

# Introdução ao CSS

## Conceito de CSS

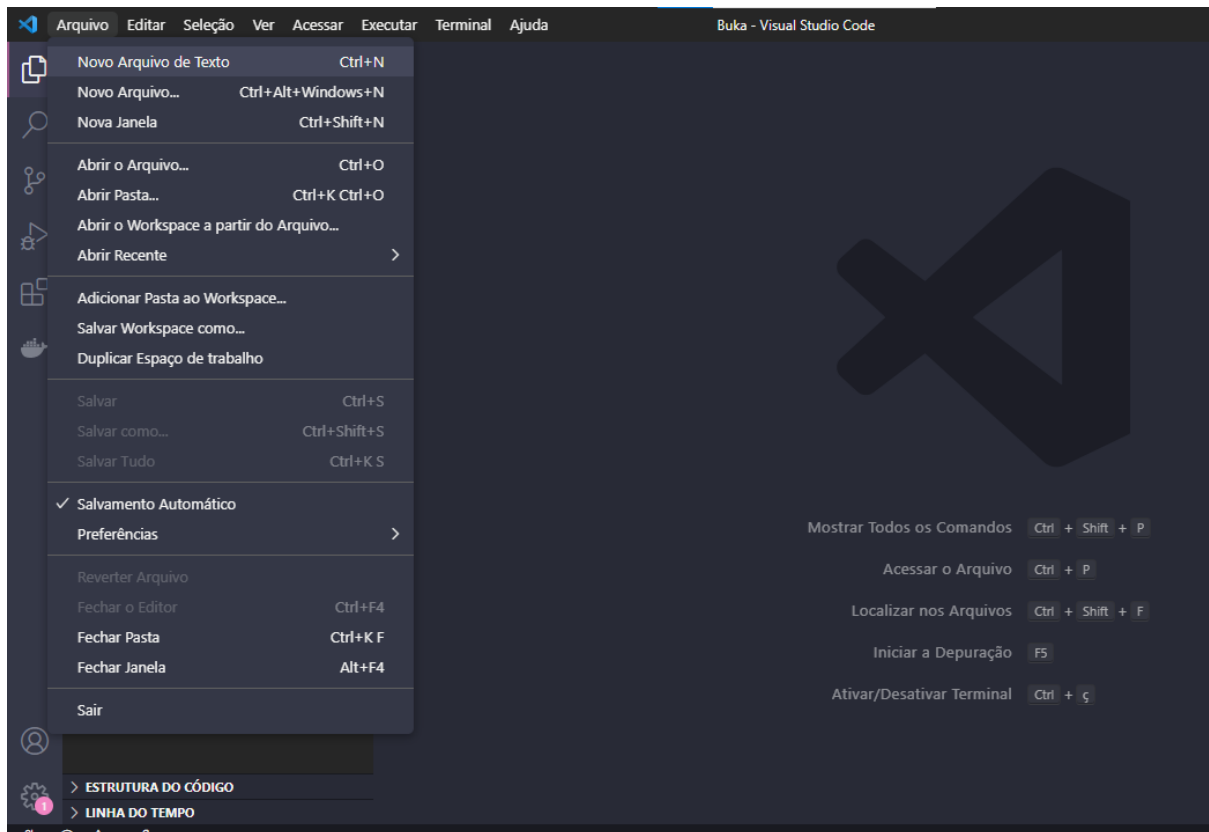
CSS é a sigla para o termo em inglês *Cascading Style Sheets* que, traduzido para o português, significa Folha de Estilo em Cascatas. O CSS é fácil de aprender e entender e é facilmente utilizado com as linguagens de marcação HTML.

A relação entre HTML e CSS é bem forte. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

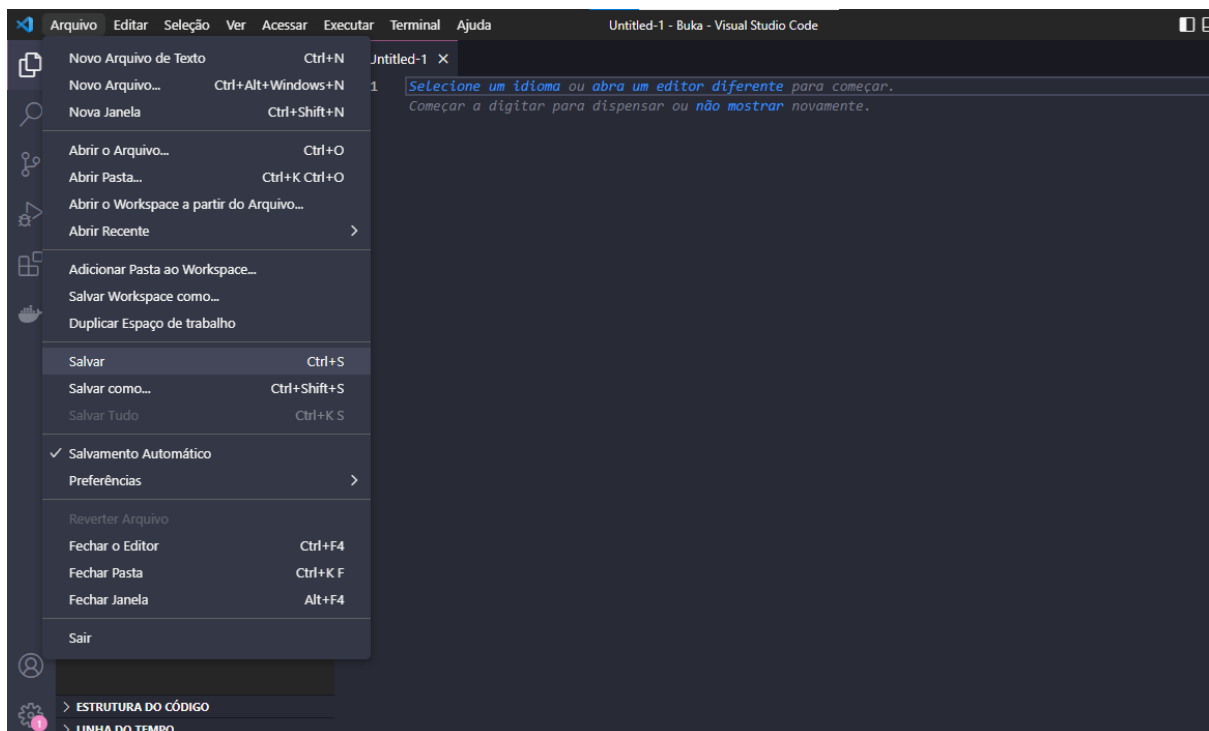
CSS não é tecnicamente uma necessidade, mas provavelmente você não gostaria de olhar para um site que usa apenas HTML, pois isso pareceria completamente abandonado.

### Como Criar uma Folha de Estilo

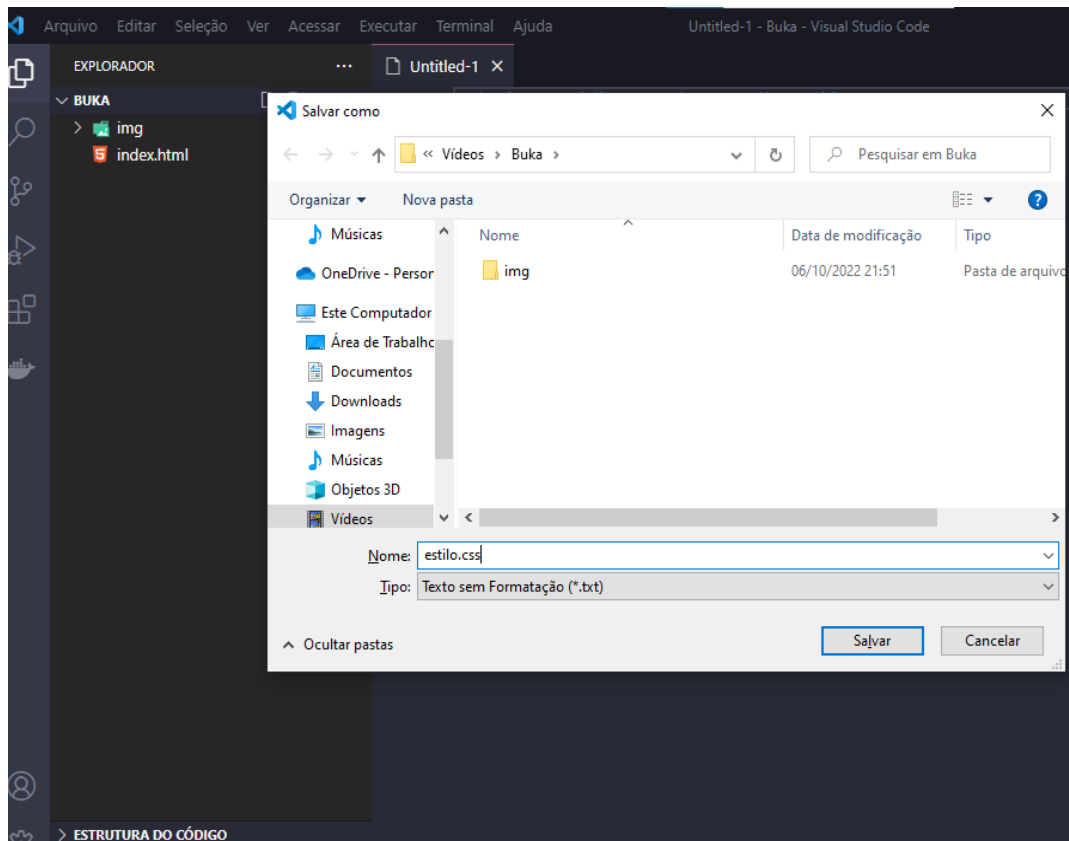
Para criar uma folha de estilos é bem simples. Basta abrir seu editor de códigos, como o visual studio code, e em seguida demos um clique no menu “**Arquivo**” do visual studio code.



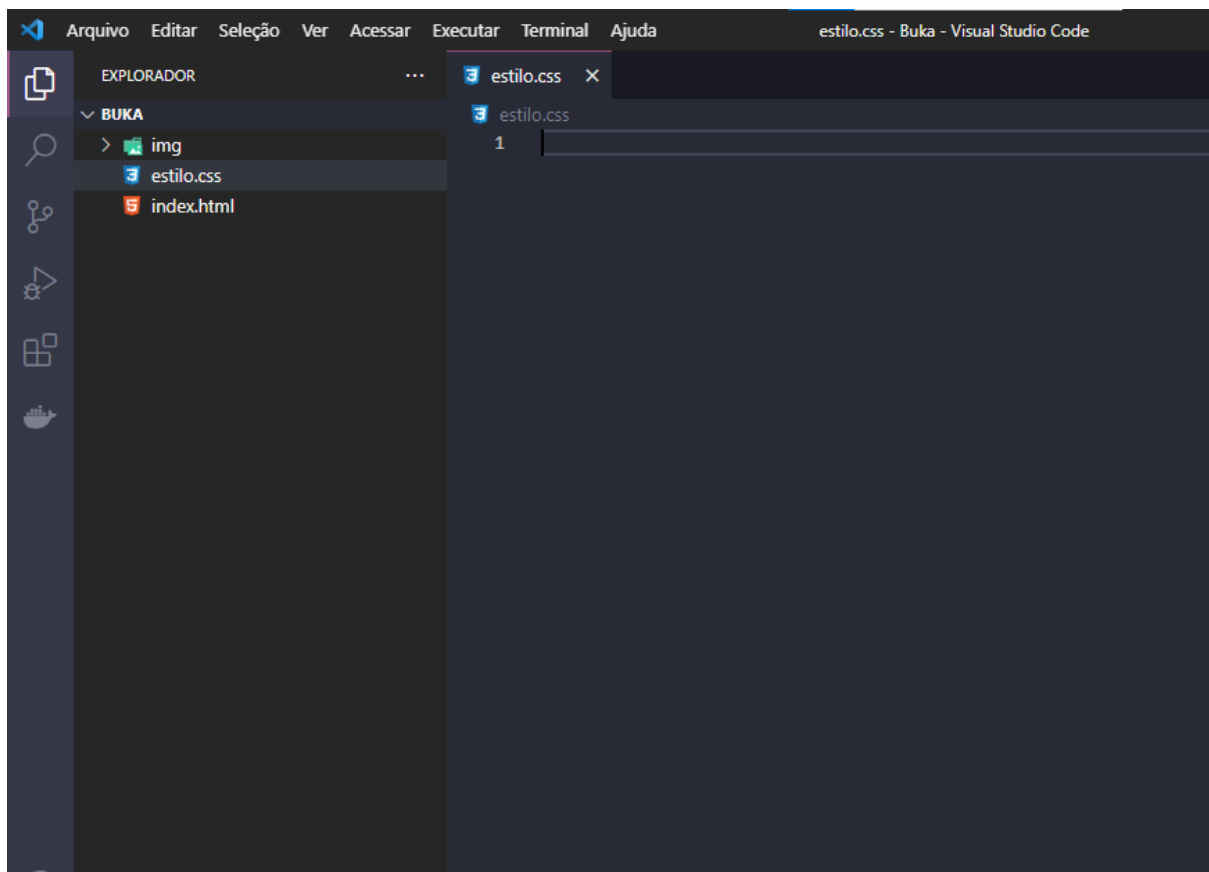
Assim que demos um clique no menu “**Arquivo**”, vai abrir uma aba de menu, devemos clicar no menu “**Novo arquivo de texto**” ou usar combinação de teclado **CTRL + N**, vai abrir uma nova folha com nome untitled..



Para salvar a nossa folha e darmos o nome de **estilo.css**, precisamos salvar o ficheiro. Devemos usar a combinação de teclado **CTRL + S** ou devemos ir no **menu Arquivo** e depois dentro do **menu arquivo** vamos encontrar o menu salvar.



Assim que clicarmos no menu salvar, vai abrir essa página. devemos dar o nome de **estilo.css** e depois clicar no menu **salvar**.



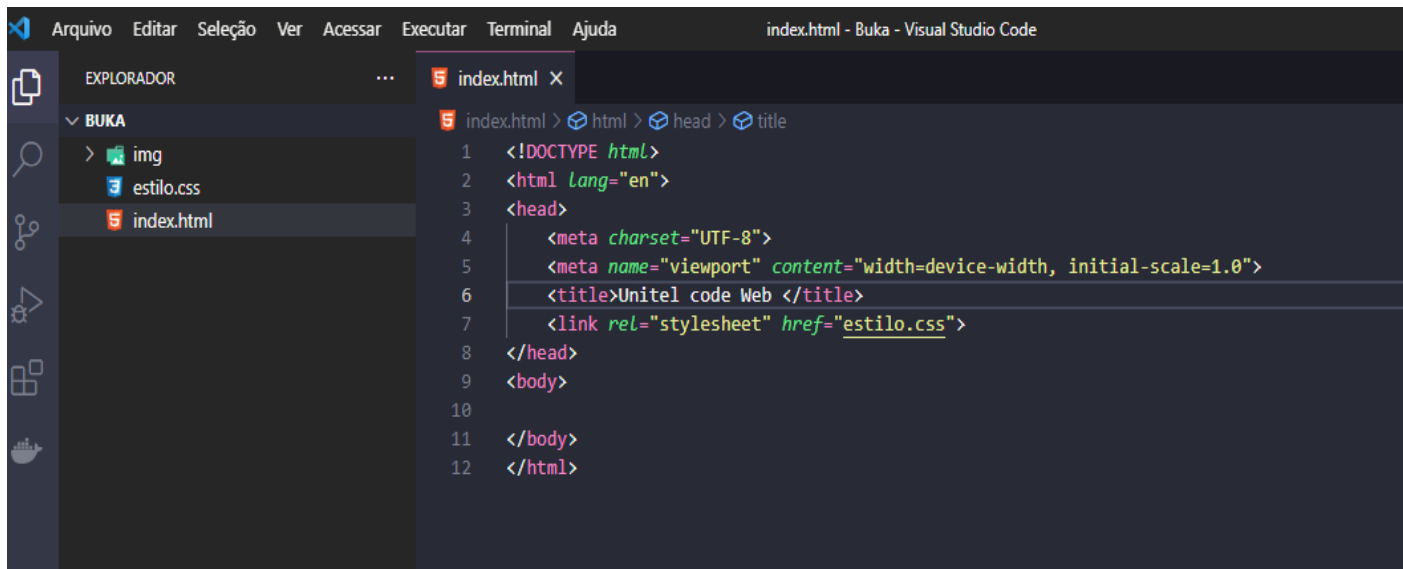
De acordo com a imagem, já conseguimos visualizar o ficheiro css criado.

## Como integrar o CSS no HTML?

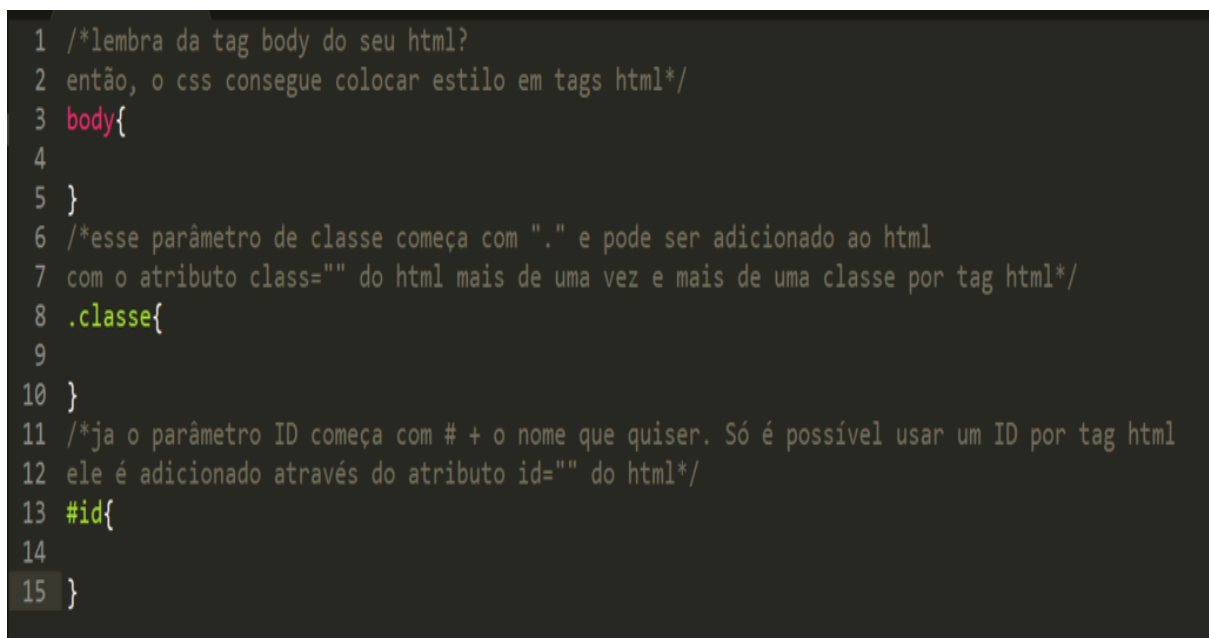
Após salvar como `estilo.css` precisamos colocar o arquivo em nosso arquivo HTML, porque é ele quem vai carregar o CSS, o CSS sozinho não faz site! E é com *tags* de referência que você avisa ao HTML que existe um arquivo de estilo para ele, veja:

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

Esta *tag* vai indicar ao HTML que tem um arquivo CSS para ser adicionado à página. Isto deve ser adicionado dentro da *tag* `<head> </head>` do seu HTML, como pode ver nesse print abaixo.



Agora abra o documento **estilo.css** com o Visual Studio Code, está na mesma pasta Buka, onde está também o html e a pasta img.



A partir desta última imagem, vamos entender como o CSS funciona com três parâmetros básicos:

## Elementos Básicos do CSS

Os elementos são tags do HTML, #id e .classe.

## 1. Tags HTML

```
body  
h1  
h2  
h3  
h4  
h5  
h6  
span  
p
```

## 2. Classes CSS

Criar e usar uma classe CSS é simples. No documento CSS basta colocar .nome-da-classe-que-deseja . Abaixo criamos a classe título:

```
.título{  
    font-size: 3rem; /*tamanho da fonte do título*/  
    color: #333; /*esta é a cor do texto*/  
    text-align: center; /*o alinhamento do texto que carregar esta classe*/  
}
```

Note que existe um ponto (.) antes do nome da classe. Isso faz com que o CSS entenda que se está criando uma classe para ser usada no HTML.

Para se adicionar essa classe que criamos dentro do CSS, basta que na *tag* HTML tenha o parâmetro class="".

**Observação:** dentro das aspas, não precisa usar o ponto (.) antes da classe pois o nome *class* no HTML já avisa para o CSS que é para ele buscar pelo nome da classe com “.”

```
<h1 class="titulo">Melhor site</h1>
```

Lembrando que uma *tag* HTML, como h1 ou qualquer outra, pode ter mais de uma classe, basta separá-las com espaço como mostra no exemplo abaixo:

```
<h1 class="titulo fonte-grande sublinhado">Melhor site</h1>
```

Nesta tag foram usadas três classes: “titulo”, “fonte-grande” e “sublinhado”. Mas para que todas estas classes tenham efeito é preciso criá-las no CSS. Veja como criamos estas classes:

```
.titulo{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
.fonte-grande{
  font-size: 40px; /*tamanho da fonte do seu título ou texto em pixels*/
}
.sublinhado{
  text-decoration: underline; /*ess propriedade faz com que seu texto fique sublinhado*/
}
```

Quando abrirmos nosso HTML o título deverá ficar assim:

**Melhor site**

### 3- Criando ID's CSS:

O uso dos ID's em CSS é parecido com as classes, mas as principais diferenças são que os id's são identificados com # e só podem ser usados um id por tag html. Para adicionar um ID ao HTML, adicione como parâmetro id="".

Veja que a quando usamos no HTML, também excluimos o # do id da mesma forma que excluimos o . da classe.

```
.titulo{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
#botao-bonito{
  color:white; /*cor da fonte*/
  padding-top: 15px; /*espaço superior entre o texto e o elemento*/
  padding-bottom: 15px; /*espaço inferior entre o texto e o elemento*/
  padding-right: 35px; /*espaço a direita entre o texto e o elemento*/
  padding-left: 35px; /*espaço a esquerda entre o texto e o elemento*/
  background-color:green; /*cor verde para o fundo do elemento, nesse caso o botão*/
}

<h1 class="titulo fonte-grande sublinhado">Melhor site</h1>
<a id="botao-bonito" href="link-para-alguma-pagina">clique nesse botão</a>
```

Dessa forma, o botão deverá ficar assim:





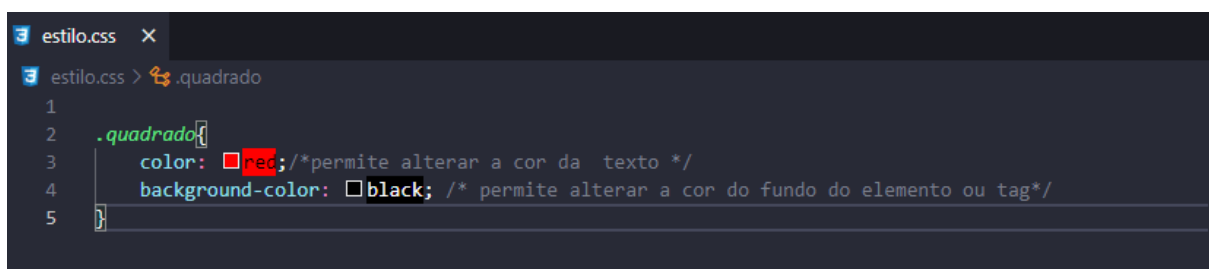
## PROPRIEDADES CSS3

Nesse momento vamos criar ficheiros HTML e CSS, de modo a ver como está a funcionar o css.



```
index.html x
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7   <link rel="stylesheet" href="estilo.css">
8 </head>
9 <body>
10
11 <div class="quadrado">
12   <h1>Seja bem vindo</h1>
13 </div>
14
15 </body>
16 </html>
```

De acordo com a imagem, já temos o ficheiro html que tem como nome de **index.html** e também o ficheiro css que tem como nome **estilo.css**.



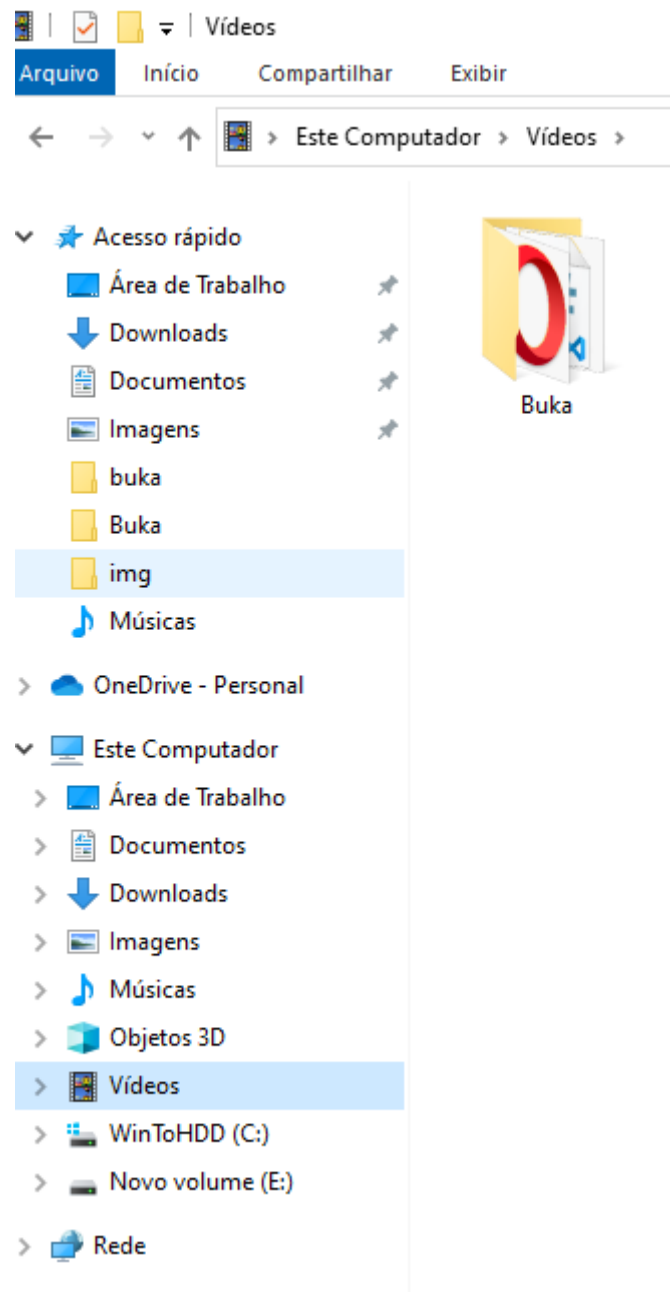
```
estilo.css x
estilo.css > .quadrado
1
2 .quadrado{
3   color: red; /*permite alterar a cor da texto */
4   background-color: black; /* permite alterar a cor do fundo do elemento ou tag*/
5 }
```

No ficheiro estilo.css usamos o selector de class para que tem o nome de quadrado, nesse quadrado aplicamos uma cor no texto e demos uma cor no fundo da minha tag div que tem como nome da class quadrado.

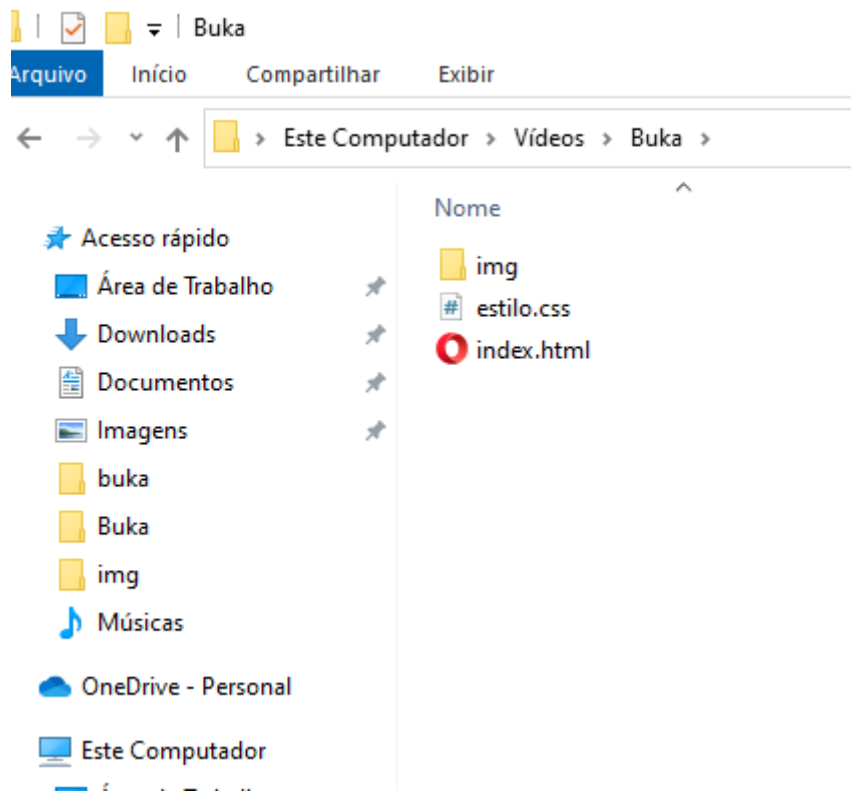
**background-color:** A propriedade background-color permite configurar uma cor de fundo a um elemento, podemos aplicar praticamente a qualquer elementos que desejamos mudar a cor do fundo, body, p, div, h1 a h6, header, section, etc.

**Color:** A propriedade permite configurar cor do texto de um elemento, como parágrafo e outros elementos.

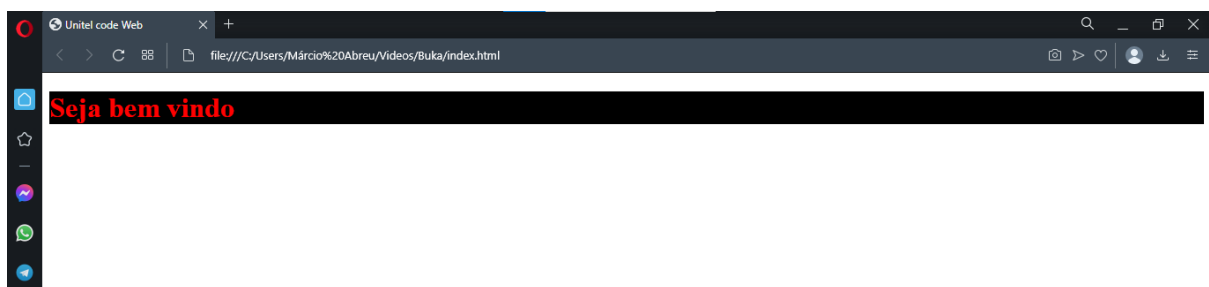
Agora vamos ir na pasta onde está o nosso ficheiro html e css.



A minha pasta Buka está no vídeo do meu computador. Devo dar um clique na pasta Buka, para eu poder os meus ficheiros html e css.



Agora devemos dar um clique no ficheiro index.css, para que a nossa página abra no navegador.



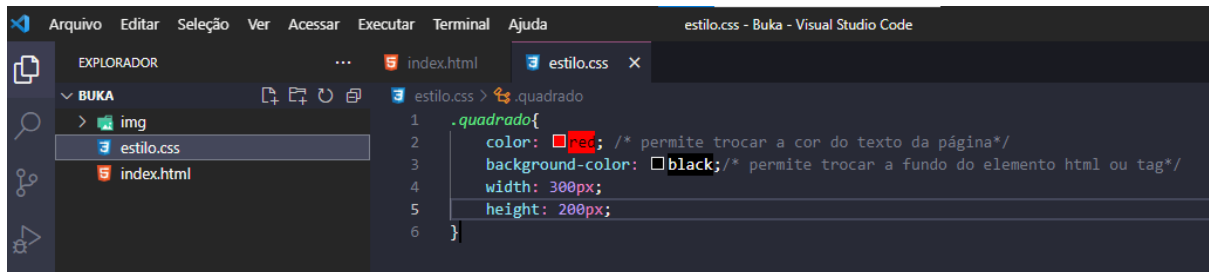
Esse é o resultado do que nós fizemos até agora. O nosso fundo está ocupado a tela inteira, porque não definimos um tamanho no fundo.

**width – largura** : A propriedade width define a largura de um elemento, pode usar várias unidades de medida, px, %, em, rem etc.

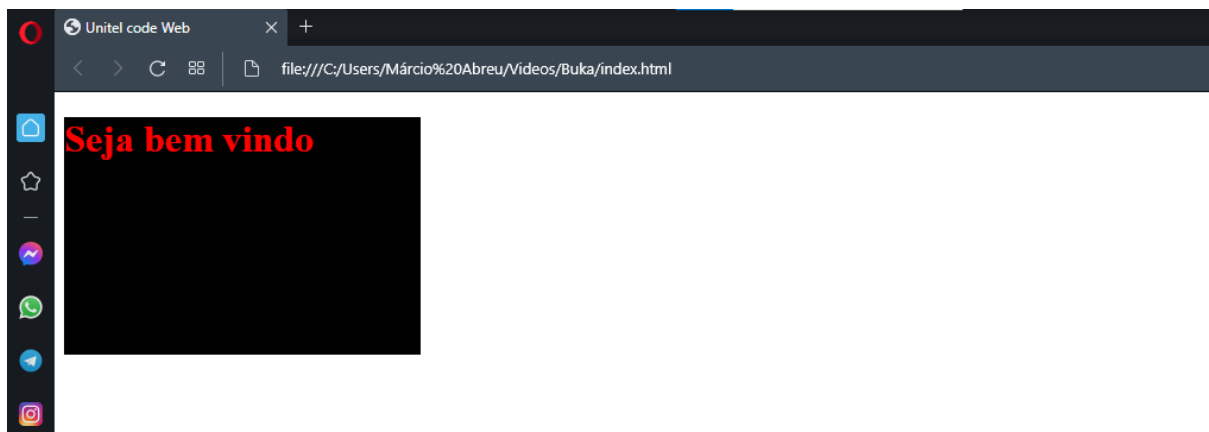
**height - Altura** – Altura Já a propriedade height define a altura de um determinado elemento, vamos usar o mesmo código anterior.

## max-width & min-width

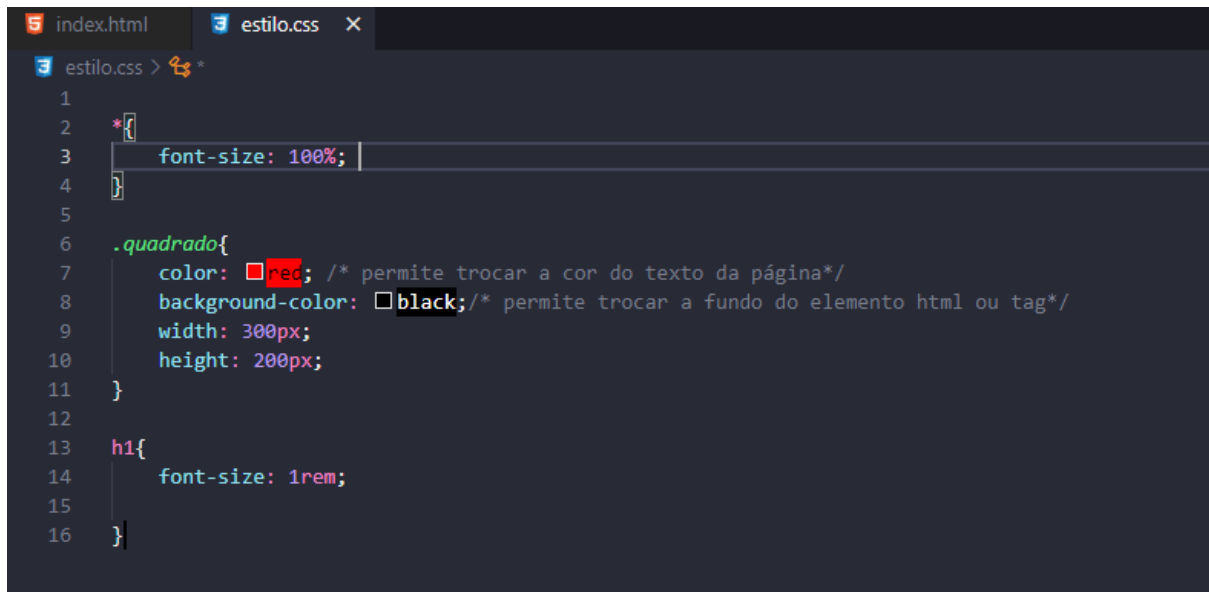
Podemos configurar um valor para largura máxima e mínima, em caso de uso de unidades de medida relativas como %, o elemento será aumentado ou diminuído de acordo com a largura da tela, mas nunca será menor que min-width ou maior que max-width. Mas essa propriedade é mais utilizada quando estivermos a trabalhar com página responsiva.



Logo em seguida, vamos abrir o nosso navegador e ver se sofreu alguma alteração, na nossa página.

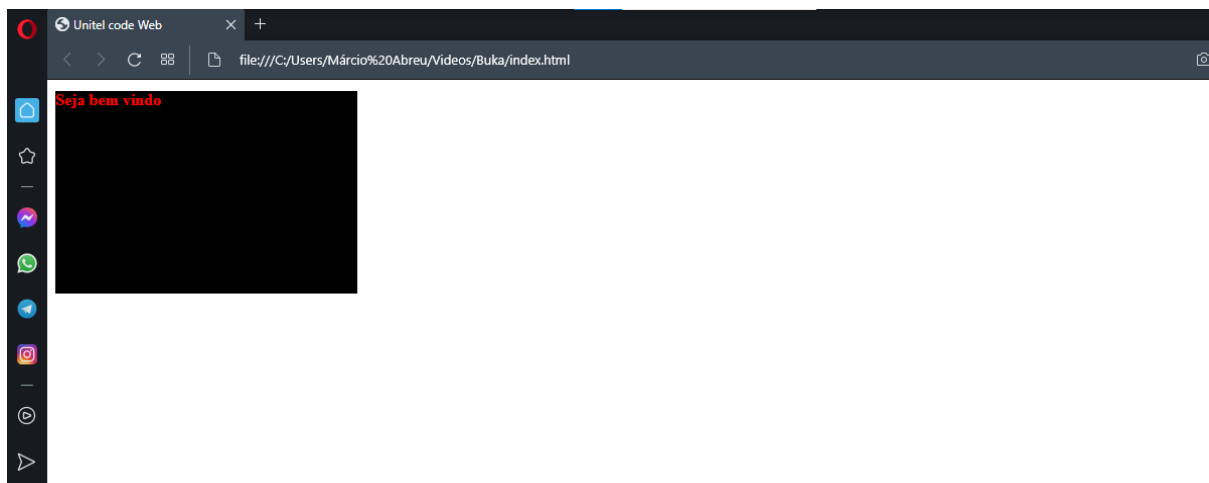


Podemos notar que diminuimos a largura, que ocupa um tamanho de 300px e uma altura de 200px. Caso queira usar um tamanho que ocupa a tela inteira usa a unidade %. A tela do nosso computador é de 100%. As outras unidades de medida como **rem** e **em**, são usadas quando estamos a usar o tamanho de um título ou parágrafo. 1 rem é igual ao tamanho padrão da fonte da raiz(root) do documento, normalmente 16px, logo 1 rem é igual à 16px. Já o **em** Essa medida se comporta semelhante ao **rem**, a diferença dela é que 1 em se baseia no tamanho do elemento pai de um parágrafo.



```
1
2 *{
3   font-size: 100%;
4 }
5
6 .quadrado{
7   color: red; /* permite trocar a cor do texto da página*/
8   background-color: black; /* permite trocar a fundo do elemento html ou tag*/
9   width: 300px;
10  height: 200px;
11 }
12
13 h1{
14   font-size: 1rem;
15 }
16 }
```

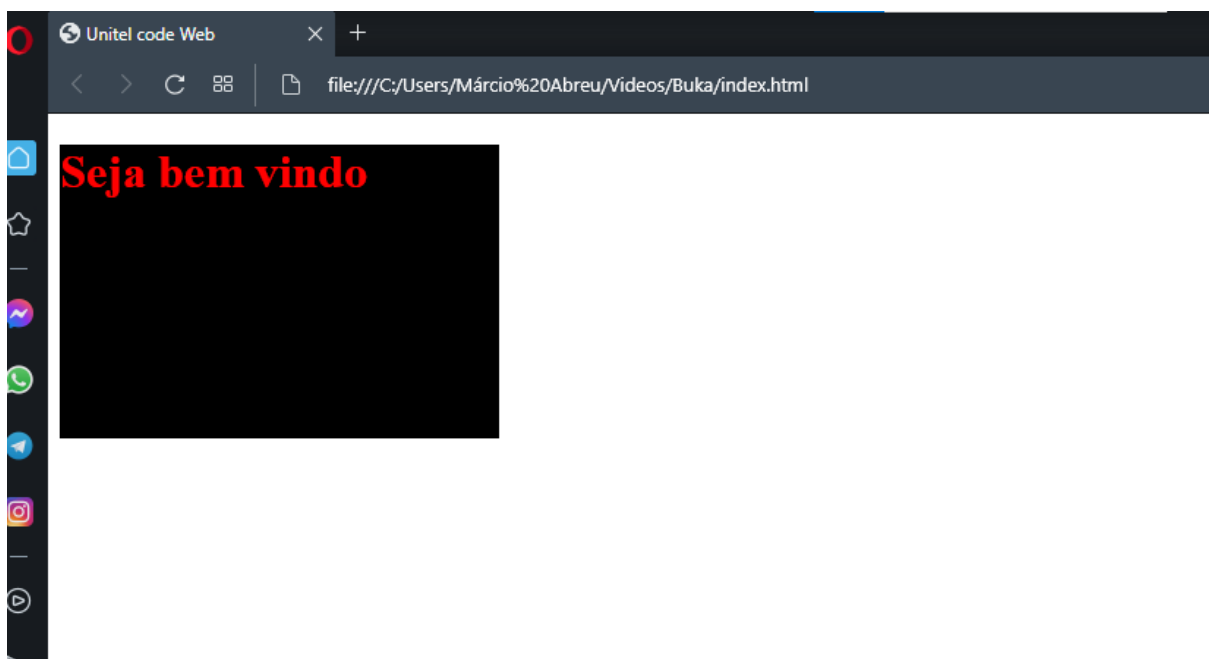
De acordo com a imagem estamos ver a nossa tag raiz(\*) ou global que tem um tamanho da fonte de letra de 100%, que quer dizer que todos os elementos vai receber um tamanho de 16px. Se nós prestar bem atenção, o nosso título está com um tamanho de 1rem, que quer dizer que tem um tamanho de 16px. se nós colocarmos 2rem, quer dizer que ele será 32px. porque ele faz uma multiplicação entre o elemento raiz com a tag que pretendemos dar um tamanho.



Podemos perceber que o tamanho do nosso texto diminuiu. Para nós aumentar, basta colocar uma **font-size: 2rem.** em outra palavras , esse 2rem, será 32px.

```
index.html  estilo.css x
estilo.css > h1
1
2  *{
3    font-size: 100%;
4  }
5
6  .quadrado{
7    color: red; /* permite trocar a cor do texto da página*/
8    background-color: black; /* permite trocar a fundo do elemento html ou tag*/
9    width: 300px;
10   height: 200px;
11 }
12
13 h1{
14   font-size: 2rem;
15 }
16 }
```

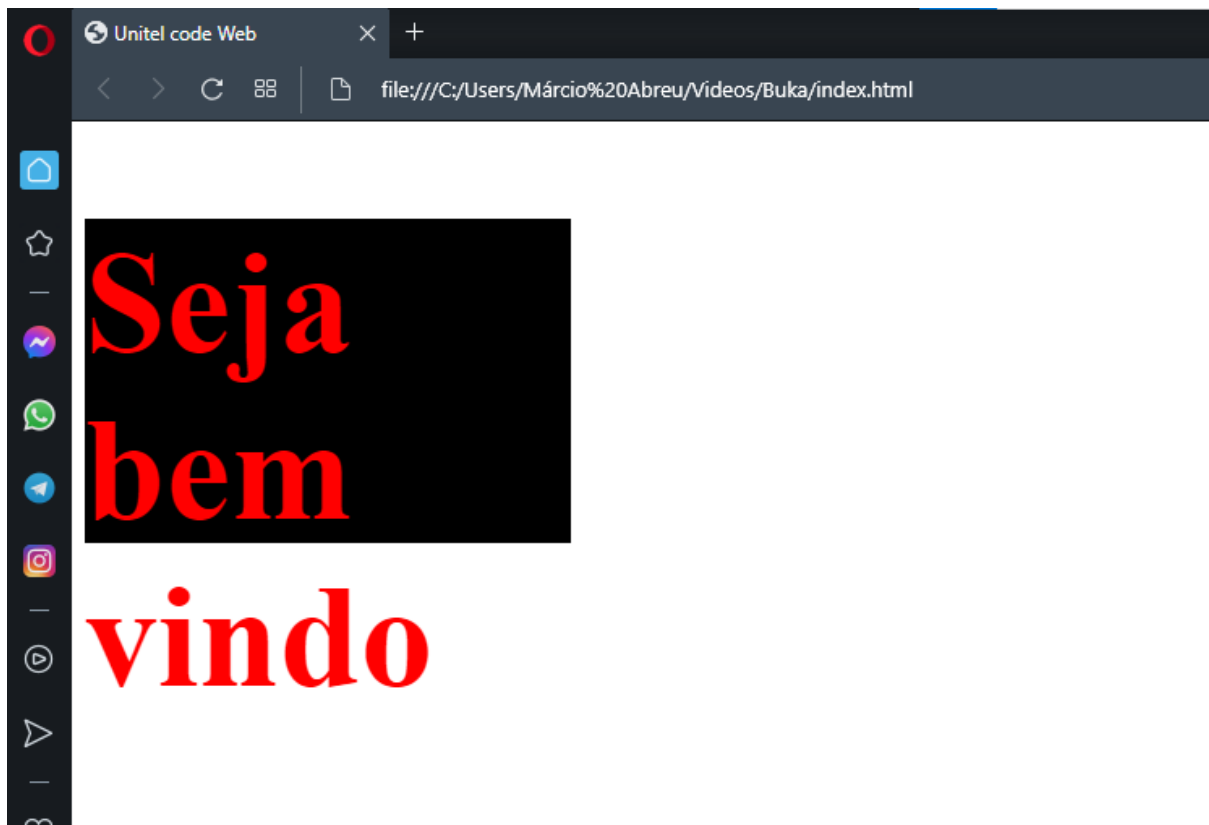
Agora vamos ver o resultado.



O mesmo acontece com a unidade de medida , a única que muda é que ele não depende do elemento raiz, mas sim do seu pai.

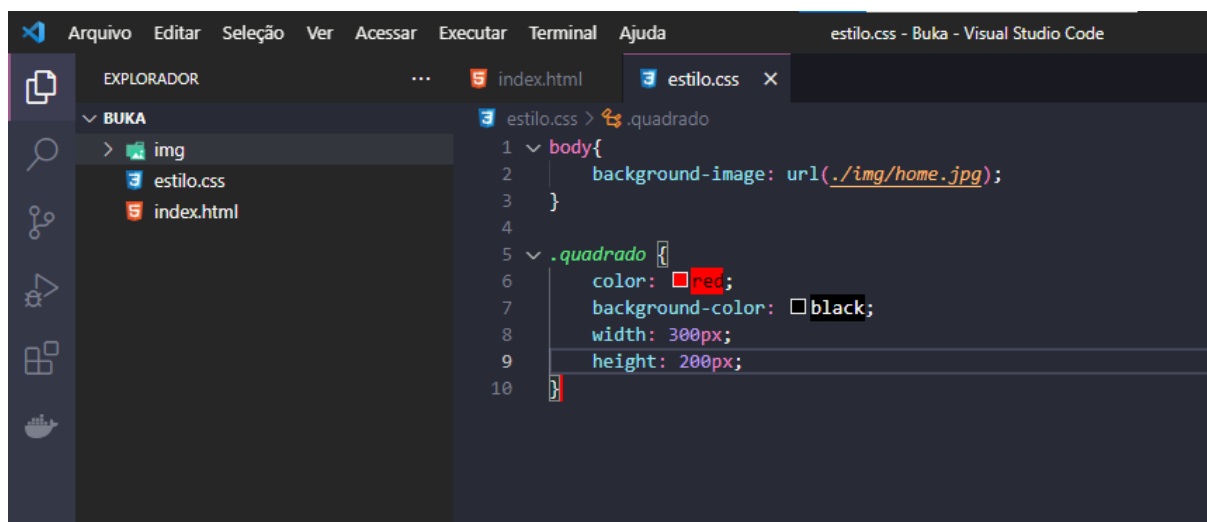
```
index.html  estilo.css  X
estilo.css > h1
1
2  *{
3    font-size: 100%;
4  }
5
6  .quadrado{
7    color: red; /* permite trocar a cor do texto da página*/
8    background-color: black; /* permite trocar a fundo do elemento html ou tag*/
9    width: 300px;
10   height: 200px;
11   font-size: 30px
12 }
13
14 h1{
15   font-size: 3em;
16 }
```

De acordo com a imagem, podemos perceber que a classe é elemento pai do h1. Nesse caso o elemento h1 é o elemento filho, quando aplicarmos uma medida no elemento pai, o elemento filho também vai obedecer o mesmo tamanho do elemento pai que é de font-size: 30px. para que nós mexermos no elemento filho, devemos usar a unidade de medida (**em**).



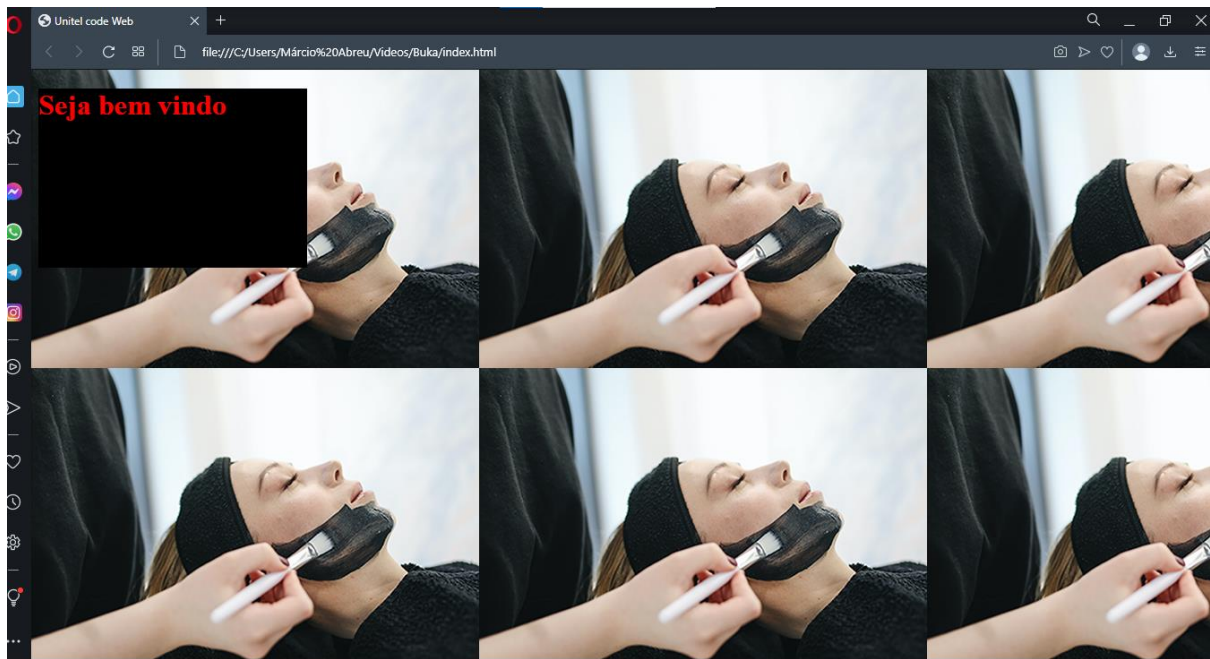
De acordo com o resultado do navegador, podemos ver que o título aumentou. Além do quadrado ter um tamanho de 30px, quando aplicarmos um tamanho título de 3em, ele multiplicou  $30 \times 3 = 90\text{px}$ . Isso quer dizer que o nosso título terá um tamanho de 90px.

**background-image:** Caso precise adicionar uma imagem de fundo basta usar background-image, veja o código a seguir onde inserimos a imagem “cfb.jpg” como imagem de fundo da página.



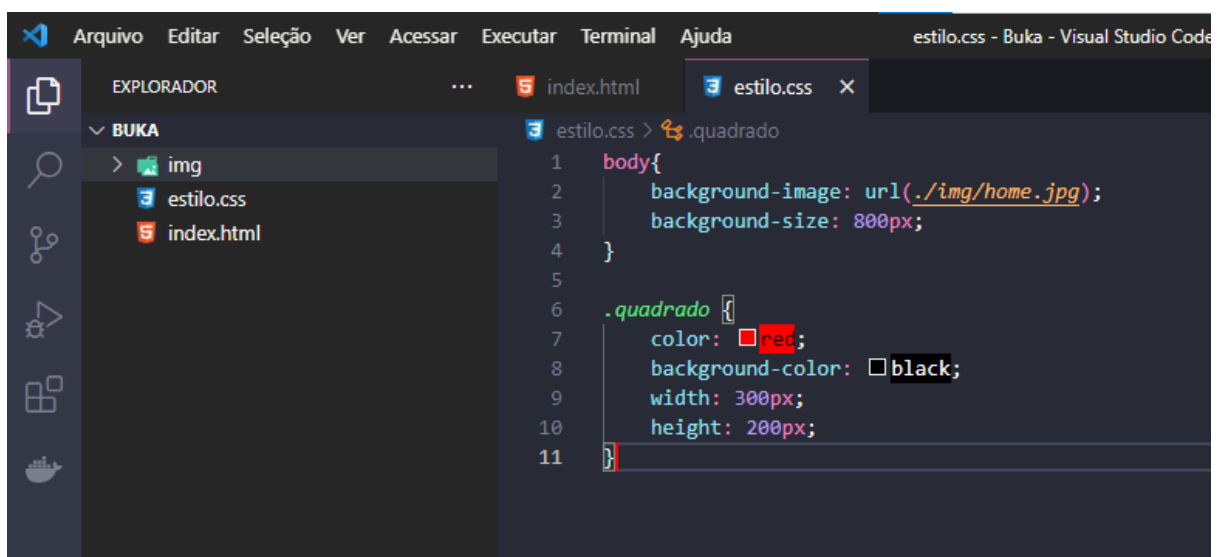
Assim como estamos a ver na imagem, temos um body que é o corpo do html está a receber uma imagem de fundo. isso quer dizer que toda nossa pagina terá essa imagem. a **url()** é o caminho onde está a nossa pasta imagem.



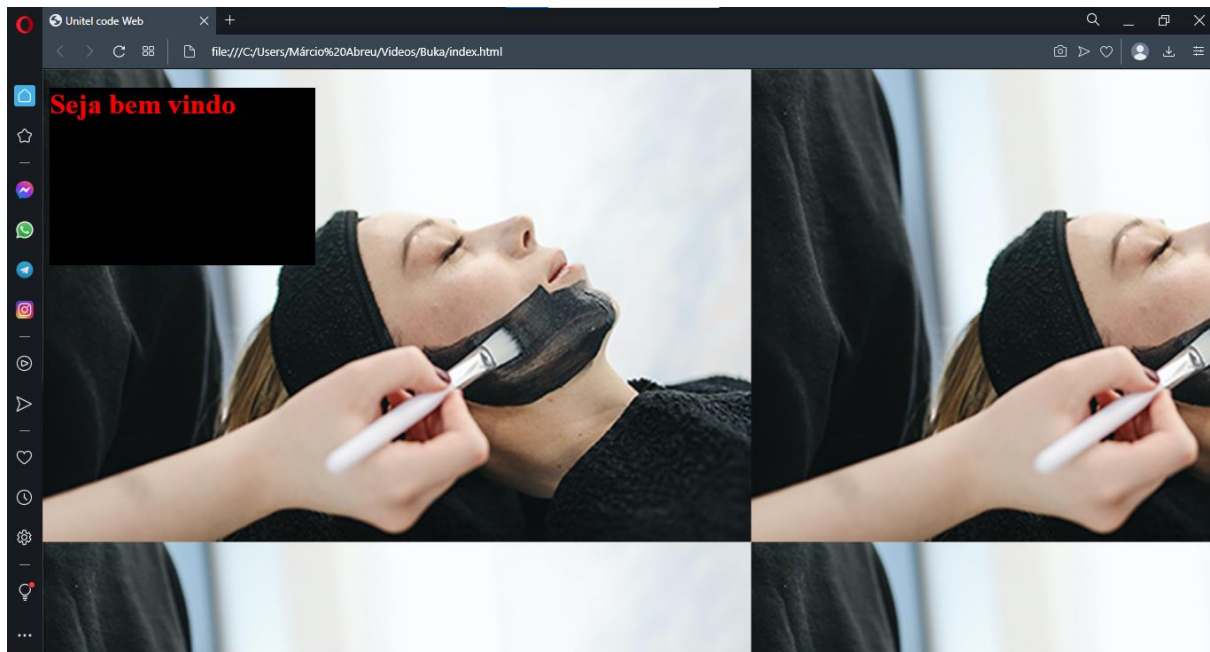


Esse é o resultado por nós colocar a imagem no fundo. Isso não quer dizer que só podemos colocar imagem de fundo no body, podemos colocar em qualquer tag html.

**background-size:** Com a propriedade background-size podemos configurar o tamanho da imagem de fundo. Adicionando esta propriedade podemos controlar o tamanho da largura e altura da imagem, em nosso código de exemplo usamos o valor 800px para definir a largura, quando usamos somente um valor para largura, automaticamente a altura aumenta ou diminui na proporção devida.



Agora vamos ver o resultado no navegador.

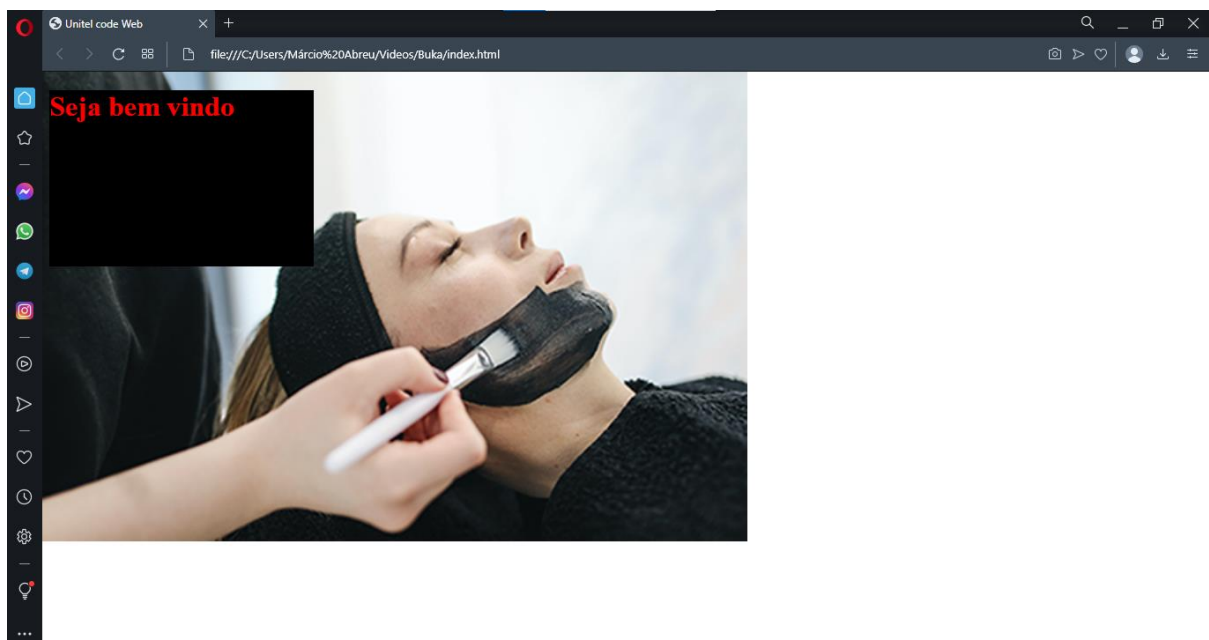


Podemos notar que o tamanho da nossa imagem aumentou, porque aplicamos um tamanho de 800px.

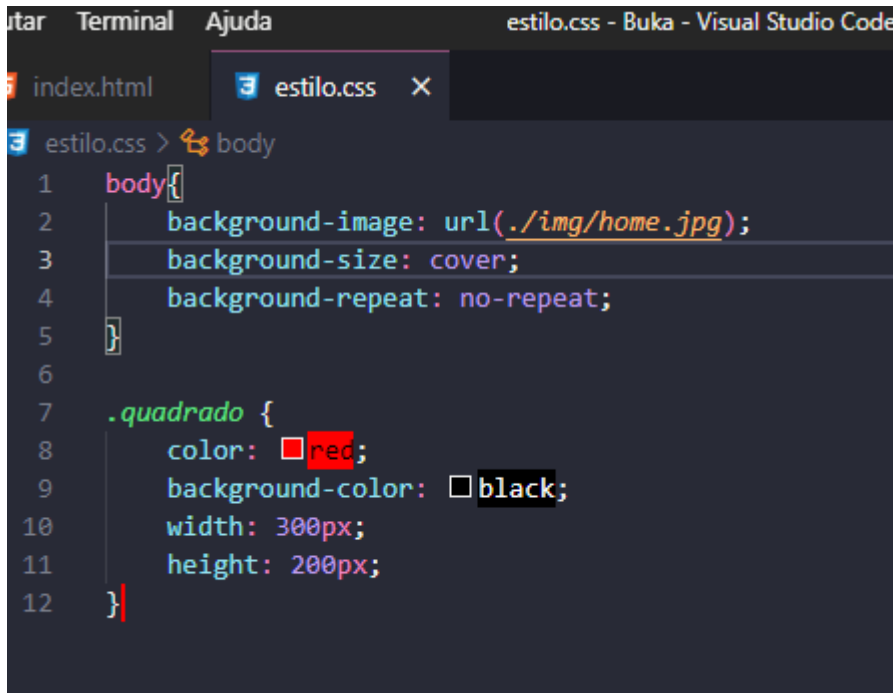
**background-repeat:** Esta propriedade configura a repetição da imagem, por padrão a imagem de fundo se repete até que preencha toda área da janela, mas podemos controlar se vai repetir somente na horizontal, vertical ou sem repetição.

```
index.html | estilo.css x
estilo.css > body
1 body{
2   background-image: url(./img/home.jpg);
3   background-size: 800px;
4   background-repeat: no-repeat;
5 }
6
7 .quadrado {
8   color: red;
9   background-color: black;
10  width: 300px;
11  height: 200px;
12 }
```

Quando aplicamos um no-repeat, fica apenas uma única. A imagem só tende a repetir quando é pequena, então ela repete, para ocupar inteira do computador.

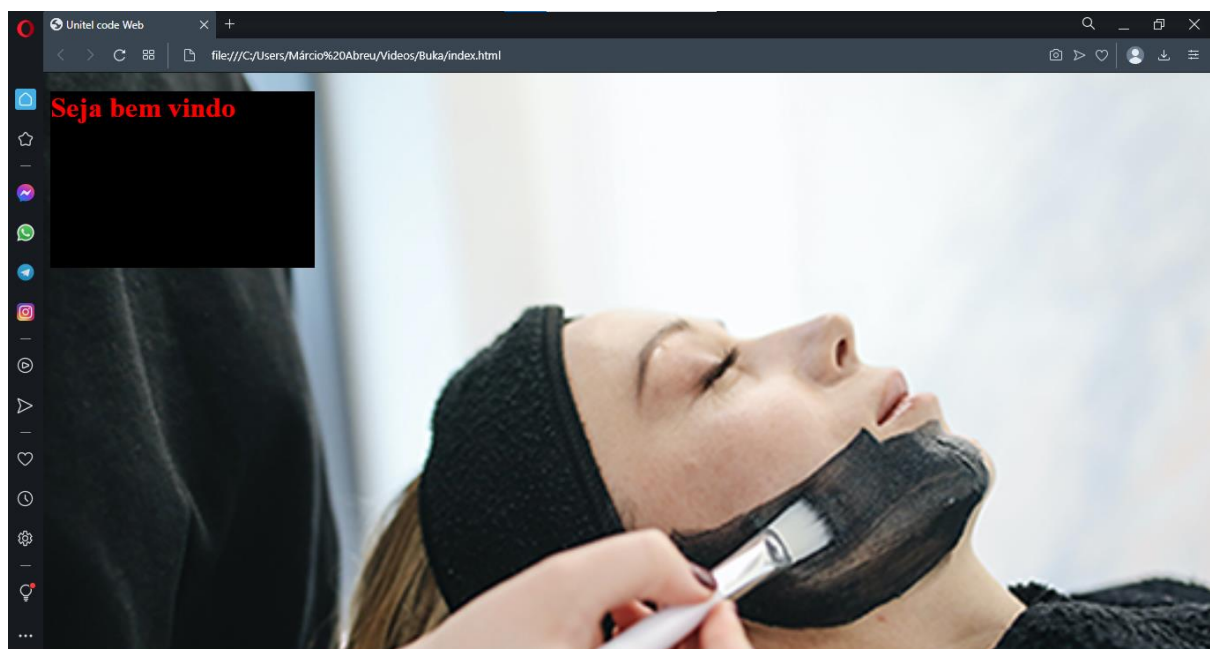


Podemos notar que só tem agora uma única imagem. Se nós queremos colocar que a imagem ocupa a tela inteira devemos usar a propriedade **cover**.

The image shows a Visual Studio Code editor window with the file 'estilo.css' open. The code defines styles for the 'body' and a class '.quadrado'. The 'body' rule sets the background image to './img/home.jpg', background size to 'cover', and background repeat to 'no-repeat'. The '.quadrado' rule sets the color to 'red', background-color to 'black', width to '300px', and height to '200px'.

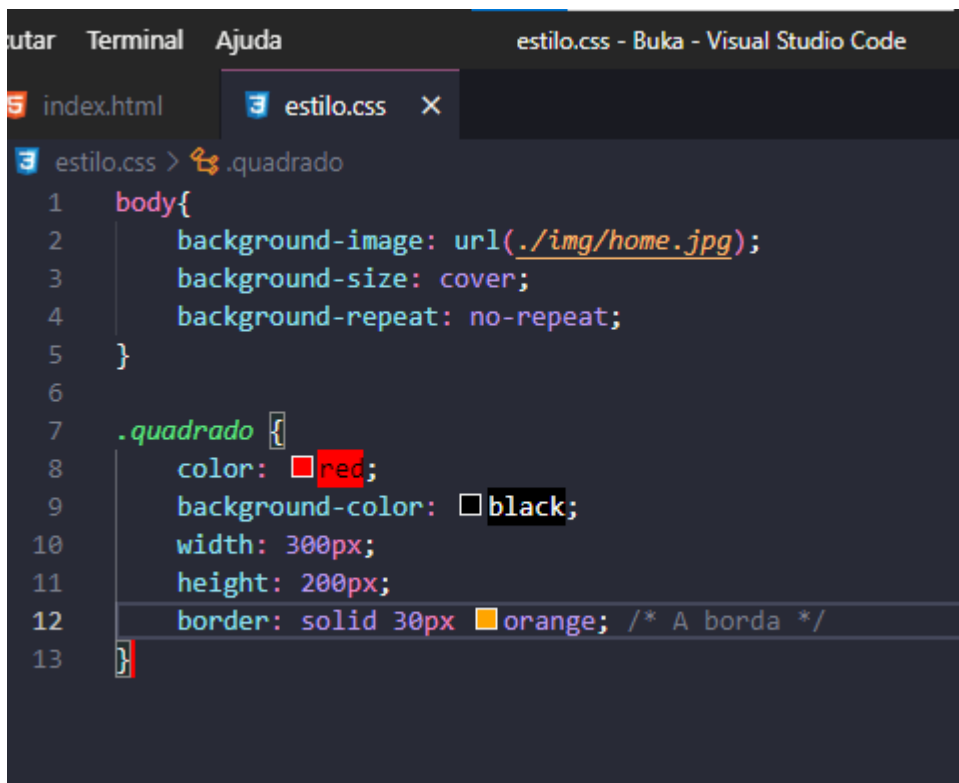
```
1 body{  
2     background-image: url(./img/home.jpg);  
3     background-size: cover;  
4     background-repeat: no-repeat;  
5 }  
6  
7 .quadrado {  
8     color: red;  
9     background-color: black;  
10    width: 300px;  
11    height: 200px;  
12 }
```

Agora vamos ver o resultado no navegador.



**border – bordas / linhas de contorno** : A metapropriedade border define as configurações das bordas do elemento, podemos definir o estilo da borda, largura e cor de forma simples.

A sintaxe é a seguinte: border: estilo largura cor;

A screenshot of the Visual Studio Code editor interface. The top bar shows 'estilo.css - Buka - Visual Studio Code'. Below it, there are two tabs: 'index.html' and 'estilo.css'. The 'estilo.css' tab is active, showing the following CSS code:

```
1  body{
2      background-image: url(./img/home.jpg);
3      background-size: cover;
4      background-repeat: no-repeat;
5  }
6
7  .quadrado {
8      color: red;
9      background-color: black;
10     width: 300px;
11     height: 200px;
12     border: solid 30px orange; /* A borda */
13 }
```

Existem vários estilos que podemos aplicar, veja a seguir.

**dotted** – Bordas pontilhadas

**dashed** – Bordas tracejadas

**solid** – Bordas com linha sólida / contínua

**double** – Bordas com linhas duplas

**groove** – Bordas em estilo 3D

**ridge** – Bordas em estilo 3D com as cores ao contrário de groove

**inset** – Efeito 3D no estilo baixo relevo.

**outset** – Efeito 3D no estilo alto relevo.

**none** – Sem bordas.

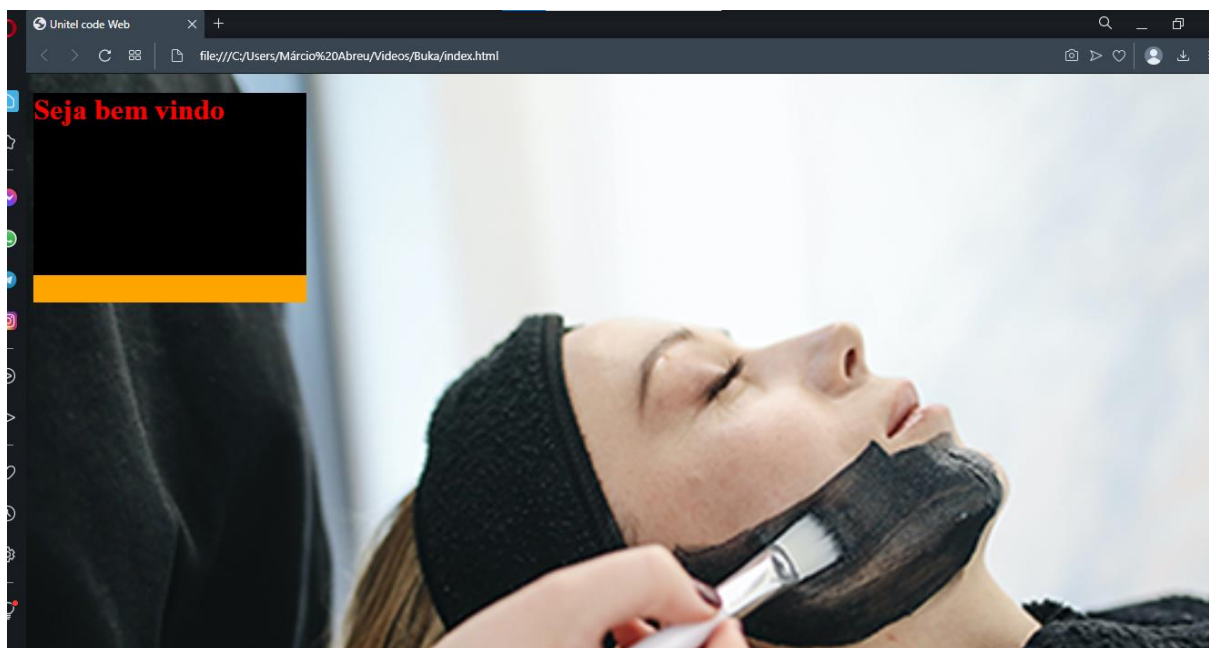
**hidden** – Bordas ocultas

Podemos trabalhar com as bordas de forma individual, ou seja cada borda com sua própria configuração, neste caso devemos usar uma propriedade para cada borda.

**border-top** = Borda superior.  
**border-right** = Borda direita.  
**border-bottom** = Borda inferior.  
**border-left**: esquerda.

```
index.html  estilo.css x
estilo.css > .quadrado
1  body{
2      background-image: url(../img/home.jpg);
3      background-size: cover;
4      background-repeat: no-repeat;
5  }
6
7  .quadrado {
8      color: red;
9      background-color: black;
10     width: 300px;
11     height: 200px;
12     border-bottom: solid 30px orange;
13 }
```

Agora vamos ver o resultado no navegador.



**border-radius / Cantos arredondada:** Uma novidade de CSS3 em relação aos cantos arredondados, onde podemos trabalhar de forma que cada canto tenha um arredondamento ou de forma que todos os os cantos tenham o mesmo valor.

O uso da propriedade é bem simples, basta informar o valor do arredondamento com a unidade de medida desejada, pixel, %, em, etc.

**margin – margens:** Os elementos HTML possuem margens externas que podem ser controladas para configurar em certo espaço entre os elementos.

Também podemos usar os parâmetros separadamente.

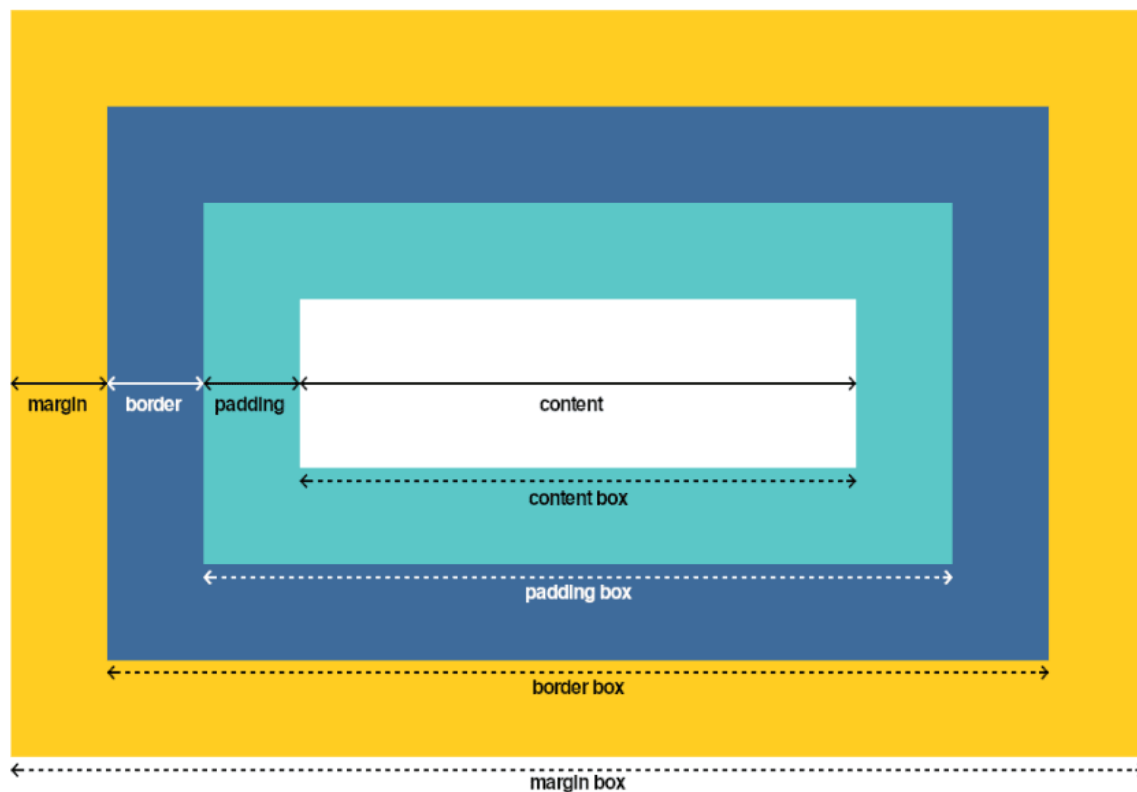
**margin-top**= Margem superior

**margin-right** = Margem direita

**margin-bottom** = Margem inferior

**margin-left** = Margem esquerda

**padding - margens internas :** Padding são as margens internas dos elementos, usamos o padding da mesma forma que usamos margin, porém, na parte interna dos elementos, veja a ilustração.



Note que a margem “cresce” para o lado de fora da borda e padding para o lado de dentro.

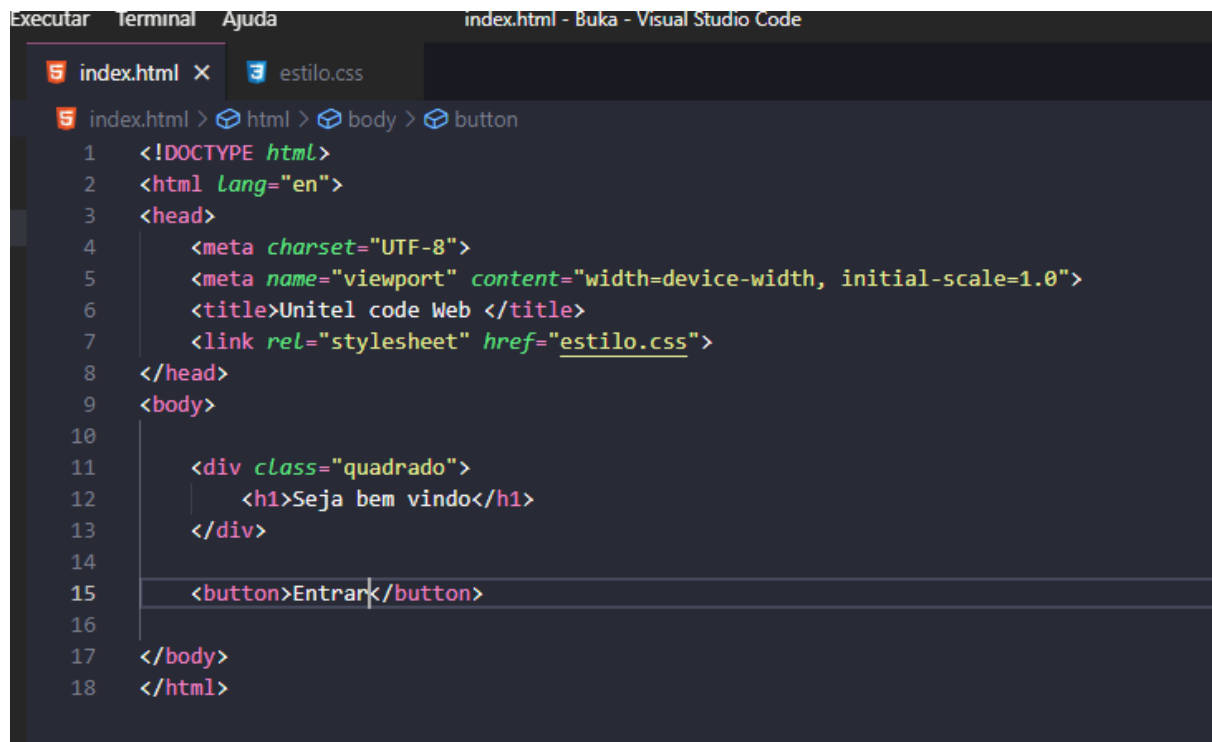
**padding-top** = Espaçamento interno superior

**padding-right**=Espaçamento interno direito

**padding-bottom**= Espaçamento interno inferior

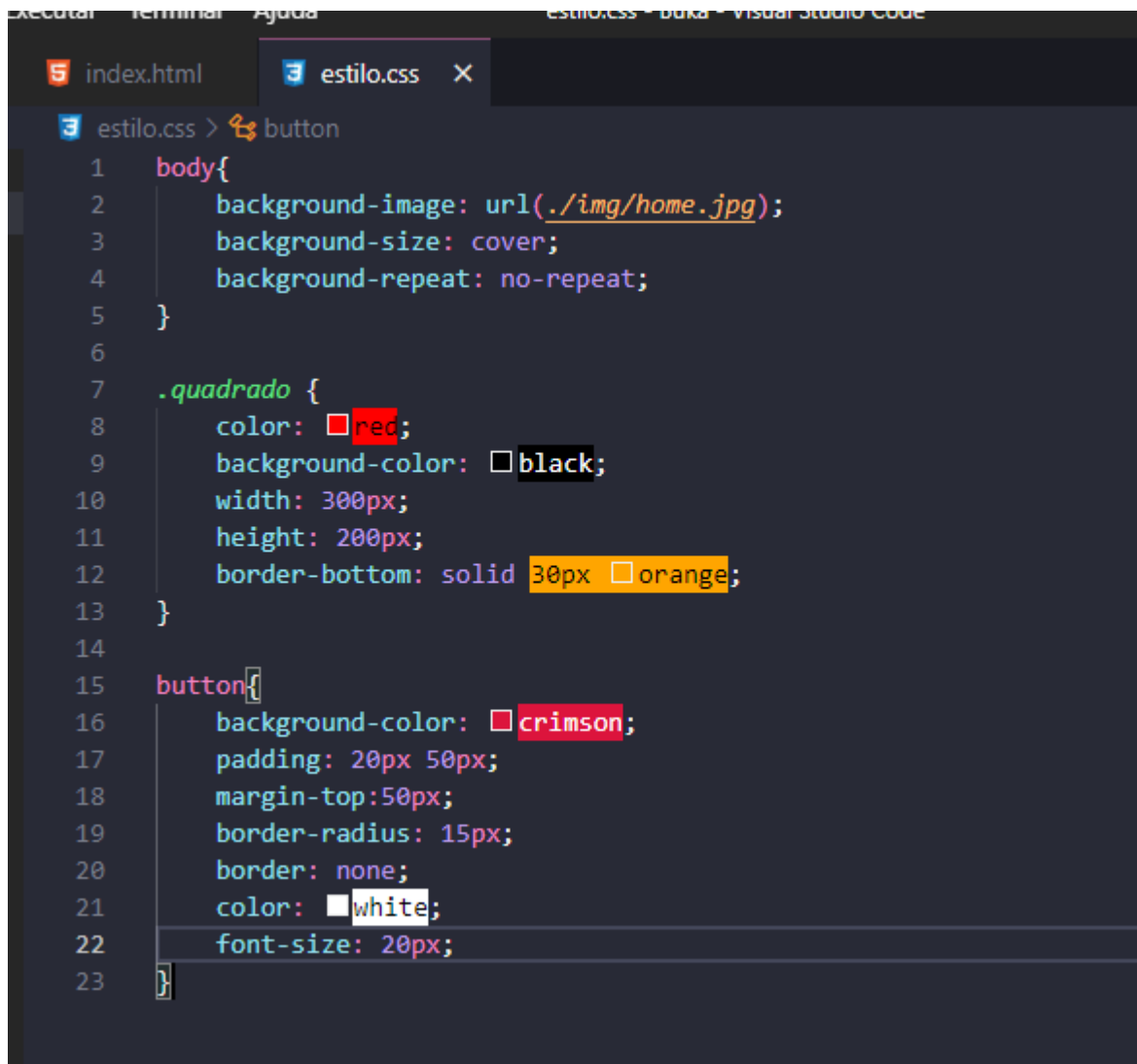
**padding-left** = Espaçamento interno esquerdo





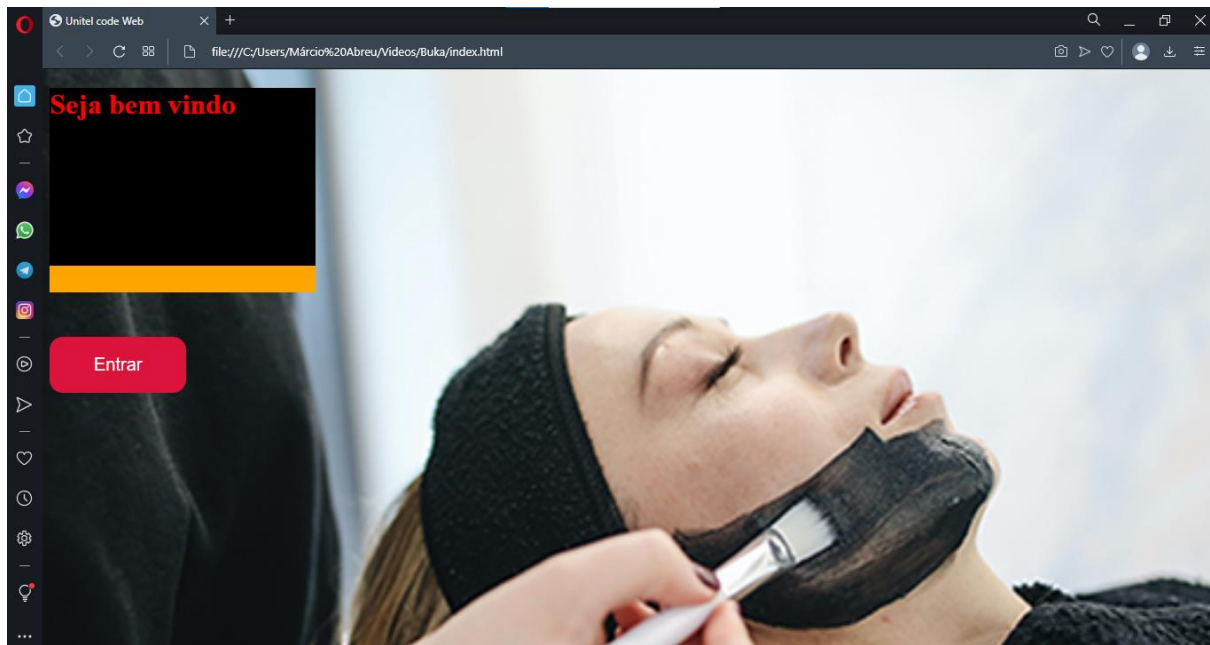
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Unitel code Web </title>
7   <link rel="stylesheet" href="estilo.css">
8 </head>
9 <body>
10
11   <div class="quadrado">
12     <h1>Seja bem vindo</h1>
13   </div>
14
15   <button>Entrar</button>
16
17 </body>
18 </html>
```

No html criamos um botão que tem como nome. Agora vamos mexer no css.

The image shows a screenshot of the Visual Studio Code editor with two tabs open: 'index.html' and 'estilo.css'. The 'estilo.css' tab is active, showing CSS code. The code is as follows:

```
1 body{
2     background-image: url(../img/home.jpg);
3     background-size: cover;
4     background-repeat: no-repeat;
5 }
6
7 .quadrado {
8     color: red;
9     background-color: black;
10    width: 300px;
11    height: 200px;
12    border-bottom: solid 30px orange;
13 }
14
15 button{
16     background-color: crimson;
17     padding: 20px 50px;
18     margin-top: 50px;
19     border-radius: 15px;
20     border: none;
21     color: white;
22     font-size: 20px;
23 }
```

No botão que acabamos de criar, demos um fundo de cor rosa, padding inteiro que vai receber dois valores de medida. o primeiro valor : 20px vai ser tanto para **top**(superior) e **bottom**(inferior) e 50px vai ser tanto para **left** (esquerda) e **right**(direita). Agora no caso da margin, estamos a especificar que queremos mexer na **margin-top**(margin-superior) que vai receber um tamanho de 50px. Agora o border-radius é para tornar o nosso botão em formato de círculo, e está a receber um tamanho de 15px(left,top,right, bottom). aplicarmos uma cor branca na letra do texto e um tamanho de 20px. Agora vamos ver o resultado no navegador.



Esse é o resultado que nós fizemos.

**box-shadow:** A propriedade box-shadow tem a mesma sintaxe de text-shadow, porém aplica sombras aos outros elementos diferentes de texto, como <div> por exemplo.

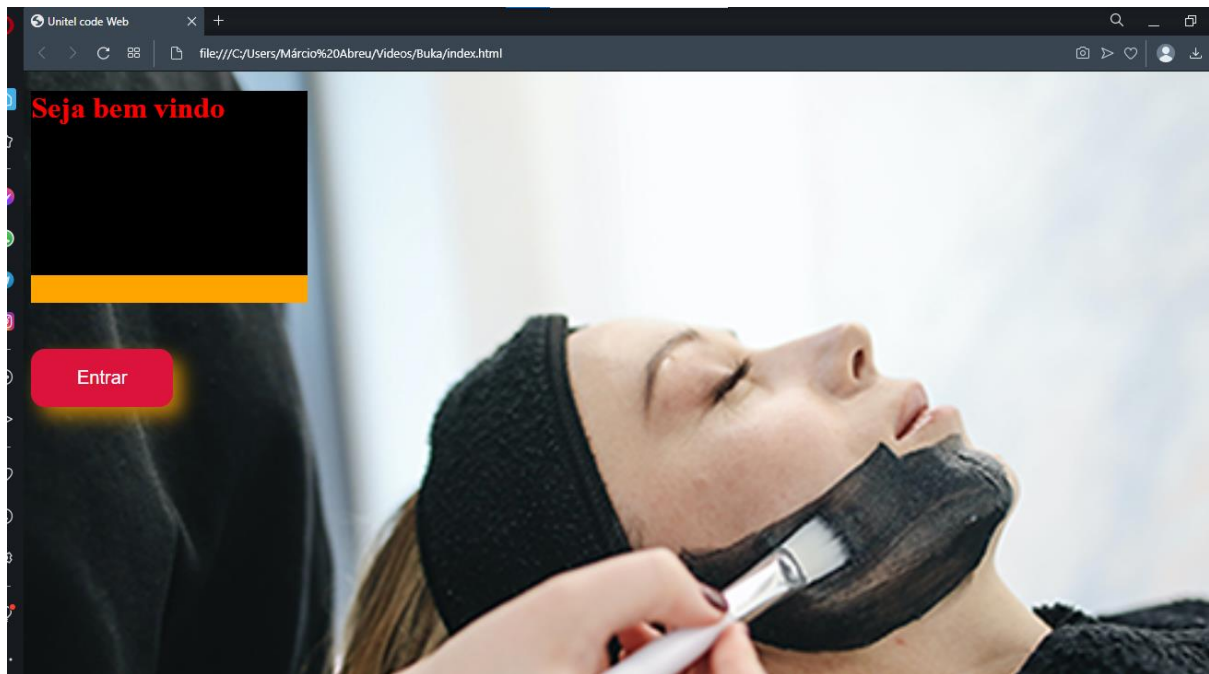
```
tilo.css > ...
body{
  background-image: url(./img/home.jpg);
  background-size: cover;
  background-repeat: no-repeat;
}

.quadrado {
  color: red;
  background-color: black;
  width: 300px;
  height: 200px;
  border-bottom: solid 30px orange;
}

button{
  background-color: crimson;
  padding: 20px 50px;
  margin-top: 50px;
  border-radius: 15px;
  border: none;
  color: white;
  font-size: 20px;
  box-shadow: 10px 10px 20px rgb(192, 127, 7);
}

|
```

Agora aplicamos uma sombra no botão. Agora vamos visualizar no navegador como ficou.

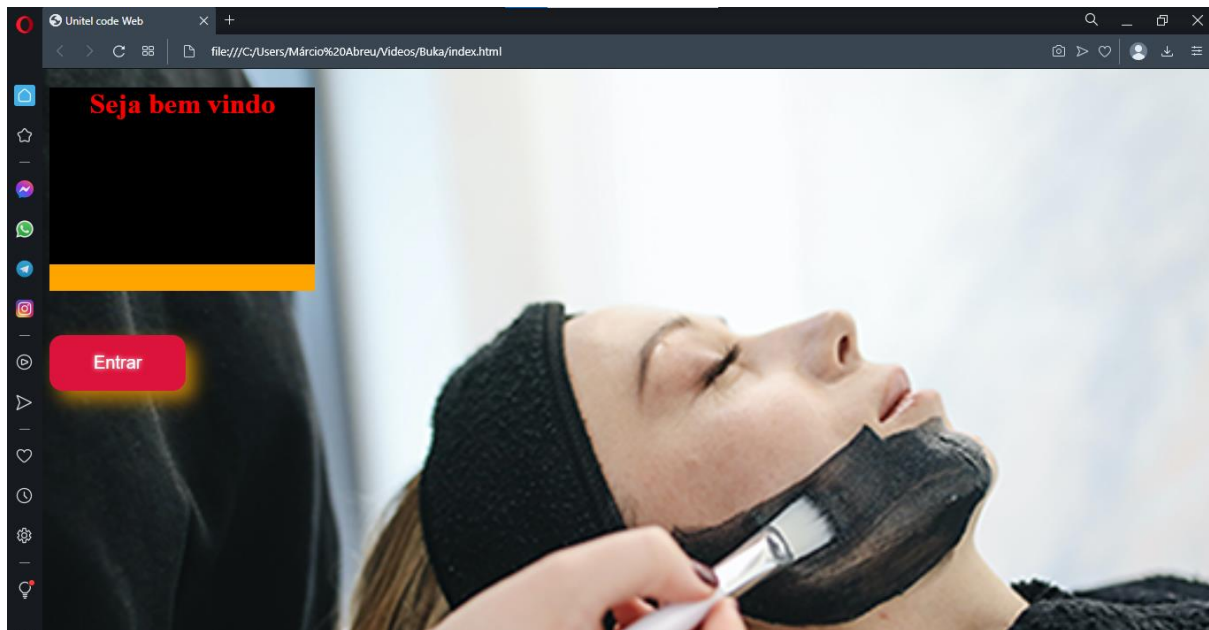


Agora estamos ver o efeito da sombra que aplicamos. isso não quer dizer que o único elemento ou tag que merece sombra é o button. os texto também podem receber sombra, basta aplicar **text-shadow** e usar as mesma propriedade do **box-shadow**.

**text-align - alinhamento** : Esta propriedade define o alinhamento do texto.

```
estilo.css > .quadrado
1  body{
2      background-image: url(../img/home.jpg);
3      background-size: cover;
4      background-repeat: no-repeat;
5  }
6
7  .quadrado {
8      color: red;
9      background-color: black;
10     width: 300px;
11     height: 200px;
12     border-bottom: solid 30px orange;
13     text-align: center;
14 }
```

Quando aplicarmos o `text-align:center;` o nosso texto vai ficar no centro. Vamos ver o resultado no navegador.



Nesse momento estamos a ver o título seja bem vindo centralizado no meu quadrado.

Os valores possíveis para text-align são:

**left** = Esquerda

**right** = Direita

**center** = Centro

**justify** = Justificado

**text-decoration – Decoração** : Configura o método de decoração do texto, a decoração são tipos de linhas como abaixo do texto, acima do texto ou riscado.

```
Ver  Acessar  Executar  Terminal  Ajuda  index.html - Buka - Visual Studio Code

...  index.html x  estilo.css

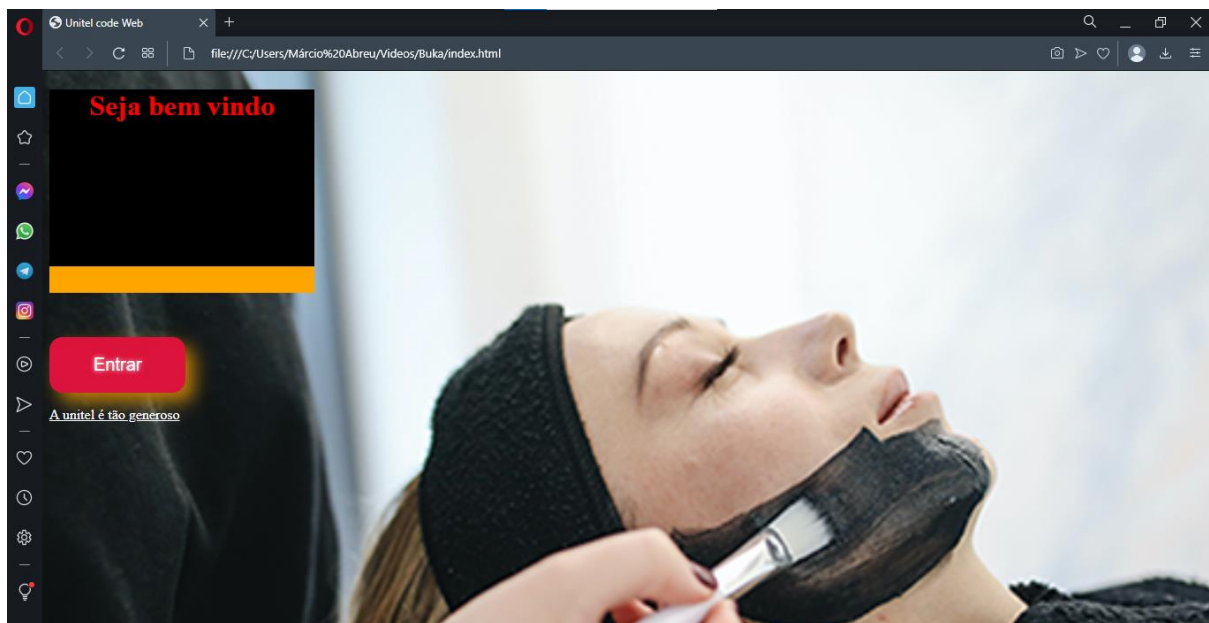
index.html > html > body > p
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Unitel code Web </title>
7      <link rel="stylesheet" href="estilo.css">
8  </head>
9  <body>
10
11      <div class="quadrado">
12          <h1>Seja bem vindo</h1>
13      </div>
14
15      <button>Entrar</button>
16
17      <p> A unitel é tão generoso</p>
18
19  </body>
20  </html>
```

Criamos um parágrafo no html. agora vamos ver o nosso css.

```
index.html  estilo.css x

estilo.css > p
20  border-radius: 15px;
21  border: none;
22  color: white;
23  font-size: 20px;
24  box-shadow: 10px 10px 20px rgb(192, 127, 7);
25  text-shadow: 0px 0px 5px #FFF;
26  }
27
28  p{
29      text-decoration: underline; /* sublinha o texto */
30      color: white;
31  }
32
33
```

Nosso parágrafo recebeu um texto decorado sublinhado. vamos visualizar no navegador.



Nesse momento estamos a ver o texto sublinhado.

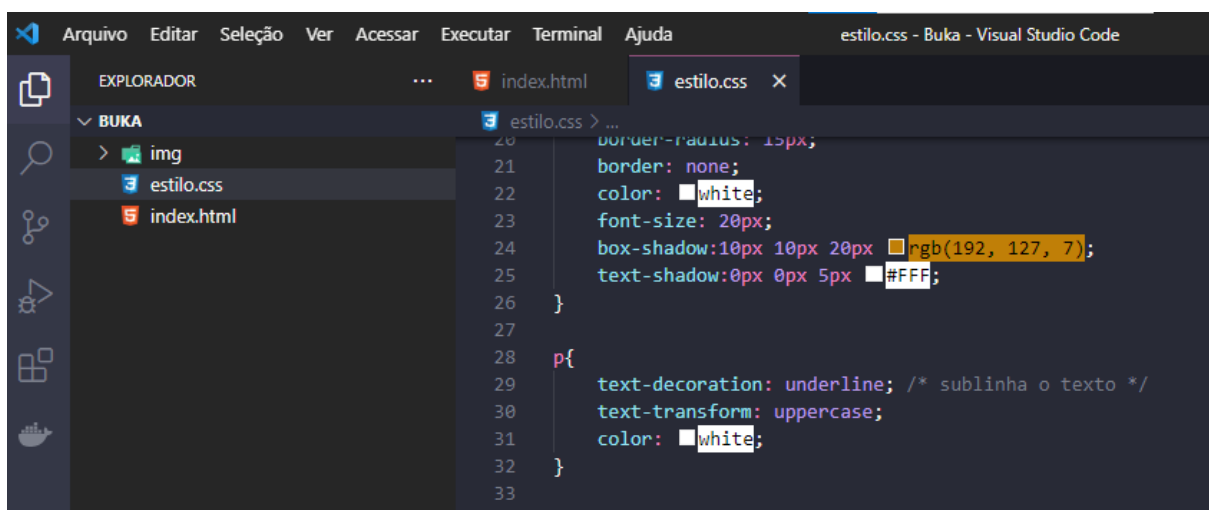
**text-decoration:underline;** - sublinhado o texto em baixo

**text-decoration:overline;** - sublinhado o texto em cima

**text-decoration:line-through;** - sublinhado o texto no meio

**text-decoration:none** - tirar o texto caso esteja sublinhado.

**text-transform – Transformações** : Com esta propriedade podemos configurar letras em maiúsculas, minúsculas e primeiras letras em maiúsculo.



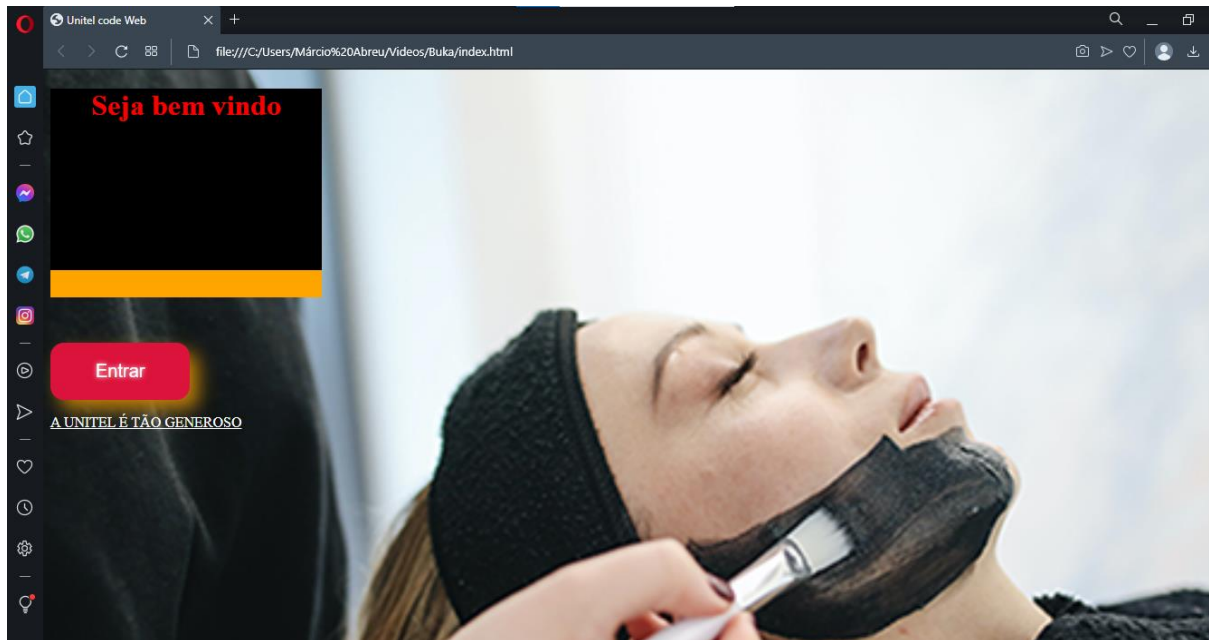
**text-transform:uppercase;** Converte o texto em maiúsculo



**text-transform:lowercase;** Converte o texto em minúsculo

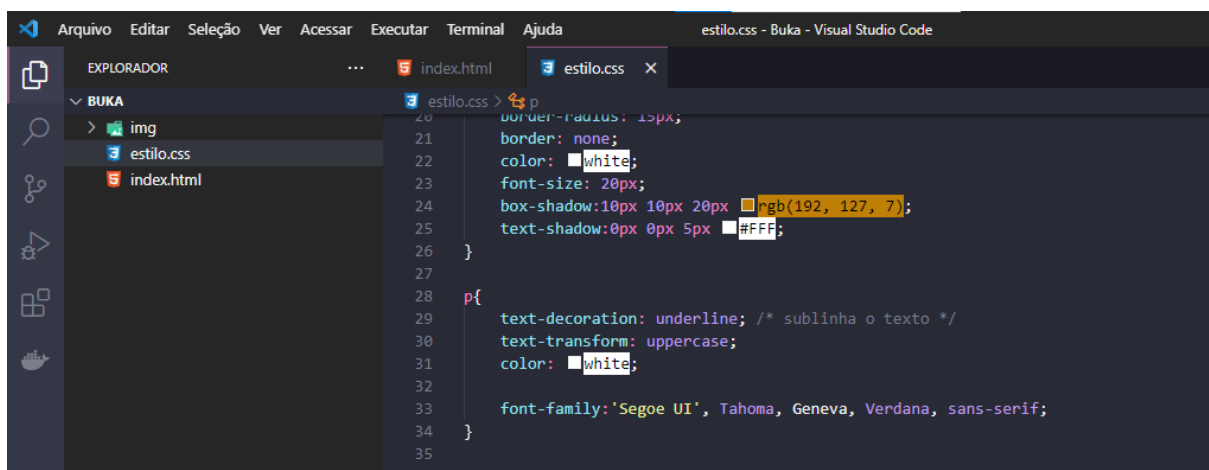
**text-transform:capitalize;** Converte as primeiras letras do texto em maiúsculo.

Agora vamos ver como ficou no navegador.

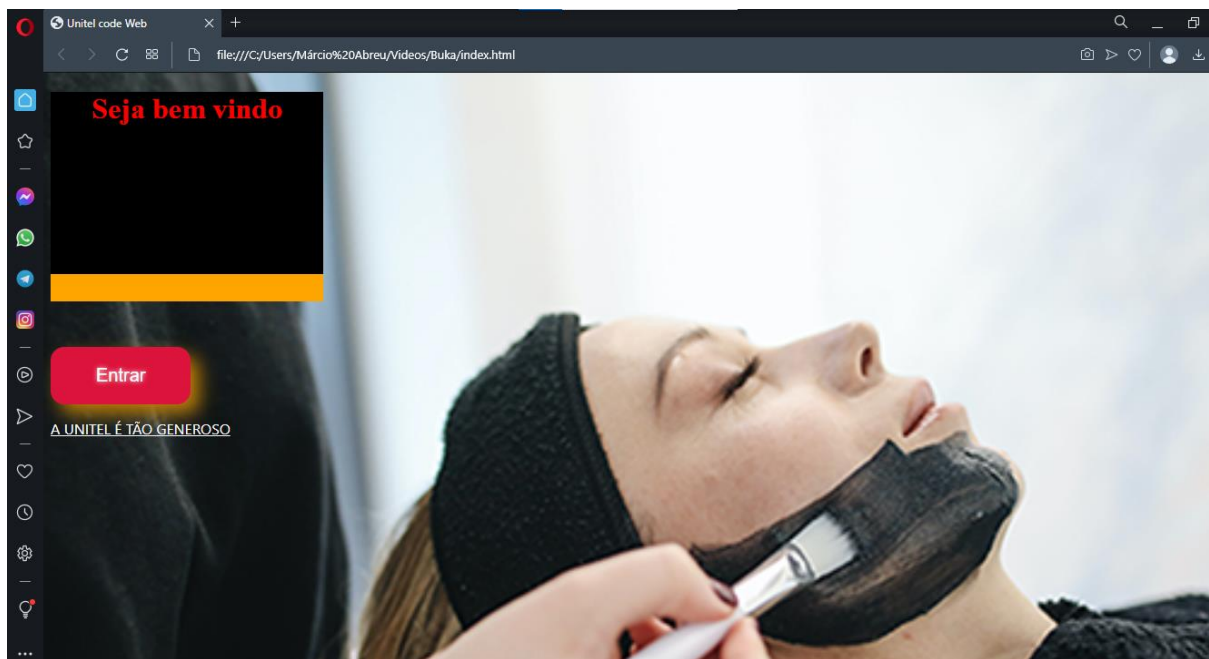


De acordo com a página, estamos a ver o texto todo em maiúsculo. porque aplicarmos **text-transform:uppercase;**

**font-family:** Com parâmetro font-family podemos configurar grupos de fontes para serem usadas em um determinado texto, a vantagem de especificar grupos de fontes é tentar garantir que sejam usadas somente fontes definidas pelo programador, o que não é 100% garantido.



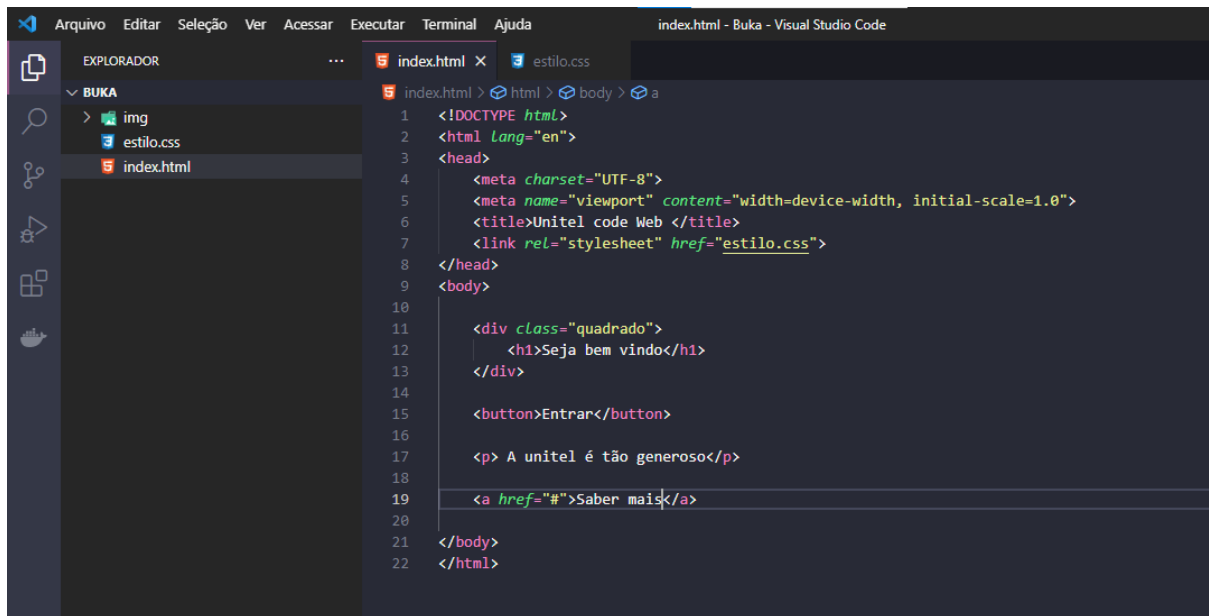
Aplicamos um tipo de tipo de letra diferente do que estamos a usar na nossa página. É de realçar que os navegadores não entendem todos os tipos de letras.



Nesse momento estamos a ver o resultado do tipo de letra que há entre o texto do parágrafo com os demais.

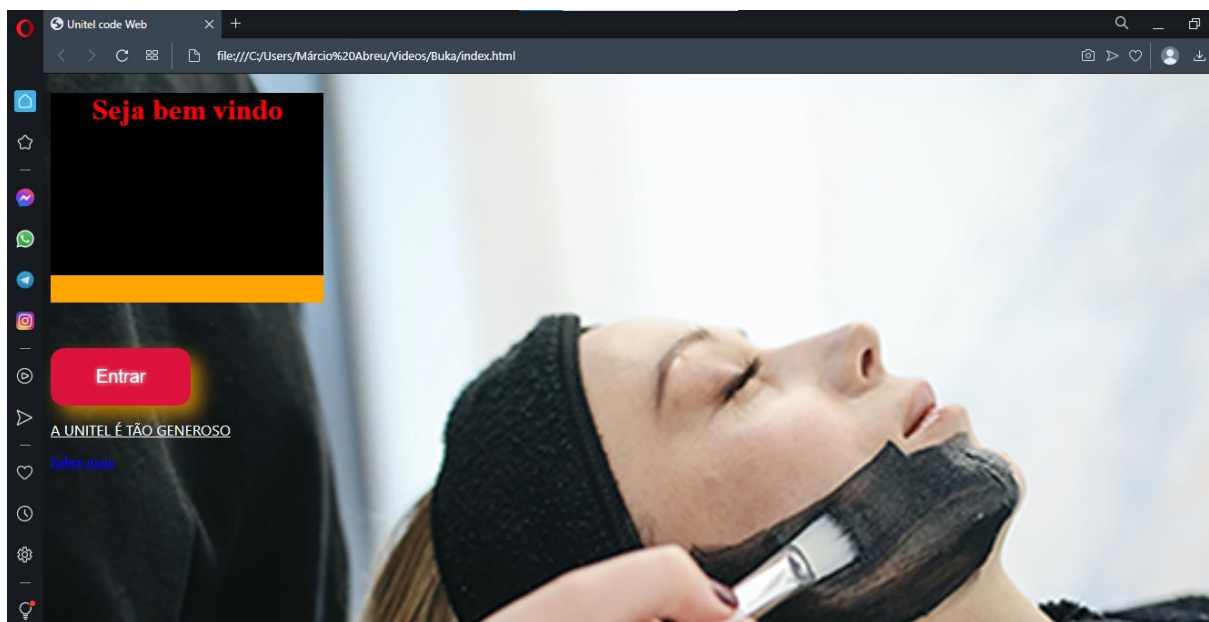
## **Formatando o link como um botão**

Podemos usar as propriedades CSS para transformar o link na tag normal como se fosse um botão.

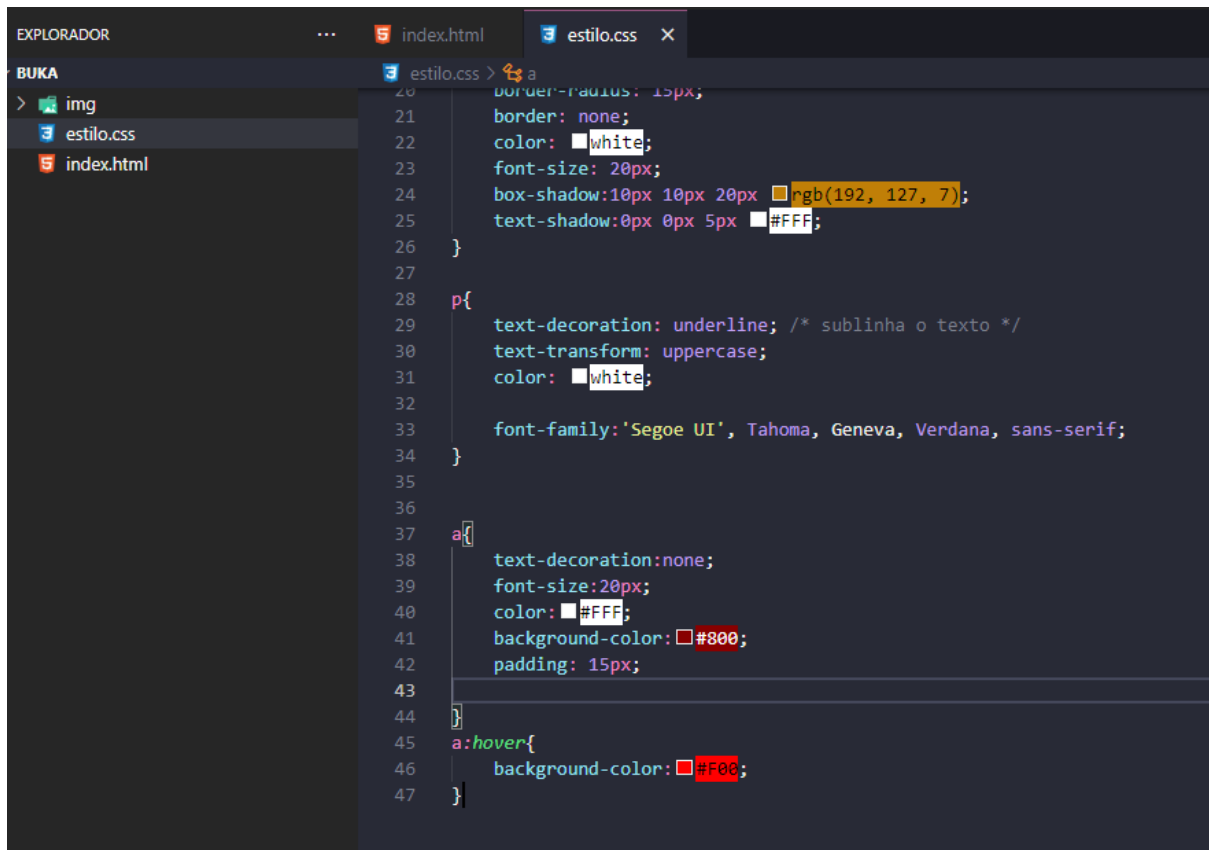


```
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Unitel code Web </title>
7   <link rel="stylesheet" href="estilo.css">
8 </head>
9 <body>
10
11   <div class="quadrado">
12     <h1>Seja bem vindo</h1>
13   </div>
14
15   <button>Entrar</button>
16
17   <p> A unitel é tão generoso</p>
18
19   <a href="#">Saber mais</a>
20
21 </body>
22 </html>
```

Assim que aplicamos o link <a> a nossa página vai ficar assim.



O saber mais está com cor azul e texto sublinhado como padrão. Para nós podermos tirar isso, devemos usar o css.

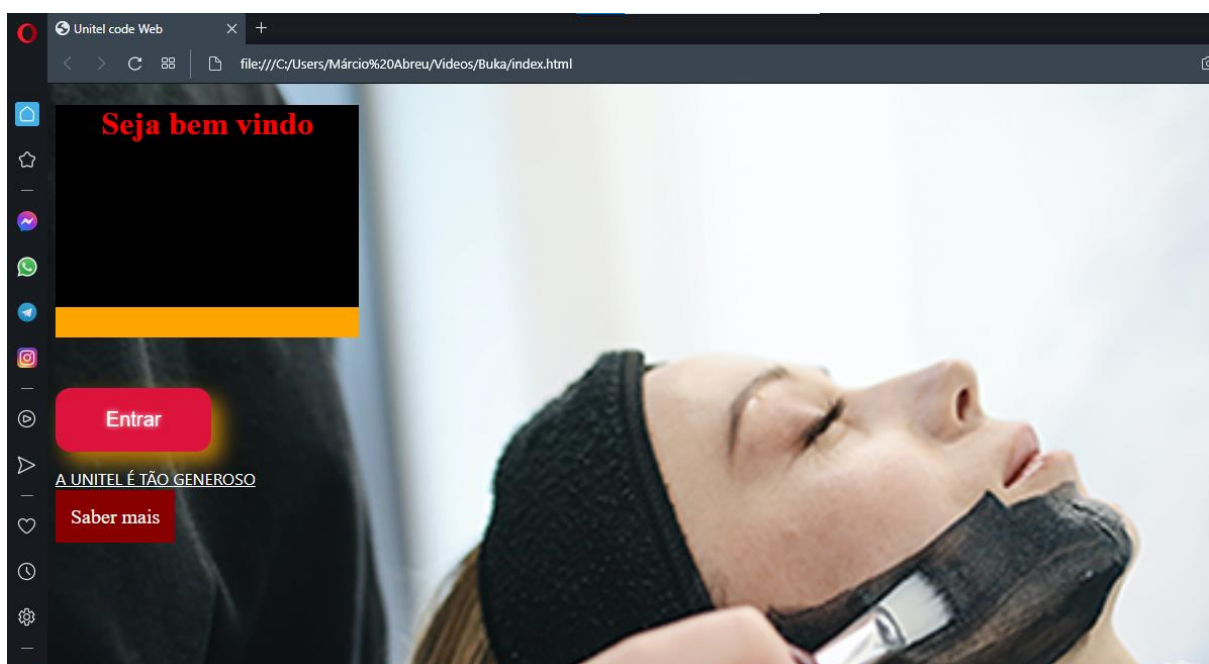


```
EXPLORADOR
...
index.html
estilo.css

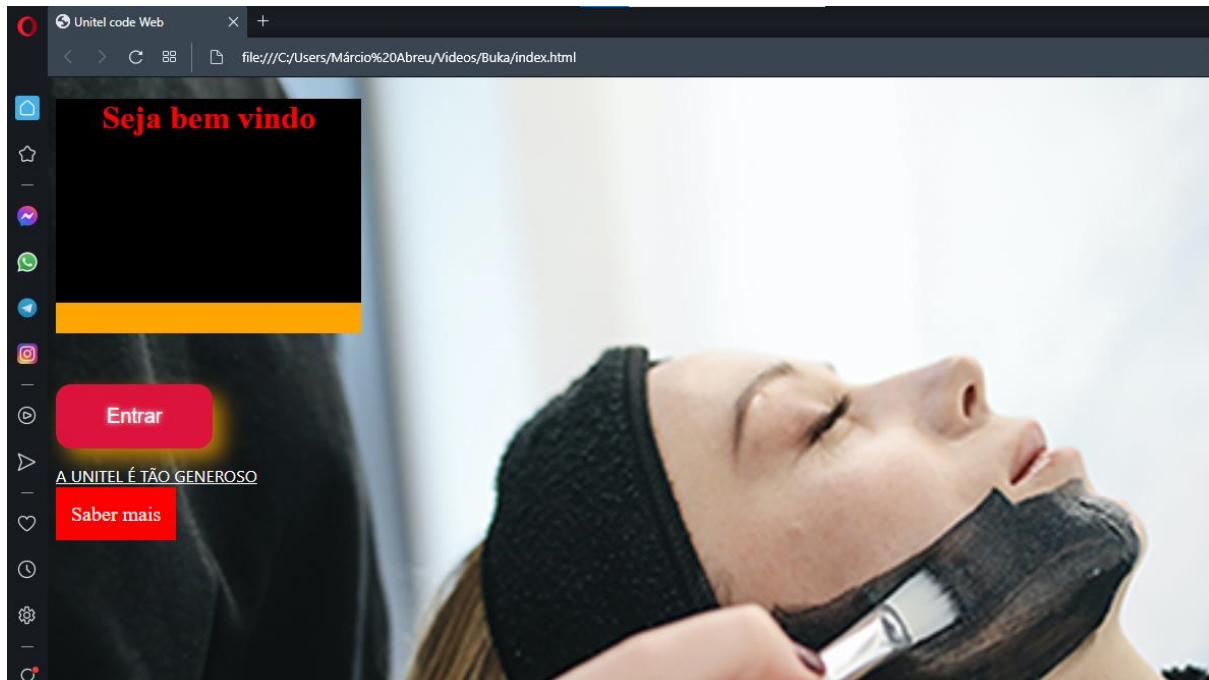
BUKA
> img
estilo.css
index.html

estilo.css
20 border-radius: 15px;
21 border: none;
22 color: white;
23 font-size: 20px;
24 box-shadow: 10px 10px 20px rgb(192, 127, 7);
25 text-shadow: 0px 0px 5px #FFF;
26 }
27
28 p{
29 text-decoration: underline; /* sublinha o texto */
30 text-transform: uppercase;
31 color: white;
32
33 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
34 }
35
36
37 a{
38 text-decoration: none;
39 font-size: 20px;
40 color: #FFF;
41 background-color: #800;
42 padding: 15px;
43 }
44
45 a:hover{
46 background-color: #F00;
47 }
```

De acordo com o código, a única coisa nova é o hover. O **:hover** é uma pseudo-classe **CSS** que ao ser utilizada ativa estilos em um elemento. O **:hover** possibilita um elemento ter o seu estilo alterado através da ação do usuário, quando o próprio passa o cursor do mouse sobre o elemento.

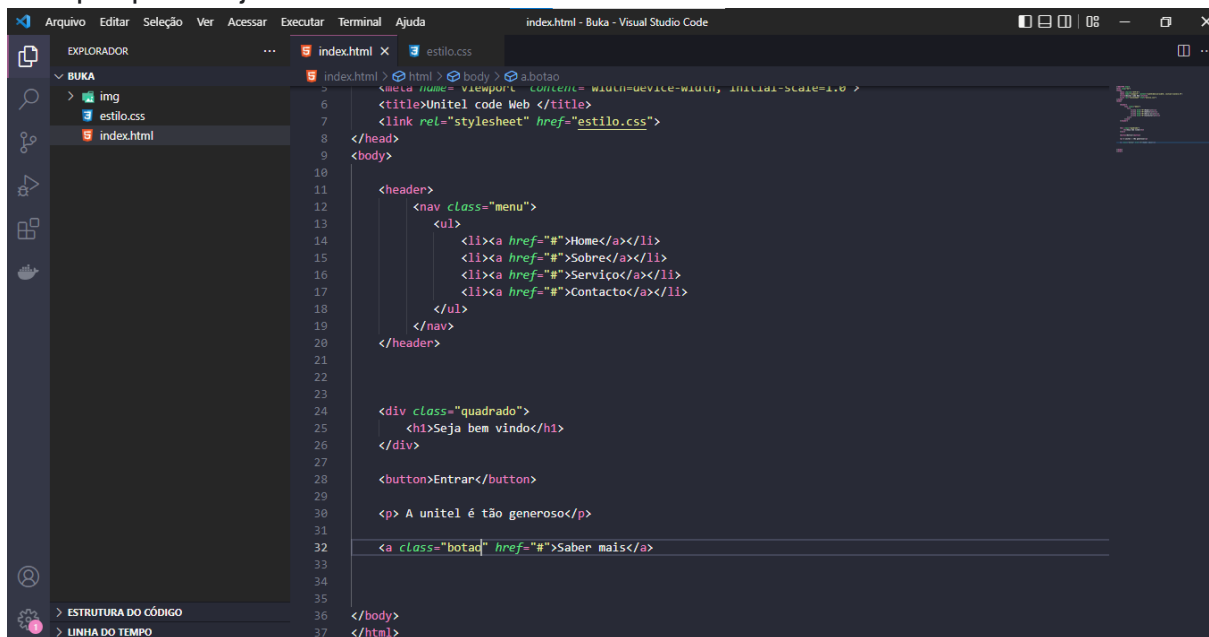


Até ao momento ainda não aproximamos o cursor até ao botão. Assim que nós encontrar o cursor no botão, a cor do botão vai trocar.

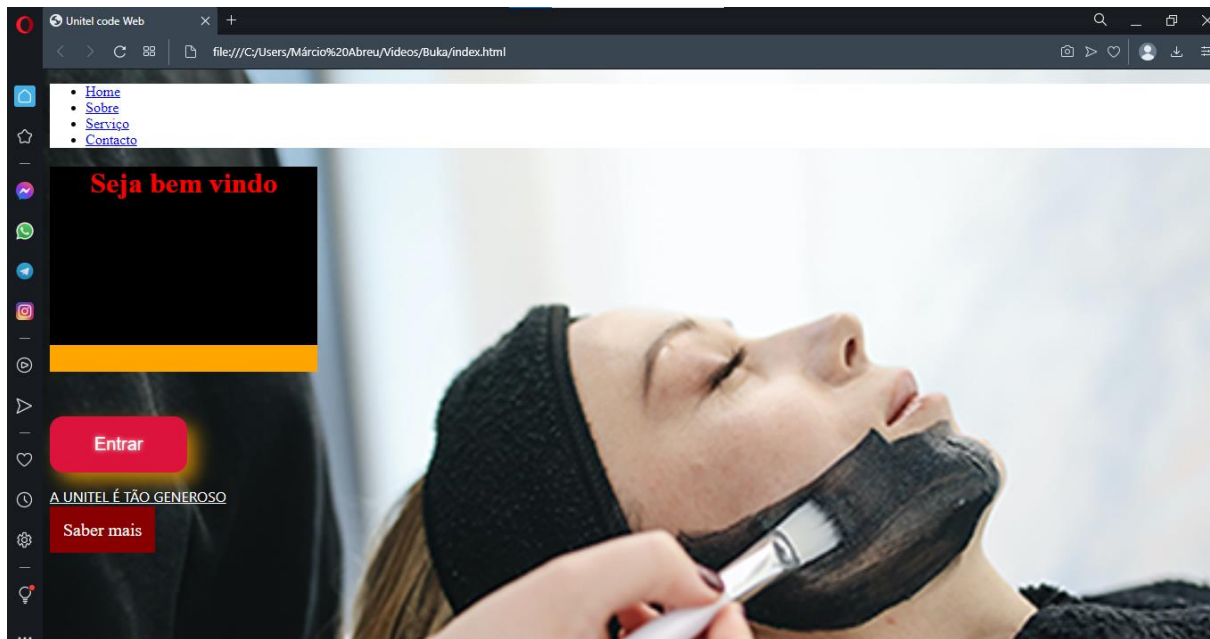


Nesse momento estamos a ver que a cor mudou. Porque aproximamos o cursor no botão.

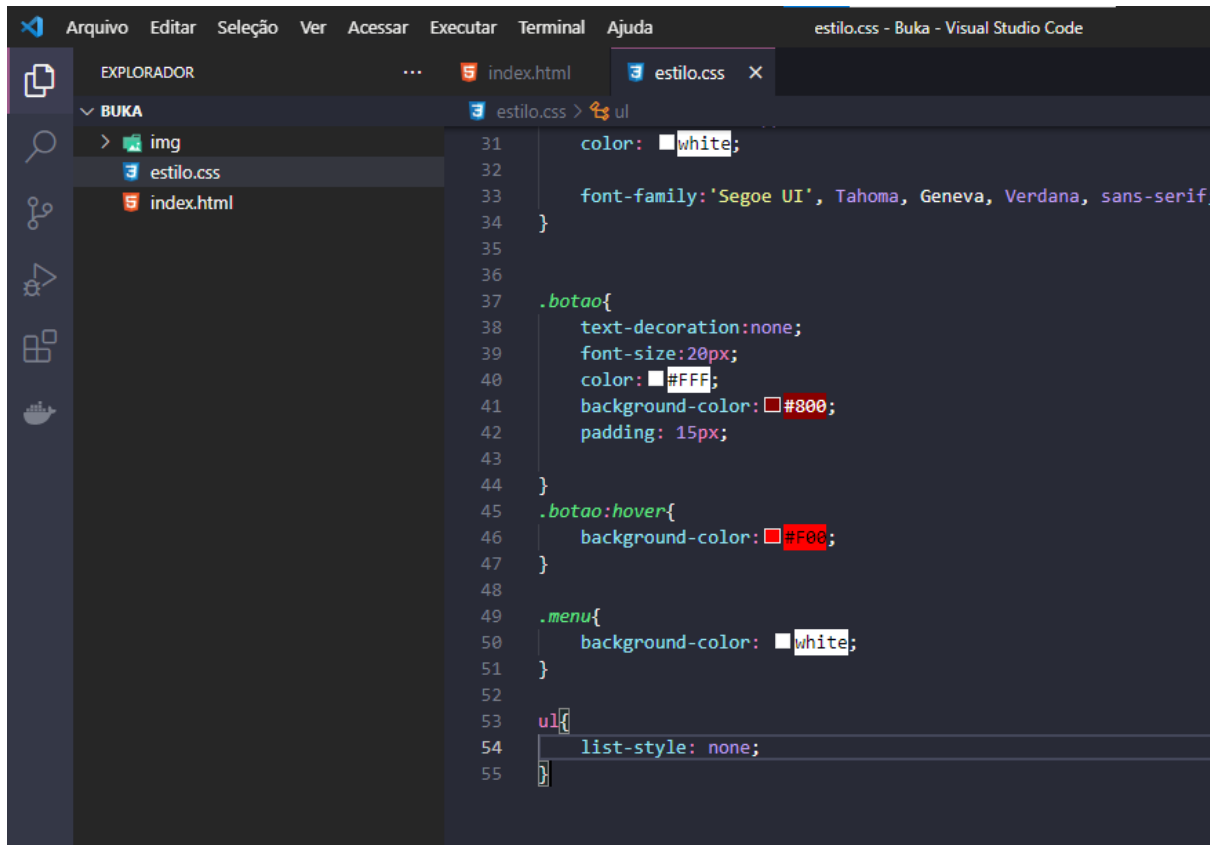
**list-style:** A meta propriedade list-style nos permite colocar e tirar o estilo da lista. Mas por padrão já vem com estilo de lista definido.



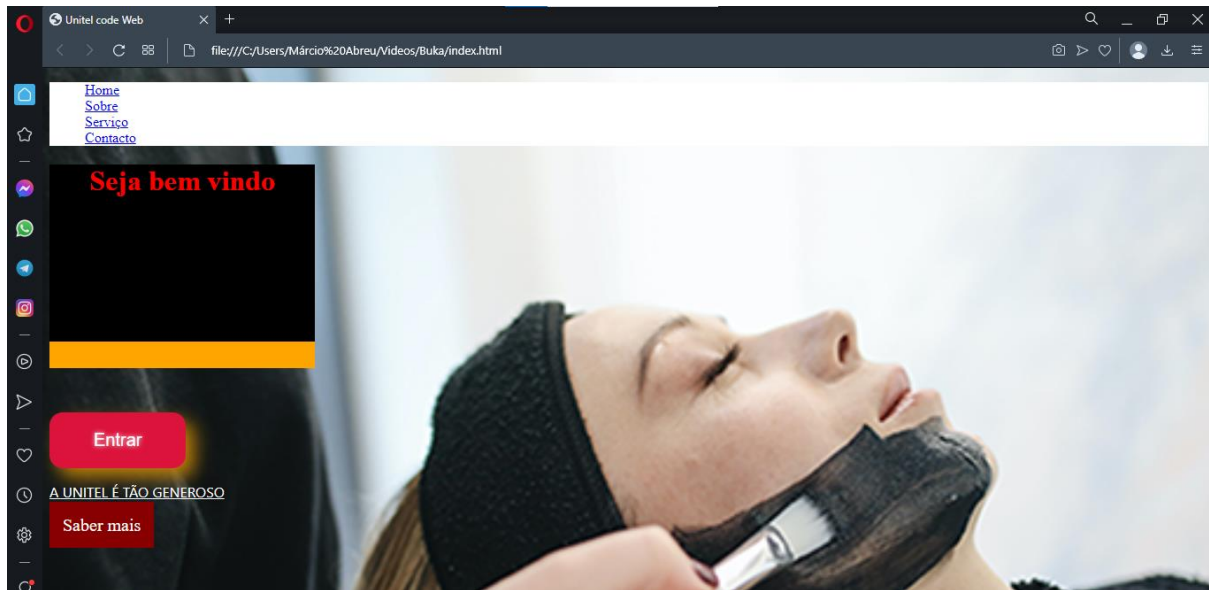
Nesse momento já criamos a nossa lista, que é nosso menu. agora vamos ver como está no navegador.



Estamos a ver que a nossa lista de estilo vem como pontos por padrão. para tirar isso, devemos mexer no css. nesse caso devemos mexer na tag **ul** para tirar esses pontos.

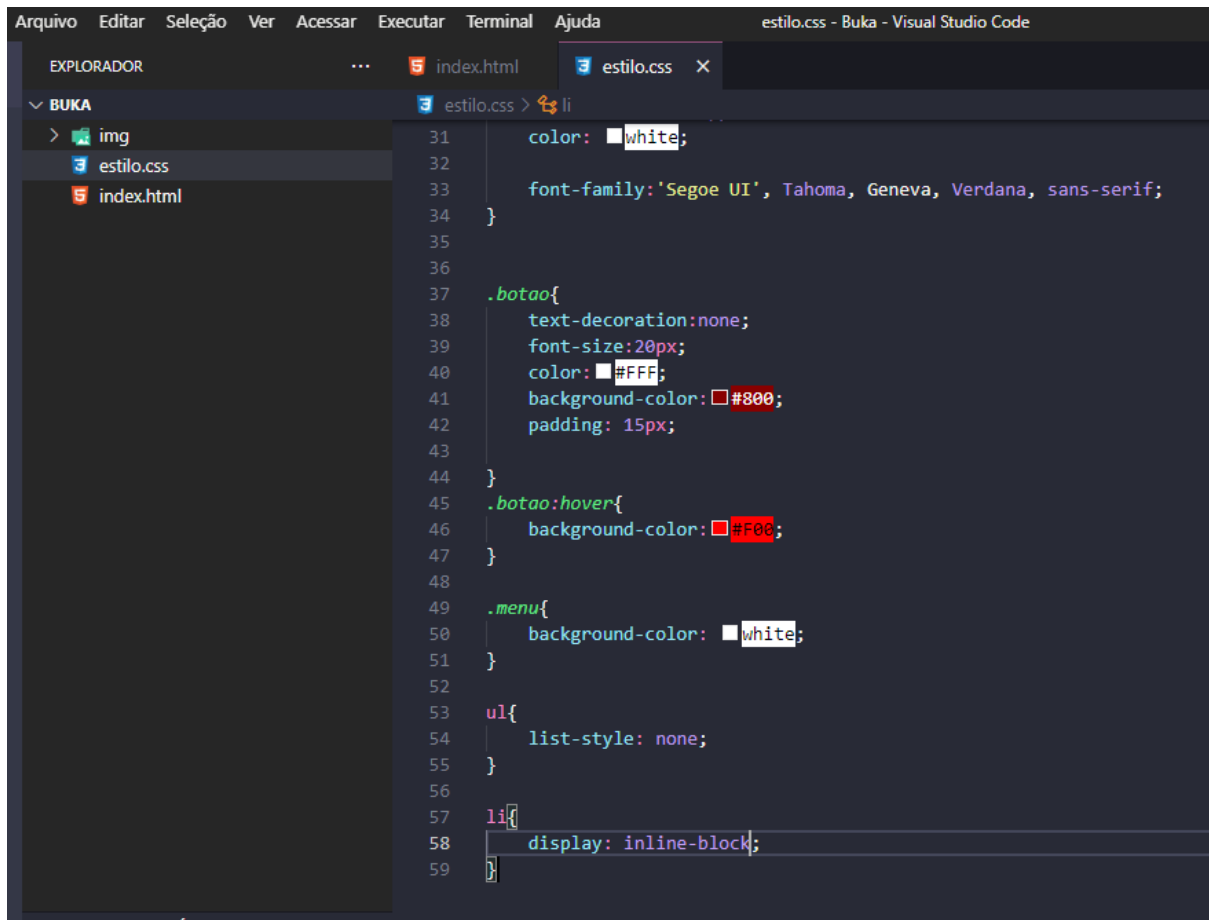


Não aplicamos nenhuma lista de estilo. Por isso tem um **list-style:none**. Agora vamos ver como está no navegador.



Agora, caso queira meter todos em formato de linha, devemos aplicar a propriedade **display: inline-block;** na tag **<li>**.

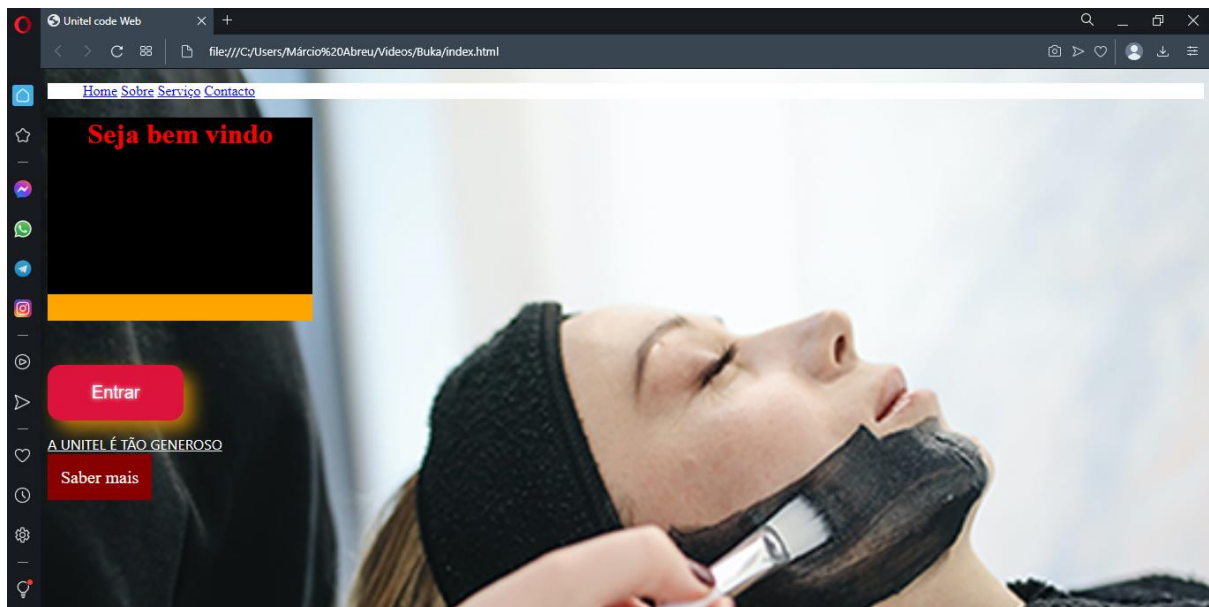




The image shows a screenshot of the Visual Studio Code editor. The Explorer panel on the left shows a project named 'BUKA' with files 'img', 'estilo.css', and 'index.html'. The main editor area shows the 'estilo.css' file with the following CSS code:

```
31 color: white;
32
33 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
34 }
35
36
37 .botao{
38   text-decoration: none;
39   font-size: 20px;
40   color: #FFF;
41   background-color: #800;
42   padding: 15px;
43 }
44
45 .botao:hover{
46   background-color: #F00;
47 }
48
49 .menu{
50   background-color: white;
51 }
52
53 ul{
54   list-style: none;
55 }
56
57 li{
58   display: inline-block;
59 }
```

Vamos ver o resultado no navegador.





## Exercício para os formandos

1- tirar o texto sublinhado do menu e trocar a sua cor. Devem cor vermelha. E devem trocar o fundo do menu e usar um fundo com cor preta.

### @media – layouts responsivos

As regras media permitem configurar diferentes regras CSS para diferentes tipos de mídia, diante da necessidade de facilitar a configuração do site para se adaptar bem em dispositivos de diferentes tamanhos, daí vem o conceito de design responsivo ou layout responsivo.

Os valores que podemos usar para os tipos de mídia são:

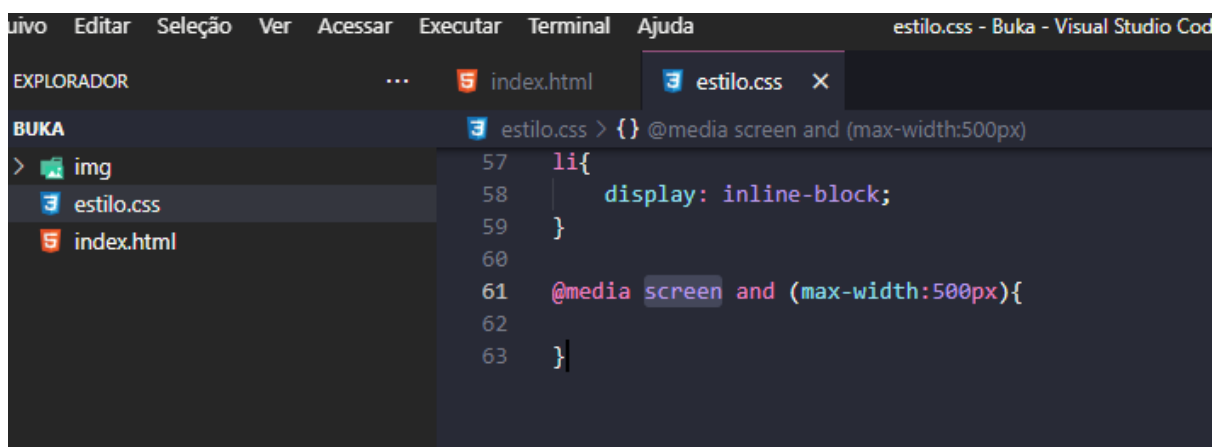
**all** = Todos os tipos de dispositivos

**print** = Usado para impressoras

**screen** = usado para telas, computador, celulares, tablets, etc.

**speech** = Usado para telas de leitores de conteúdo, leitores de ebook por exemplo, em voz alta.

Estrutura básica do media query.



Quando falarmos de flexbox, vamos usar para tornar a página responsiva.

Provérbios do formador **Márcio D'Abreu** : *"Queres ser bom programador? não copie código. prática sempre, até o teu cérebro memorizar. Nunca usa um código que você não conhece."*

.

**E esse provérbio**