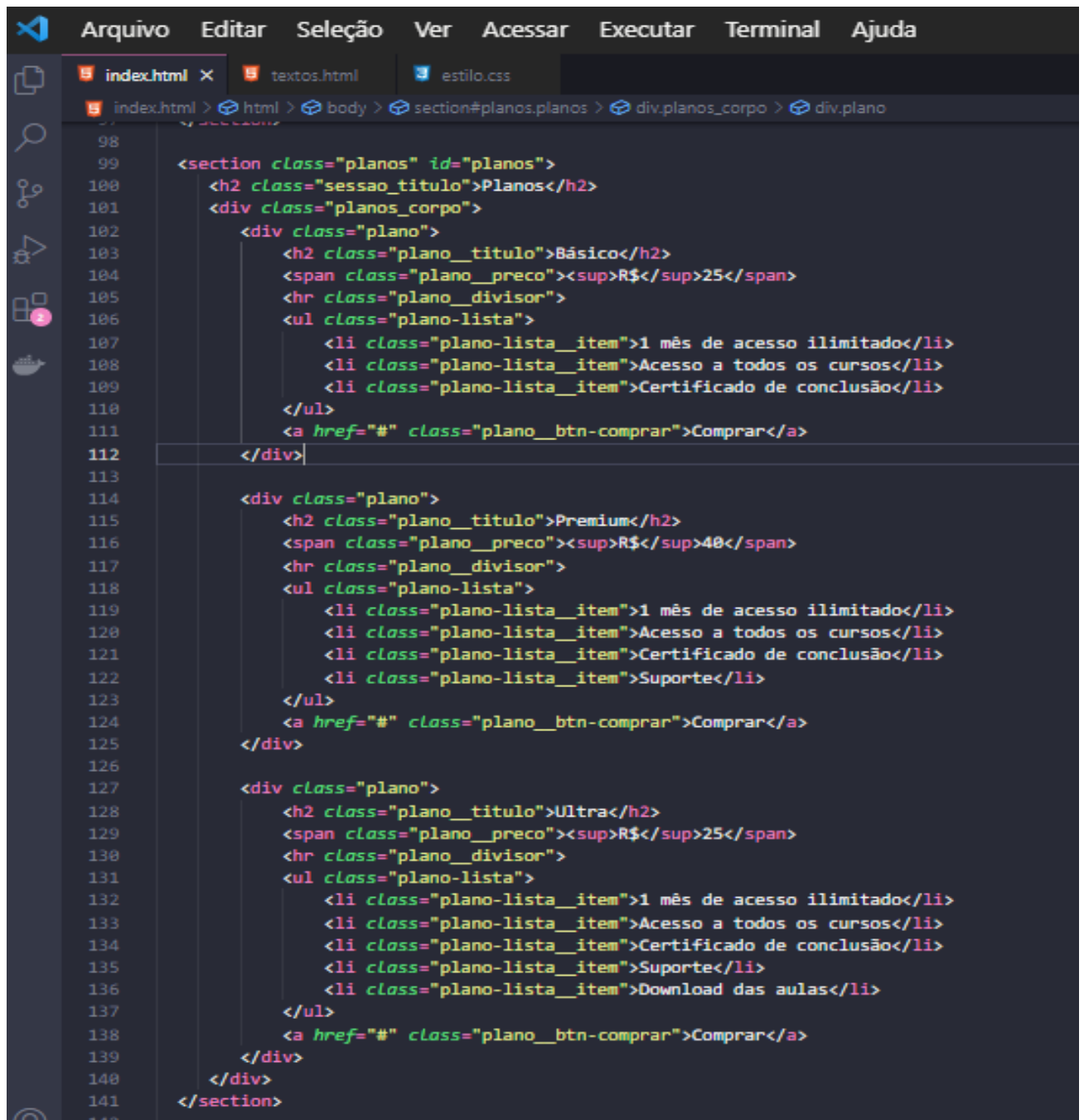
Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit. Aenean sit amet nisi.



Segurança da Informação

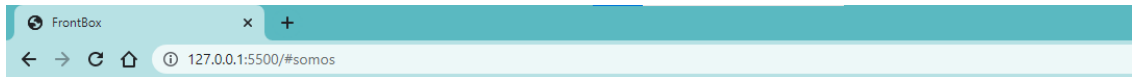
Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl ut ante blandit hendrerit. Aenean sit amet nisi.

Nesse momento estamos a ver os nossos cursos centralizados, utilizando o **justify-content: center;** vamos continuar a codificar.



```
98
99 <section class="planos" id="planos">
100   <h2 class="sessao_titulo">Planos</h2>
101   <div class="planos_corpo">
102     <div class="plano">
103       <h2 class="plano_titulo">Básico</h2>
104       <span class="plano_preco"><sup>R$</sup>25</span>
105       <hr class="plano_divisor">
106       <ul class="plano-lista">
107         <li class="plano-lista_item">1 mês de acesso ilimitado</li>
108         <li class="plano-lista_item">Acesso a todos os cursos</li>
109         <li class="plano-lista_item">Certificado de conclusão</li>
110       </ul>
111       <a href="#" class="plano_btn-comprar">Comprar</a>
112     </div>
113
114     <div class="plano">
115       <h2 class="plano_titulo">Premium</h2>
116       <span class="plano_preco"><sup>R$</sup>40</span>
117       <hr class="plano_divisor">
118       <ul class="plano-lista">
119         <li class="plano-lista_item">1 mês de acesso ilimitado</li>
120         <li class="plano-lista_item">Acesso a todos os cursos</li>
121         <li class="plano-lista_item">Certificado de conclusão</li>
122         <li class="plano-lista_item">Suporte</li>
123       </ul>
124       <a href="#" class="plano_btn-comprar">Comprar</a>
125     </div>
126
127     <div class="plano">
128       <h2 class="plano_titulo">Ultra</h2>
129       <span class="plano_preco"><sup>R$</sup>25</span>
130       <hr class="plano_divisor">
131       <ul class="plano-lista">
132         <li class="plano-lista_item">1 mês de acesso ilimitado</li>
133         <li class="plano-lista_item">Acesso a todos os cursos</li>
134         <li class="plano-lista_item">Certificado de conclusão</li>
135         <li class="plano-lista_item">Suporte</li>
136         <li class="plano-lista_item">Download das aulas</li>
137       </ul>
138       <a href="#" class="plano_btn-comprar">Comprar</a>
139     </div>
140   </div>
141 </section>
142
```

A maioria dos códigos são repetidos. A única coisa que aumentou é a tag <a> que vai servir como botão. Vamos visualizar e ver como está.



Planos

Básico

R\$25

1 mês de acesso ilimitado
Acesso a todos os cursos
Certificado de conclusão
Comprar

Premium

R\$40

1 mês de acesso ilimitado
Acesso a todos os cursos
Certificado de conclusão
Suporte
Comprar

Ultra

R\$25

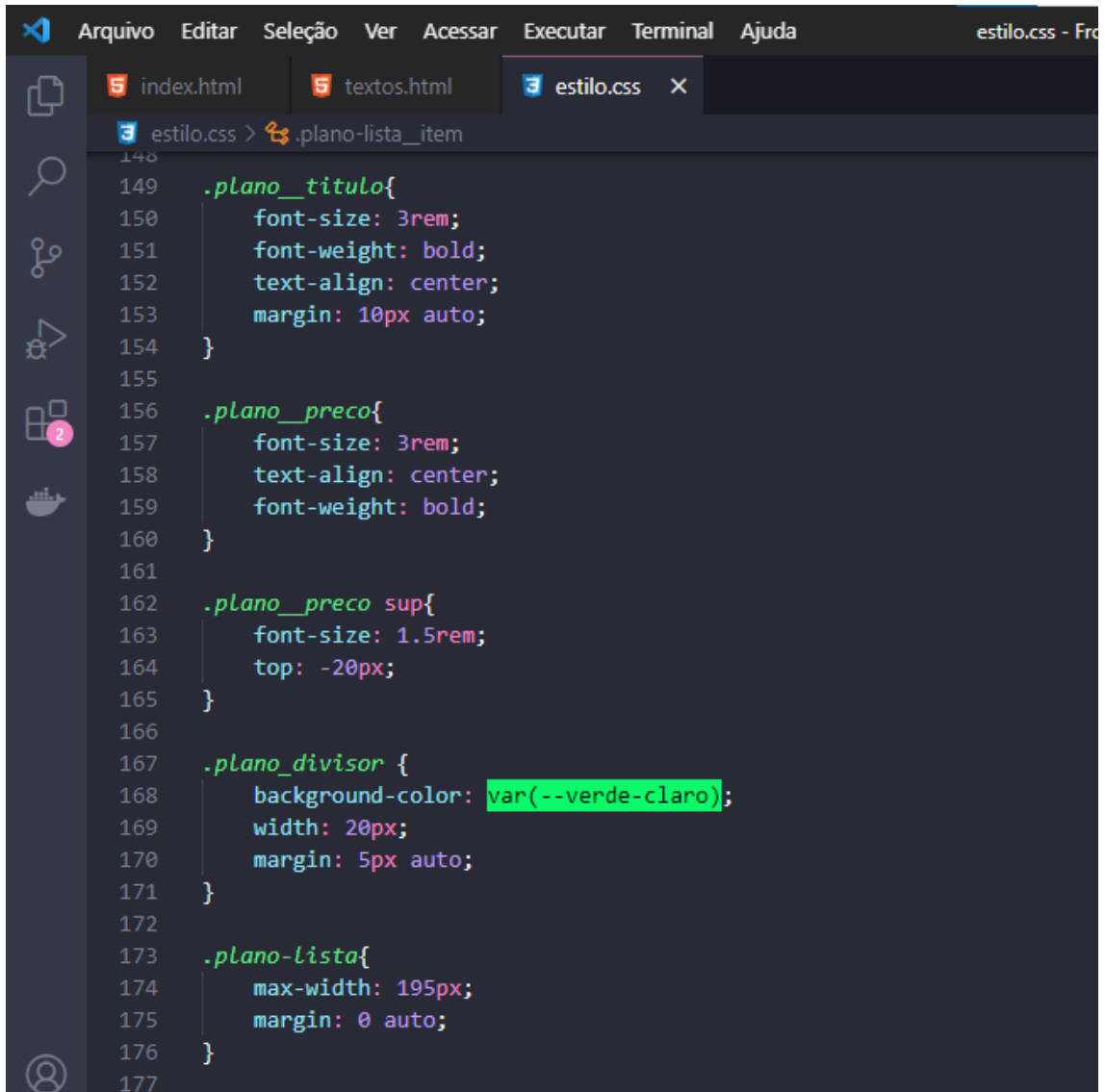
1 mês de acesso ilimitado
Acesso a todos os cursos
Certificado de conclusão
Suporte
Download das aulas
Comprar

Esse é o resultado da nossa página sem o css. Agora vamos mexer no css e ver como vai ficar.

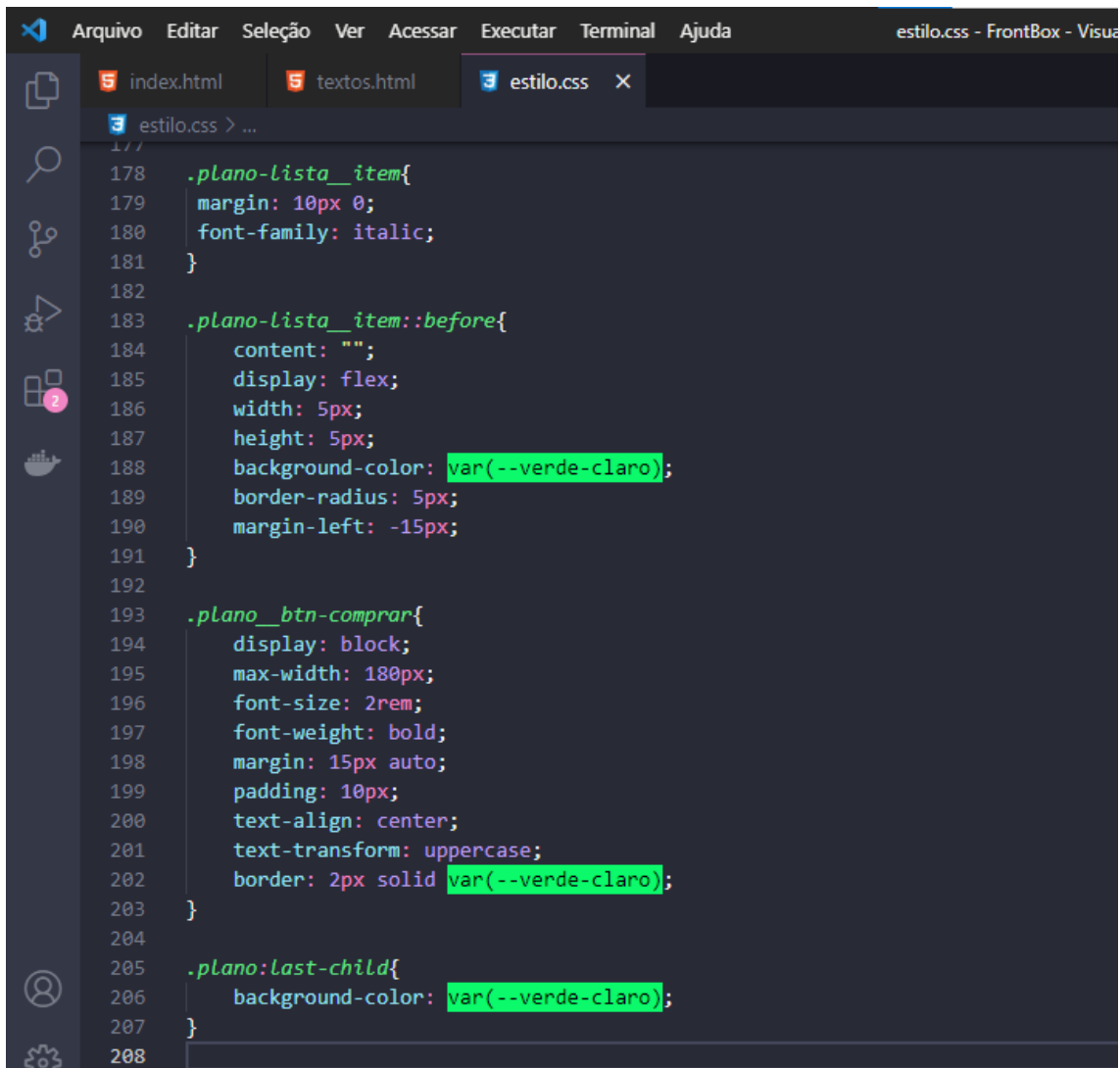
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  estilo.css - FrontBox - Visual Studio Code

index.html  textos.html  estilo.css X

estilo.css > ...
124
125  /* Planos */
126
127  .planos_corpo{
128      max-width: 960px;
129      margin: 0 auto;
130      display: flex;
131      justify-content: center;
132      flex-wrap: wrap;
133  }
134
135  ul{
136      list-style: none;
137  }
138
139  .planos{
140      background-color: var(--rosa-claro);
141  }
142
143  .plano{
144      width: 290px;
145      margin: 10px;
146      border: 2px solid var(--verde-claro);
147  }
148
149  .plano_titulo{
150      font-size: 3rem;
151      font-weight: bold;
152      text-align: center;
153      margin: 10px auto;
154  }
155
```



```
148
149 .plano_titulo{
150     font-size: 3rem;
151     font-weight: bold;
152     text-align: center;
153     margin: 10px auto;
154 }
155
156 .plano_preco{
157     font-size: 3rem;
158     text-align: center;
159     font-weight: bold;
160 }
161
162 .plano_preco sup{
163     font-size: 1.5rem;
164     top: -20px;
165 }
166
167 .plano_divisor {
168     background-color: var(--verde-claro);
169     width: 20px;
170     margin: 5px auto;
171 }
172
173 .plano-lista{
174     max-width: 195px;
175     margin: 0 auto;
176 }
177
```

A screenshot of a code editor window with a dark theme. The top menu bar includes 'Arquivo', 'Editar', 'Seleção', 'Ver', 'Acessar', 'Executar', 'Terminal', and 'Ajuda'. The title bar shows 'estilo.css - FrontBox - Visual Studio Code'. The editor has three tabs: 'index.html', 'textos.html', and 'estilo.css'. The 'estilo.css' tab is active, showing CSS code. The code includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The CSS code defines styles for a shopping cart, including a list item, a button, and a last child element, all using a color variable named '--verde-claro'.

```
177
178 .plano-lista__item{
179     margin: 10px 0;
180     font-family: italic;
181 }
182
183 .plano-lista__item::before{
184     content: "";
185     display: flex;
186     width: 5px;
187     height: 5px;
188     background-color: var(--verde-claro);
189     border-radius: 5px;
190     margin-left: -15px;
191 }
192
193 .plano__btn-comprar{
194     display: block;
195     max-width: 180px;
196     font-size: 2rem;
197     font-weight: bold;
198     margin: 15px auto;
199     padding: 10px;
200     text-align: center;
201     text-transform: uppercase;
202     border: 2px solid var(--verde-claro);
203 }
204
205 .plano:last-child{
206     background-color: var(--verde-claro);
207 }
208
```

Pelo visto o nosso código CSS ficou bastante grande. Mas isso não é um problema. Se você o que cada propriedade do CSS faz, vai facilitar a tua compreensão. Porque a maioria das propriedades estamos a usar várias vezes. Caso tenhas dúvida, volta na aula de CSS, onde explica cada propriedade CSS. Única propriedade é o **::before** Insira algo antes do conteúdo de cada elemento. **:last-child** Seleciona todos os elementos que é o último filho do seu elemento pai.



Esse é resultado do código que acabamos de o fazer. Vamos continuar com os códigos no vs code.


```

<section class="vantagens" id="vantagens">
  <div class="vantagem">
    <h2 class="vantagem__titulo">Foco em projeto</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>

  <div class="vantagem">
    <h2 class="vantagem__titulo">Portfólio incrível</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut
      ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>

  <div class="vantagem">
    <h2 class="vantagem__titulo">Aprenda fazendo</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut
      ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>

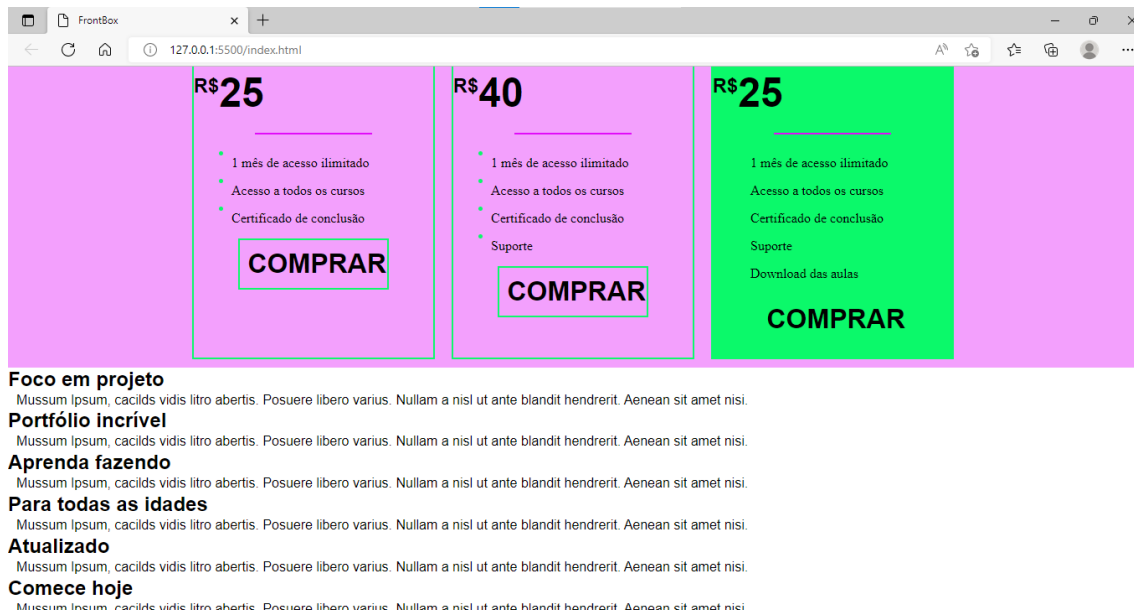
  <div class="vantagem">
    <h2 class="vantagem__titulo">Para todas as idades</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut
      ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>

  <div class="vantagem">
    <h2 class="vantagem__titulo">Atualizado</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut
      ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>

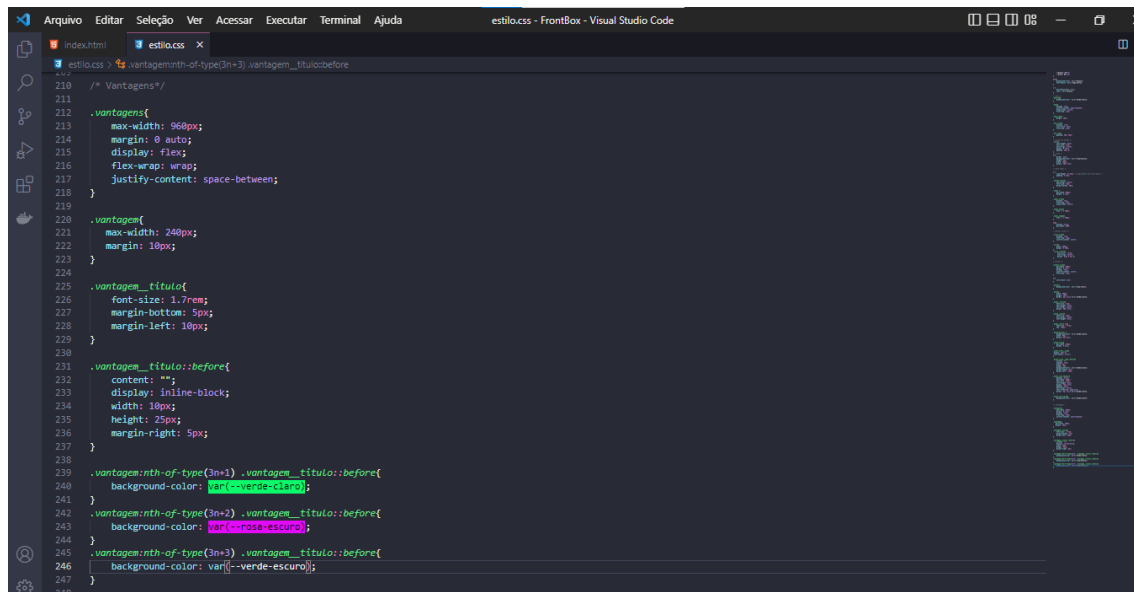
  <div class="vantagem">
    <h2 class="vantagem__titulo">Comece hoje</h2>
    <p class="vantagem__descricao">Mussum Ipsum, cacilds vidis litro abertis. Posuere libero varius. Nullam a nisl
      ut
      ante blandit hendrerit. Aenean sit
      amet nisi.</p>
  </div>
</section>

```

Nesse momento temos a vantagens do curso do site. Agora vamos ver como está sem o css.



Esse é o resultado das vantagens do site. Mas não temos o css. Agora vamos mexer no css e ver como está o nosso site.



A única coisa que temos de diferente **:nth-of-type(3n + 1)**, essa propriedade escolhe a posição do elemento de acordo com o número que está entre parenteses. Inicialmente n = 0; n = 0, 3(0) + 1 = 1; então ele vai selecionar o primeiro elemento. Depois quando n = 1, vai selecionar o ultimo elemento.



E estamos a haver o resultado, como ficou no navegador. Já estamos quase a finalizar o site.

```

192 <section class="newsletter">
193   <div class="newsletter-container">
194     <div class="newsletter-info">
195       <h2 class="newsletter-info_titulo">newsletter </h2>
196       <p class="newsletter-info_descricao"> Assine e fique por dentro das novidades </p>
197     </div>
198     <form action="" class="newsletter-form">
199       <input type="text" placeholder="seu e-mail" class="newsletter-form_input">
200       <button type="submit" class="newsletter-form_submit">Assinar</button>
201     </form>
202   </div>
203 </section>
204
205 <footer class="footer">
206   <p class="footer__copyright">Este é um projecto que servirá de maquete, para o vosso projecto final
207 </p>
208
209   <a href="index.html">
210     
211   </a>
212 </footer>
213
214 </body>
215 </html>

```

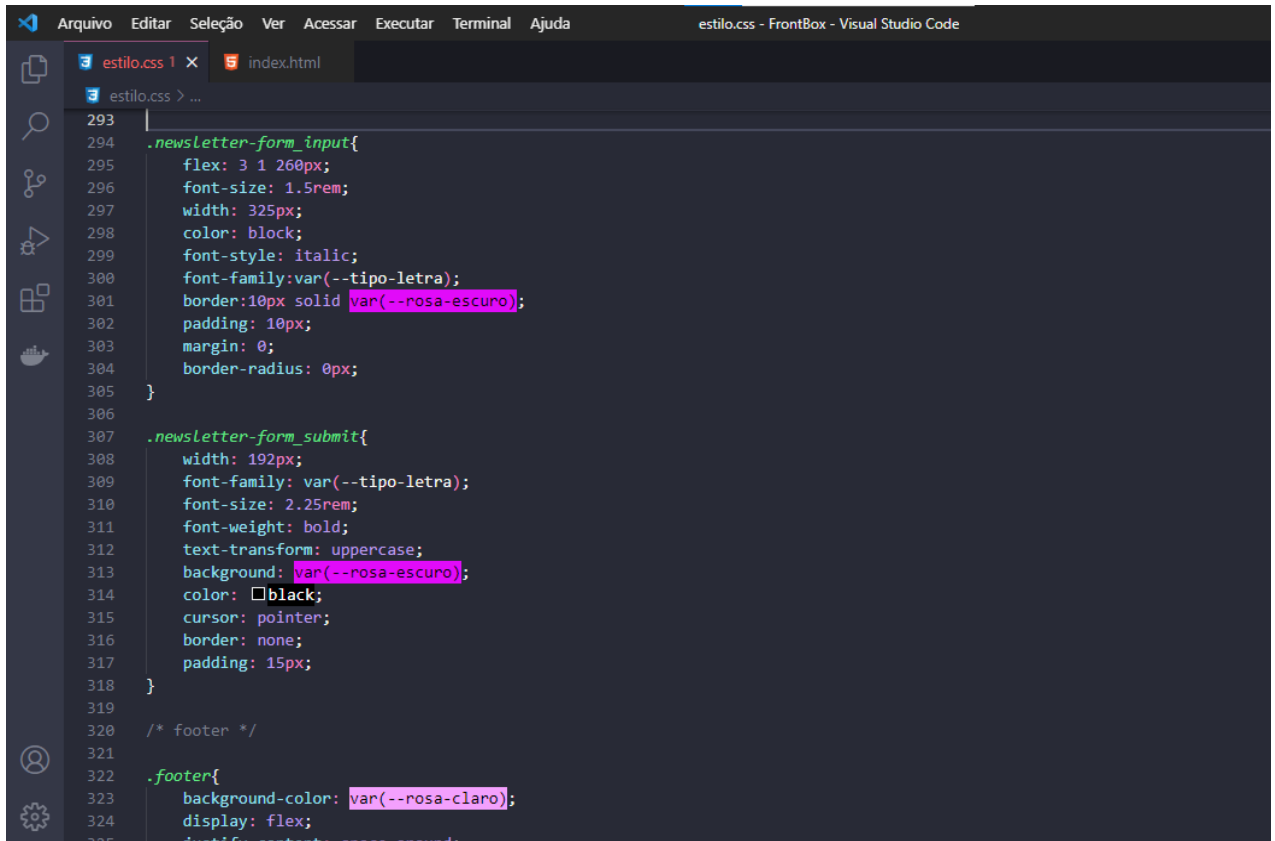
Nesse momento criamos o nosso newsletter e o footer. Agora vamos ver o código css.

```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  estilo.css - FrontBox - Visual Studio Code

estilo.css 1 x  index.html
estilo.css > .newsletter-info_titulo
257
258 /*newsletter*/
259
260 .newsletter{
261     background-color: var(--verde-claro);
262 }
263
264 .newsletter-container{
265     max-width: 960px;
266     margin: 0 auto;
267     padding: 50px 0;
268     display: flex;
269     flex-wrap: wrap;
270 }
271
272 .newsletter-info{
273     padding-left: 10px;
274     flex: 1 1 260px;
275     flex-wrap: wrap;
276     display: flex ;
277 }
278
279 .newsletter-form{
280     flex: 2 1 260px;
281     display: flex;
282     flex-wrap: wrap;
283 }
284
285 .newsletter-info_titulo{
286     font-size: 2.62rem;
287     font-weight: bold;
288 }
289
```

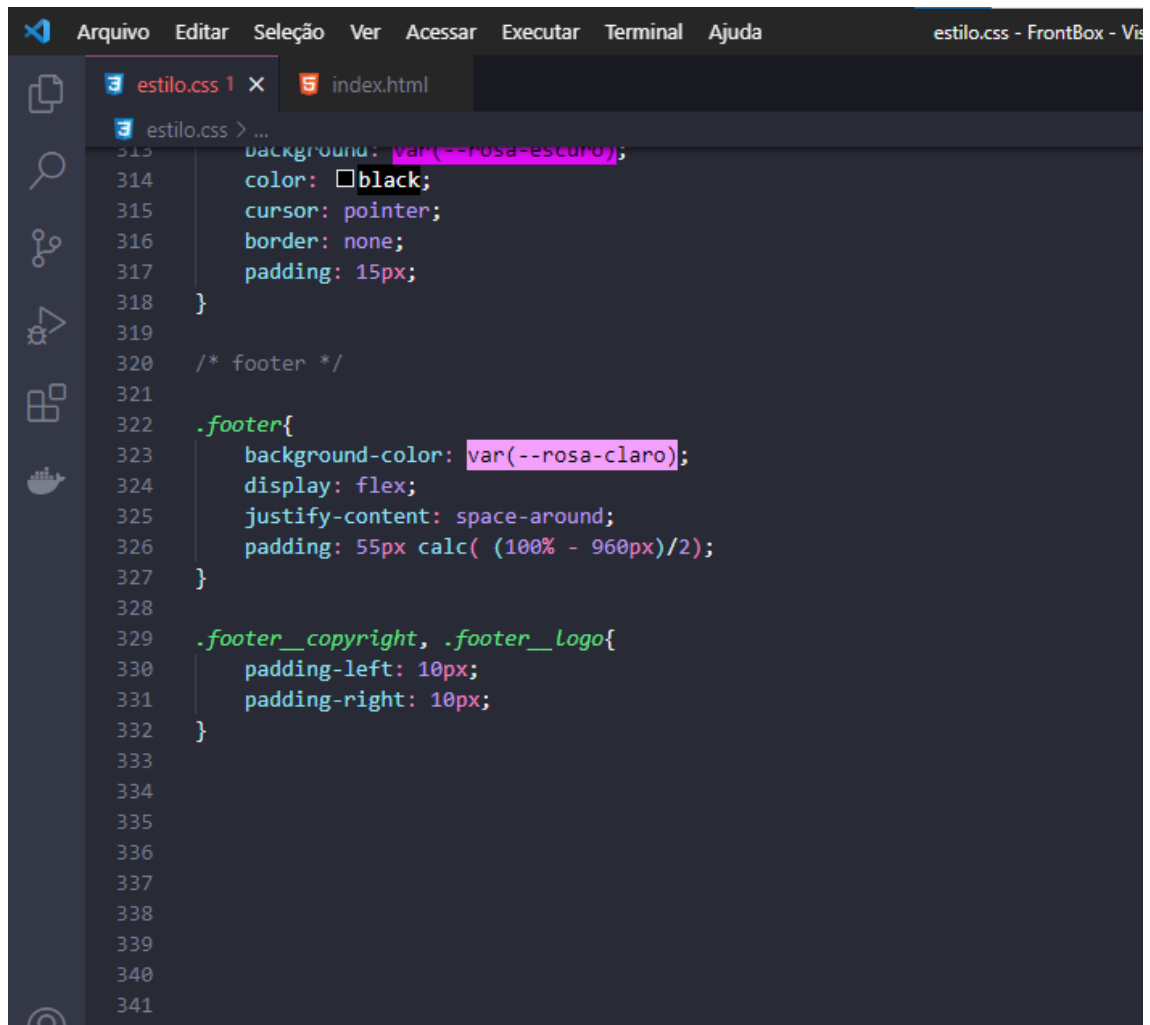
```
Arquivo  Editar  Seleção  Ver  Acessar  Executar  Terminal  Ajuda  estilo.css - FrontBox - Visual Studio Code

estilo.css 1 x  index.html
estilo.css > .newsletter-info_titulo
285 .newsletter-info_titulo{
286     font-size: 2.62rem;
287     font-weight: bold;
288 }
289
290 .newsletter-info_descricao{
291     font-style: italic;
292 }
293
294 .newsletter-form_input{
295     flex: 3 1 260px;
296     font-size: 1.5rem;
297     width: 325px;
298     color: black;
299     font-style: italic;
300     font-family: var(--tipo-letra);
301     border: 10px solid var(--rosa-escuro);
302     padding: 10px;
303     margin: 0;
304     border-radius: 0px;
305 }
306
307 .newsletter-form_submit{
308     width: 192px;
309     font-family: var(--tipo-letra);
310     font-size: 2.25rem;
311     font-weight: bold;
312     text-transform: uppercase;
313     background: var(--rosa-escuro);
314     color: black;
315     cursor: pointer;
316     border: none;
317 }
```



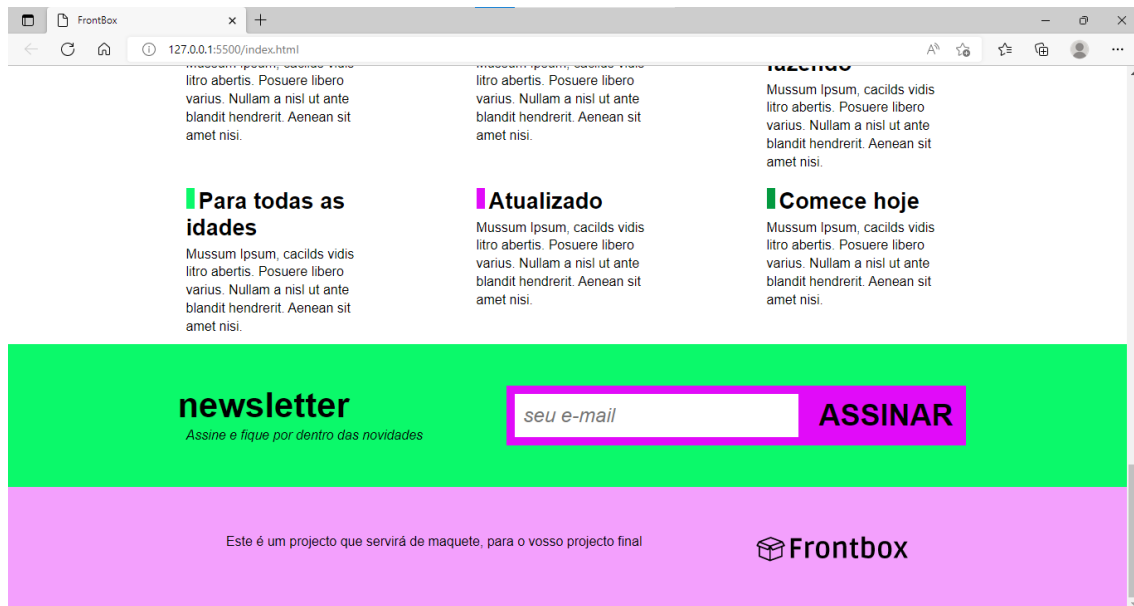
The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes 'Arquivo', 'Editar', 'Seleção', 'Ver', 'Acessar', 'Executar', 'Terminal', and 'Ajuda'. The title bar indicates the active file is 'estilo.css - FrontBox - Visual Studio Code'. The editor has two tabs open: 'estilo.css' and 'index.html'. The 'estilo.css' tab is active, showing CSS code for a newsletter form. The code is as follows:

```
293
294 .newsletter-form_input{
295     flex: 3 1 260px;
296     font-size: 1.5rem;
297     width: 325px;
298     color: block;
299     font-style: italic;
300     font-family: var(--tipo-letra);
301     border: 10px solid var(--rosa-escuro);
302     padding: 10px;
303     margin: 0;
304     border-radius: 0px;
305 }
306
307 .newsletter-form_submit{
308     width: 192px;
309     font-family: var(--tipo-letra);
310     font-size: 2.25rem;
311     font-weight: bold;
312     text-transform: uppercase;
313     background: var(--rosa-escuro);
314     color: black;
315     cursor: pointer;
316     border: none;
317     padding: 15px;
318 }
319
320 /* footer */
321
322 .footer{
323     background-color: var(--rosa-claro);
324     display: flex;
325     justify-content: space-around;
```

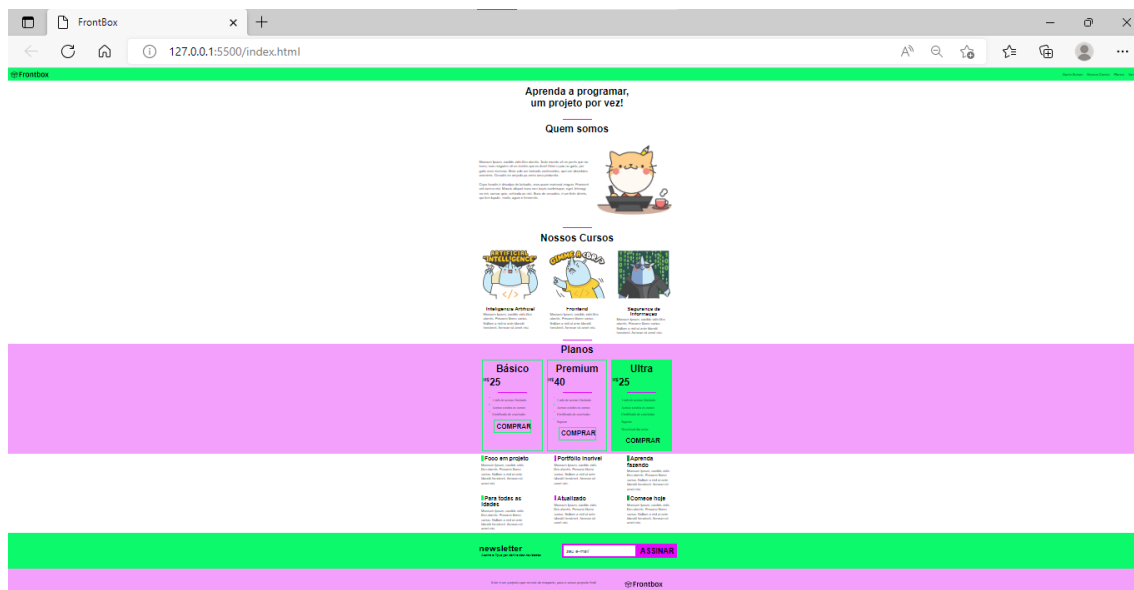
A screenshot of a code editor interface with a dark theme. The top menu bar includes 'Arquivo', 'Editar', 'Seleção', 'Ver', 'Acessar', 'Executar', 'Terminal', and 'Ajuda'. The title bar on the right says 'estilo.css - FrontBox - Vis'. The editor has two tabs: 'estilo.css 1' and 'index.html'. The 'estilo.css' tab is active, showing CSS code. The code includes a class selector (lines 313-317) with properties like 'background', 'color: black', 'cursor: pointer', 'border: none', and 'padding: 15px;'. It also has a comment '/* footer */' (line 320) followed by a '.footer' class selector (lines 322-327) with properties like 'background-color: var(--rosa-claro);', 'display: flex;', 'justify-content: space-around;', and 'padding: 55px calc((100% - 960px)/2);'. Finally, there's a selector for '.footer__copyright, .footer__Logo' (lines 329-332) with 'padding-left: 10px;' and 'padding-right: 10px;'. The left sidebar shows icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The bottom status bar shows a magnifying glass icon.

```
313     background: var(--rosa-escuro);
314     color: black;
315     cursor: pointer;
316     border: none;
317     padding: 15px;
318 }
319
320 /* footer */
321
322 .footer{
323     background-color: var(--rosa-claro);
324     display: flex;
325     justify-content: space-around;
326     padding: 55px calc( (100% - 960px)/2);
327 }
328
329 .footer__copyright, .footer__Logo{
330     padding-left: 10px;
331     padding-right: 10px;
332 }
333
334
335
336
337
338
339
340
341
```

Nesse momento terminamos com os códigos html, css. Vamos ver como está a página. Depois vamos mexer no javascript e podemos visualizar o site.



Essa foi a última parte do retoque no site sobre HTML e CSS. Conseguimos imitar a página do layout.



Nesse momento estamos a ver a página inteira do layout que foi imitado. Agora vamos mexer no **javascript**, de modo a criar um menu que vai ficar dentro do icon. Desse modo vamos ter que mexer no **HTML**, **CSS** e vamos criar um ficheiro **javascript**. Entretanto, agora mexer no **HTML**.

```

<header>
  <div class="menu-bg">
    <nav class="menu">
      <div class="menu_logo">
        <a href="index.html"></a>
      </div>
      <div class="nav">
        <ul class="nav_lista">
          <li class="nav-item"><a href="#somos" class="nav-link">Quem Somos</a></li>
          <li class="nav-item"><a href="#cursos" class="nav-link">Nossos Cursos</a></li>
          <li class="nav-item"><a href="#planos" class="nav-link">Planos</a></li>
          <li class="nav-item"><a href="#vantagem" class="nav-link">Vantagem</a></li>
        </ul>
      </div>

      <div class="mobile-menu-icon">
        <button onclick="menuShow()"></button>
      </div>
    </nav>
    <div class="mobile-menu">
      <ul>
        <li class="nav-item"><a href="#somos" class="nav-link">Quem Somos</a></li>
        <li class="nav-item"><a href="#cursos" class="nav-link">Nossos Cursos</a></li>
        <li class="nav-item"><a href="#planos" class="nav-link">Planos</a></li>
        <li class="nav-item"><a href="#vantagem" class="nav-link">Vantagem</a></li>
      </ul>
    </div>
  </div>
</header>

```

Nesse código temos, mas conteúdo dentro da nossa tags. Além de ter um menu normal, também teremos um **menu mobile**. De modo que quando eu minimizar a minha tela do navegador, ela fica no formato de um telemóvel. Por isso tivemos que criar um **menu-mobile**.

Dentro da tags **button**, criamos um evento javascript chamado **onclick** de modo que quando eu dar um clique no icon, **ele possa abrir-me um menu no formato mobile**. A imagem está no formato svg. E dentro do evento onclick vamos criar uma função chamado **MenuShow()**. Agora vamos mexer no ficheiro css.


```

.menu {
  display: flex;
  justify-content: space-between;
  padding: 1.5rem 6rem;
}

.menu_Logo {
  display: flex;
  align-items: center;
}

.menu_Logo h1 {
  color: #ffff;
}

.nav {
  display: flex;
  align-items: center;
}

.nav .nav_lista {
  display: flex;
  justify-content: center;
  list-style: none;
}

.nav-item {
  margin: 0 15px;
}

.nav-Link {
  text-decoration: none;
  font-size: 1.15rem;
  color: #ffff;
  font-weight: 400;
}

.mobile-menu-icon {
  display: none;
}

.mobile-menu {
  display: none;
}

```

Nesse código a única coisa nova é só a classe **.mobile-menu-icon** e o **.mobile-menu** que tem como propriedade **display: none;** de modo que o meu icon menu não seja mostrado enquanto estiver em tela cheia.

```

@media screen and (max-width: 730px) {
  .nav-bar {
    padding: 1.5rem 4rem;
  }
  .nav-item {
    display: none;
  }
  .Login-button {
    display: none;
  }
  .mobile-menu-icon {
    display: block;
  }
  .mobile-menu-icon button {
    background-color: transparent;
    border: none;
    cursor: pointer;
  }
  .mobile-menu ul {
    display: flex;
    flex-direction: column;
    text-align: center;
    padding-bottom: 1rem;
  }
  .mobile-menu .nav-item {
    display: block;
    padding-top: 1.2rem;
  }
  .mobile-menu .Login-button {
    display: block;
    padding: 1rem 2rem;
  }
  .mobile-menu .Login-button button {
    width: 100%;
  }
  .open {
    display: block;
  }
}

```

Agora aqui temos um css no formato mobile com um tamanho de tela de **730px**. Todos os códigos que vamos aplicar nesse momento, servirá apenas para efeito numa tela de 730px. Para isso criamos um media e aplicamos um tamanho para ela. E nesse código a única coisa que mexemos foi o menu e colocar no formato de telemóvel. Quando aplicamos uma media screen, vamos apenas seleccionar os elementos que queremos colocar no formato mobile, Porque que não podemos colocar tudo dentro media? Porque já usamos o flexbox para organizar os elementos. Agora vamos mexer no ficheiro javascript. Porque já criamos uma função chamado **menushow()**.

```

JS app.js > menuShow
1  function menuShow() {
2      let menuMobile = document.querySelector('.mobile-menu');
3      if (menuMobile.classList.contains('open')) {
4          menuMobile.classList.remove('open');
5          document.querySelector('.icon').src = "./img/menu_white_36dp.svg";
6      } else {
7          menuMobile.classList.add('open');
8          document.querySelector('.icon').src = "./img/close_white_36dp.svg";
9      }
10 }

```

Nesse código javascript criamos uma função com nome **menuShow()**. Onde dentro dessa função criamos uma variável com nome **menuMobile**, de modo que ela seleciona o documento que ela vai precisar manipular. Nesse caso é a classe **.mobile-menu** que encontra-se em uma tag html. Depois disso, criamos uma estrutura de condição onde a variável **menuMobile** vai receber um **classList** que está no css de um tipo booleano, que vai retornar verdadeiro ou falso. Agora vamos explicar o que está acontecendo dentro da estrutura de condição.

```
if (menuMobile.classList.contains('open'))
```

Se o **menuMobile** tiver uma **classList** **open** no meu css e estiver aberto, assim que demos um clique, então vamos remover ou fechar o menu.

```
menuMobile.classList.remove('open');
```

Se não tiver um menu **open**, ele vai adicionar.

```

else {
    menuMobile.classList.add('open');
    document.querySelector('.icon').src =
"./img/close_white_36dp.svg";
}

```





Esse é o resultado do site aplicando HTML, CSS e JAVASCRIPT.