



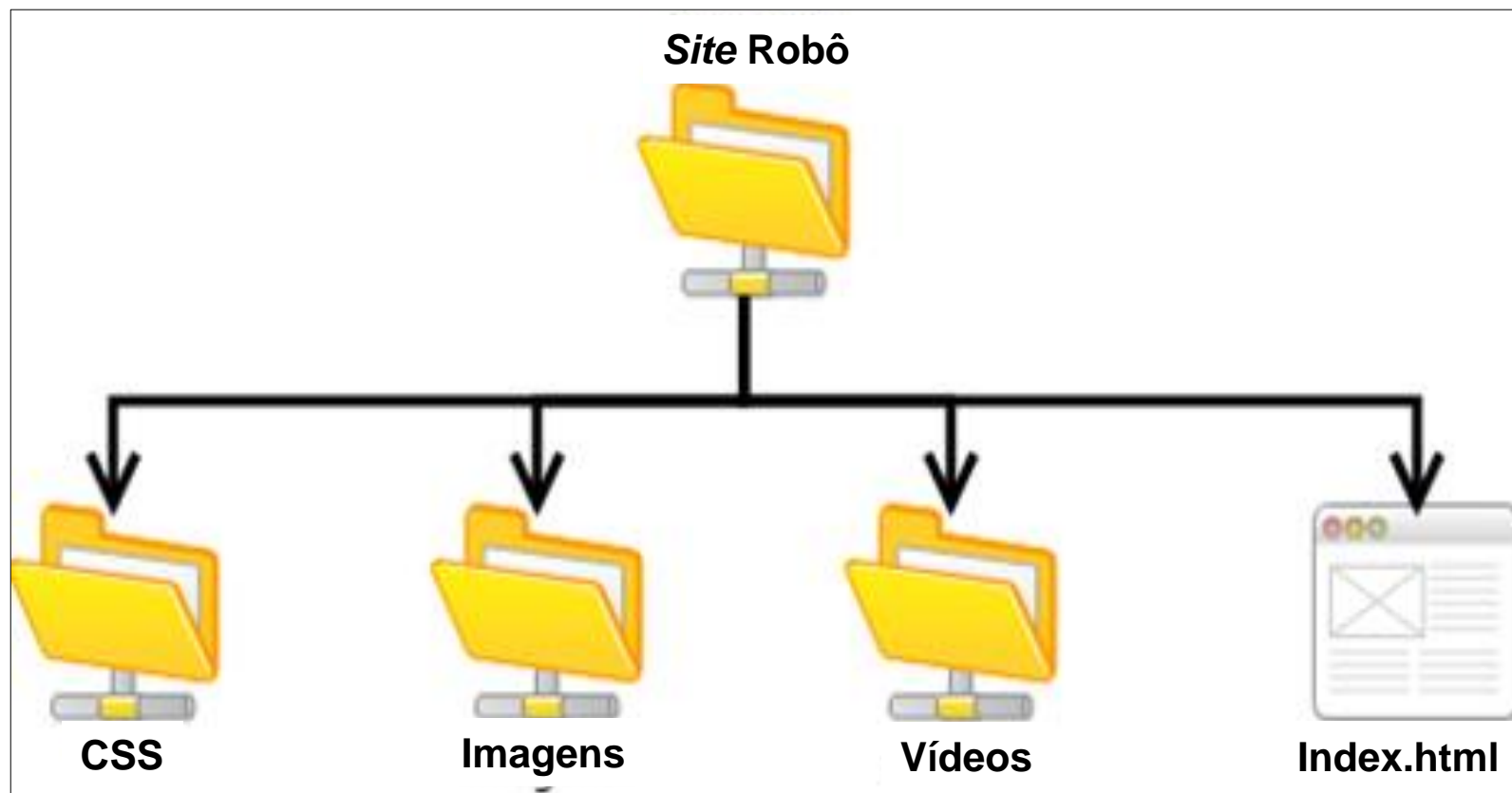
UNIDADE II

Programação
Web Responsiva

Prof. Me. Luiz Lozano

Criando e organizando o projeto

Chegou o momento de nós sermos apresentados formalmente para alguém que já ouvimos falar anteriormente, mas que não havíamos encontrado pessoalmente. Para que ocorra tudo certo no nosso primeiro encontro, vamos preparar o ambiente criando um novo projeto e organizando as nossas pastas desta forma:



Arquivo index.html

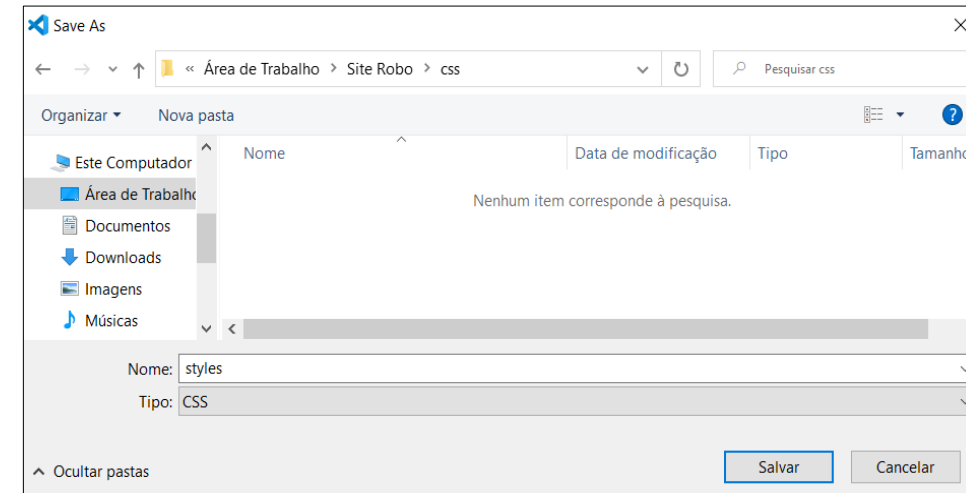
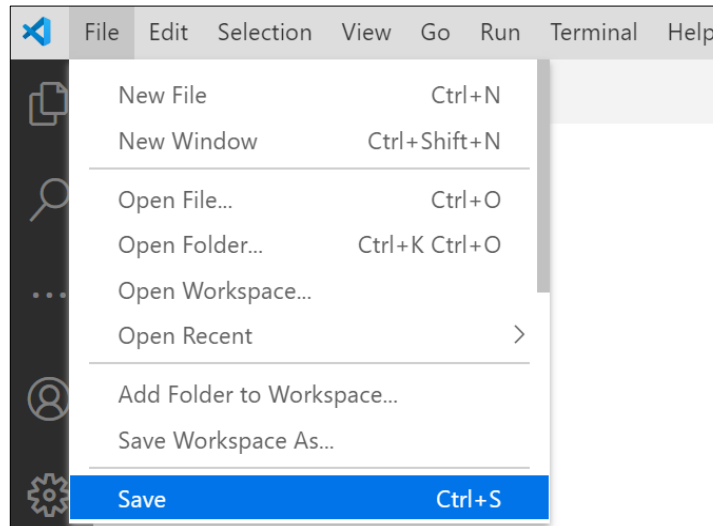
Dentro do arquivo index.html iremos inserir um código com a estrutura básica de uma página HTML:

```
<> index.html X
c: > Users > prof- > Desktop > Site Robo > <> index.html >
1  <!DOCTYPE html>
2  <html lang="pt-br">
3      <head>
4          <title>Site do Robô</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8
9      </body>
10 </html>
```

Criando um arquivo CSS

- Quando o objetivo é garantir uma formatação uniforme em todas as páginas de um *site*, as folhas de estilo em cascata (*Cascading Style Sheets*) fazem com que o trabalho seja simples. Essa tal folha de estilo é um mecanismo que adiciona estilos a elementos do HTML, como, por exemplo: cor de fundo, espaçamentos, fontes etc. Sendo assim, o CSS nos dá a possibilidade de ir além das barreiras de estilo impostas pelo HTML puro.

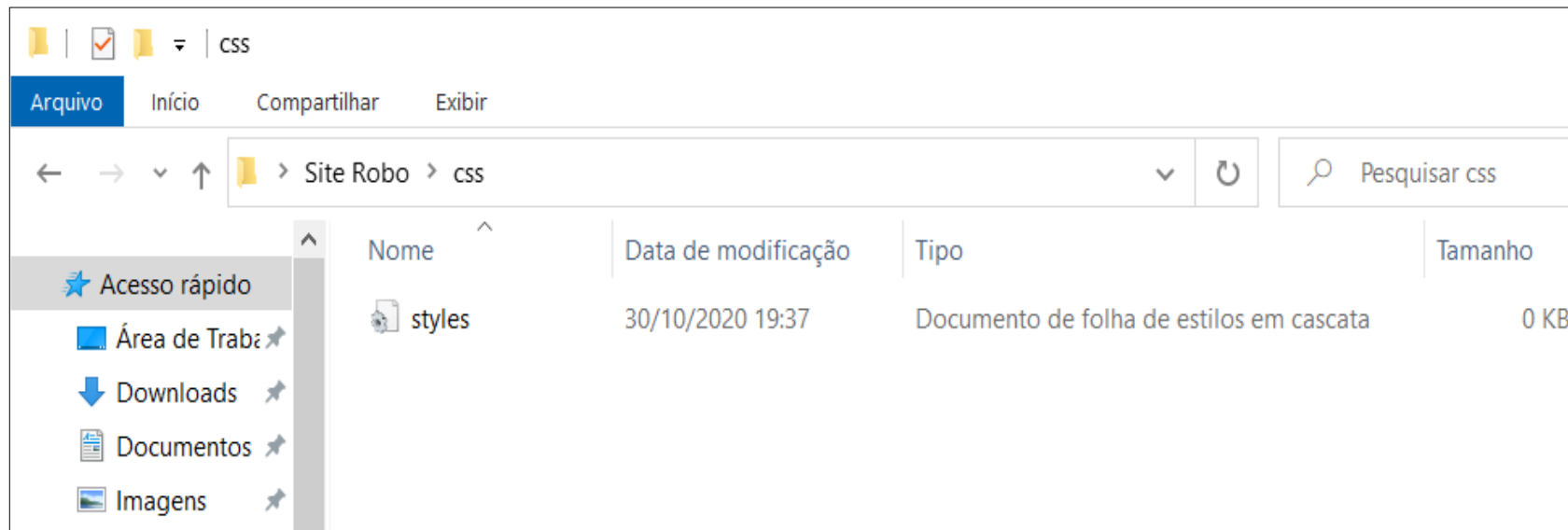
Vamos criar um arquivo CSS para adicionar estilos nas páginas HTML do nosso *site*, sendo assim, dentro de seu editor de texto crie um arquivo CSS através do *menu File*, opção *New File*:



Fonte: autoria própria.

Salvando um arquivo CSS

Após salvar o arquivo, o mesmo estará na pasta “css” dentro da pasta do projeto, conforme imagem a seguir:



Fonte: autoria própria.

Chamando o CSS no projeto

- Para que seja possível utilizar o CSS do arquivo que criamos na nossa página index.html, antes de tudo é necessário dizer que o HTML é onde se encontra o arquivo de folhas de estilo, então é necessário criar um *link* novo.

3	<head>
4	<title>Site do Robô</title>
5	<meta charset="utf-8">
6	<link rel="stylesheet" href="CSS/ <u>styles.css</u> >
7	</head>

Fonte: autoria própria.

Link dentro do head

- Note que criamos um *link* dentro do HEAD, ou seja, ele é um elemento que não faz parte do corpo da página; além disso, colocamos qual a relação do *link*; no caso, é uma folha de estilos então é um *stylesheet*.

A sintaxe do CSS

Agora, já podemos editar o nosso arquivo CSS; para isso, vamos entender como funciona a sintaxe do CSS:



Fonte: *Sintaxe do CSS*. Adaptado de: w3schools, CSS Syntax, 2019.

A sintaxe do CSS

O seletor, no caso, pode ser uma *tag* ou uma classe (falaremos de classes mais adiante), onde as suas propriedades devem ser declaradas e, em seguida, os valores, como o exemplo da anterior, que está modificando a *tag* `<h1>` para a cor azul e com um tamanho de 12 *pixel*; esse código, explicado anteriormente, poderia ser codificado da seguinte maneira:

```
1  h1 {  
2      color: blue;  
3      font-size: 12px;  
4  }
```

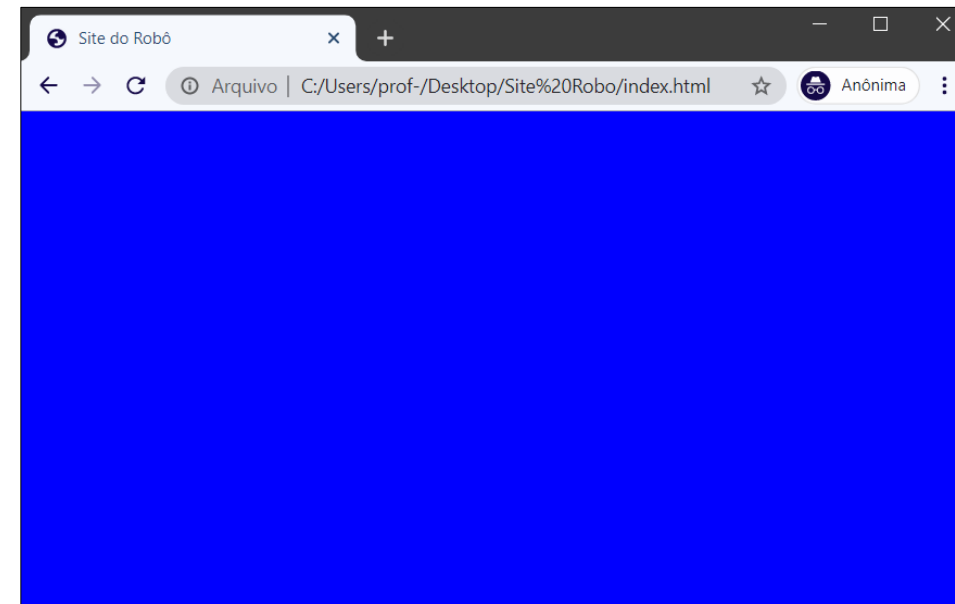
Fonte: autoria própria.

- Este código não faz parte do nosso projeto, é apenas uma amostra da estrutura do CSS.

Estilizando o <body>

- Nosso primeiro passo com o CSS, será estilizar a *tag* <body> com várias propriedades de *background*, vamos, primeiramente, alterar a cor do nosso *background* da página índex.

```
# styles.css X
c: > Users > prof- > Desktop > Site Robo > css > # styles.css >
1  ∨ body {
2    |     background-color: blue;
3    }
```




Fonte: autoria própria.

Estilizando o <body>

Podemos definir a cor de várias formas, tanto pelo nome quanto pelo código RGB ou Hexa. A seguir, um exemplo de alteração da cor utilizando o código RGB:

styles.css X

c: > Users > prof- > Desktop > Site Robo > css > # styles.css >

```
1  body {  
2  |    background-color:  rgb(254,67,20);  
3  }
```

styles.css X

c: > Users > prof- > Desktop > Site Robo > css > # styles.css >

```
1  body {  
2  |    background-color:  #3cb371;  
3  }
```

Fonte: autoria própria.

Interatividade

Quando o objetivo é garantir uma formatação uniforme em todas as páginas de um *site*, as folhas de estilo em cascata (*Cascading Style Sheets*) fazem com que o trabalho seja simples. Essa tal folha de estilo é:

- a) Um mecanismo que adiciona estilos aos elementos do JS.
- b) Um mecanismo que adiciona cor aos comandos do HTML.
- c) Um algarismo que remove estilos aos elementos do HTML.
- d) Um mecanismo que remove estilos aos comandos do HTML.
- e) Um mecanismo que adiciona estilos aos elementos do HTML.

Resposta

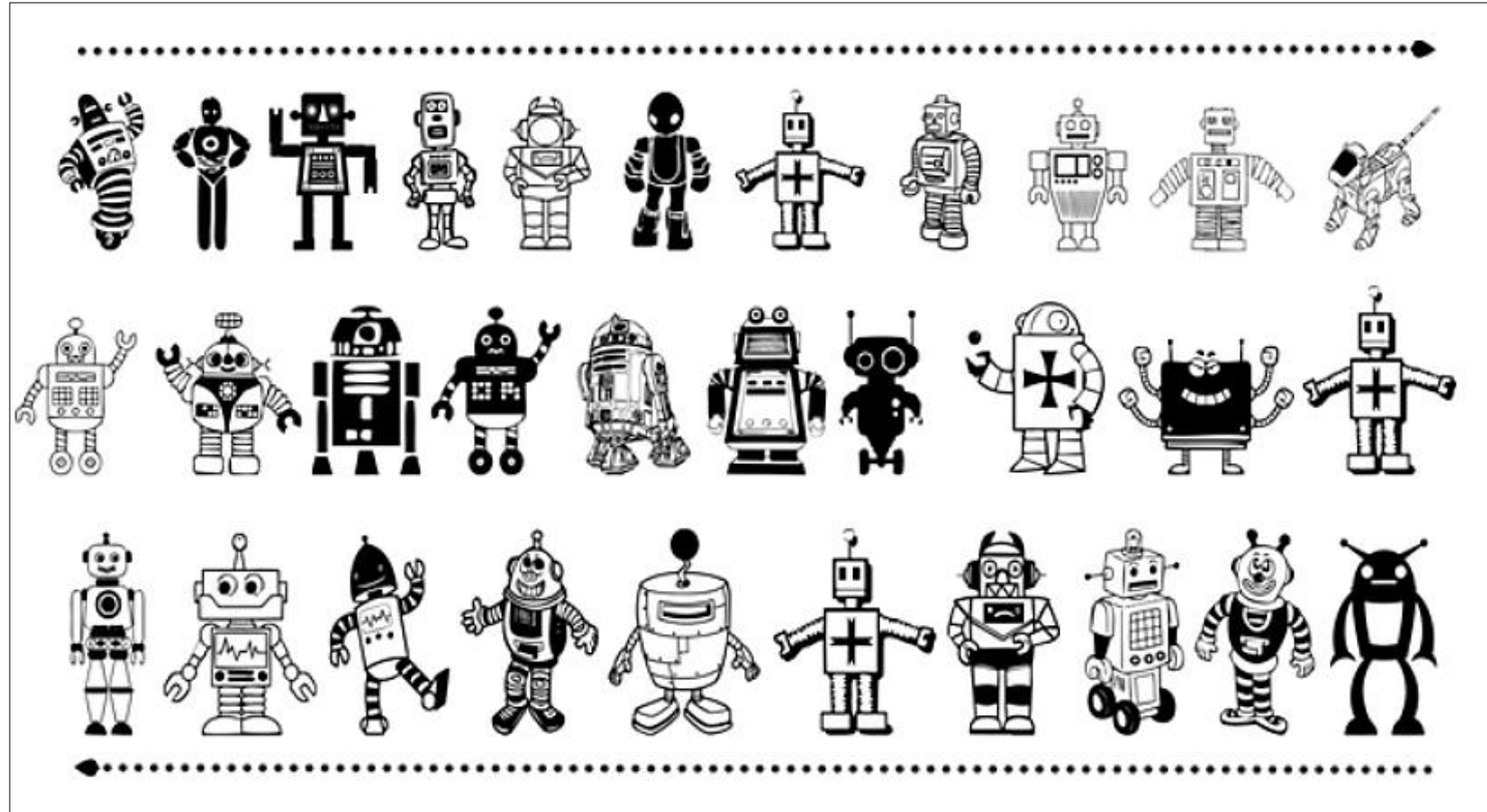
Quando o objetivo é garantir uma formatação uniforme em todas as páginas de um *site*, as folhas de estilo em cascata (*Cascading Style Sheets*) fazem com que o trabalho seja simples. Essa tal folha de estilo é:

- a) Um mecanismo que adiciona estilos aos elementos do JS.
- b) Um mecanismo que adiciona cor aos comandos do HTML.
- c) Um algoritmo que remove estilos aos elementos do HTML.
- d) Um mecanismo que remove estilos aos comandos do HTML.
- e) Um mecanismo que adiciona estilos aos elementos do HTML.

Inserindo uma imagem de *background*

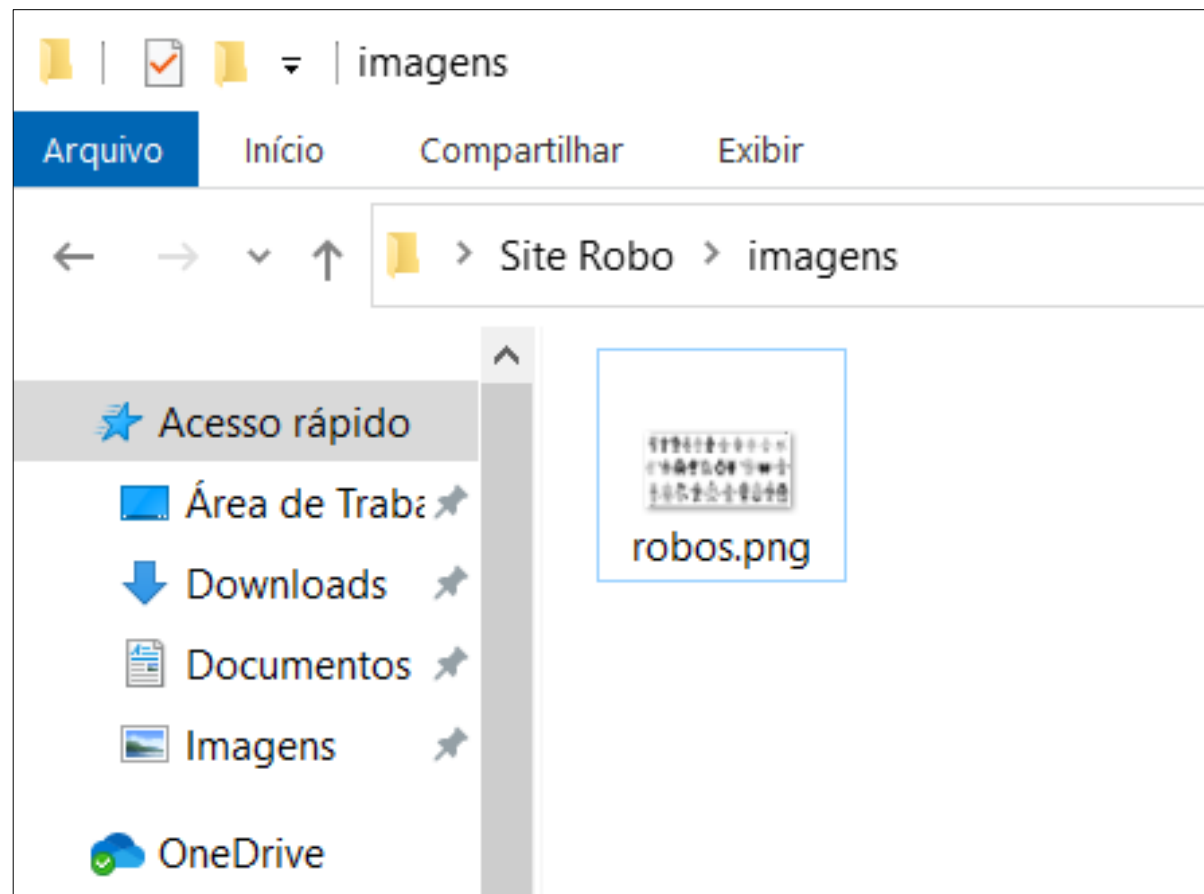
- Agora, se estava legal colocar cor no fundo de nossa página, vai ficar muito mais legal, no presente momento, pois iremos colocar imagens com o elemento *background-image*.
- Faça o *download* de uma imagem de sua preferência; no nosso caso, baixamos uma imagem de robôs.

Fonte: Pixabay.



Salvando a imagem do nosso projeto

- Logo após fazer o *download* da imagem no *site* Pixabay, salve a mesma dentro da pasta “imagens”, que foi criada dentro do projeto. Essa pasta irá armazenar todas as imagens que iremos utilizar no decorrer do projeto.



Fonte: autoria própria.

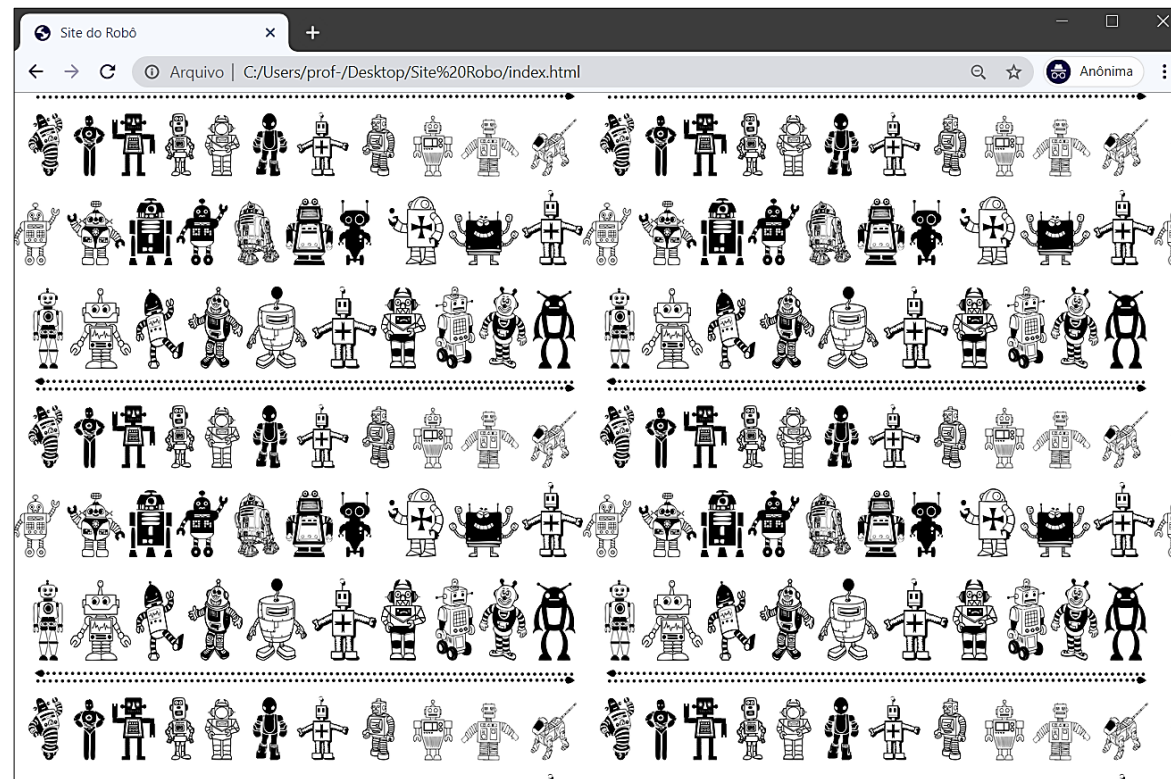
Inserindo o *background-image* no projeto

Em nosso arquivo styles.css iremos alterar:

styles.css ✕

c: > Users > prof- > Desktop > Site Robo > css > # styles.css > ...

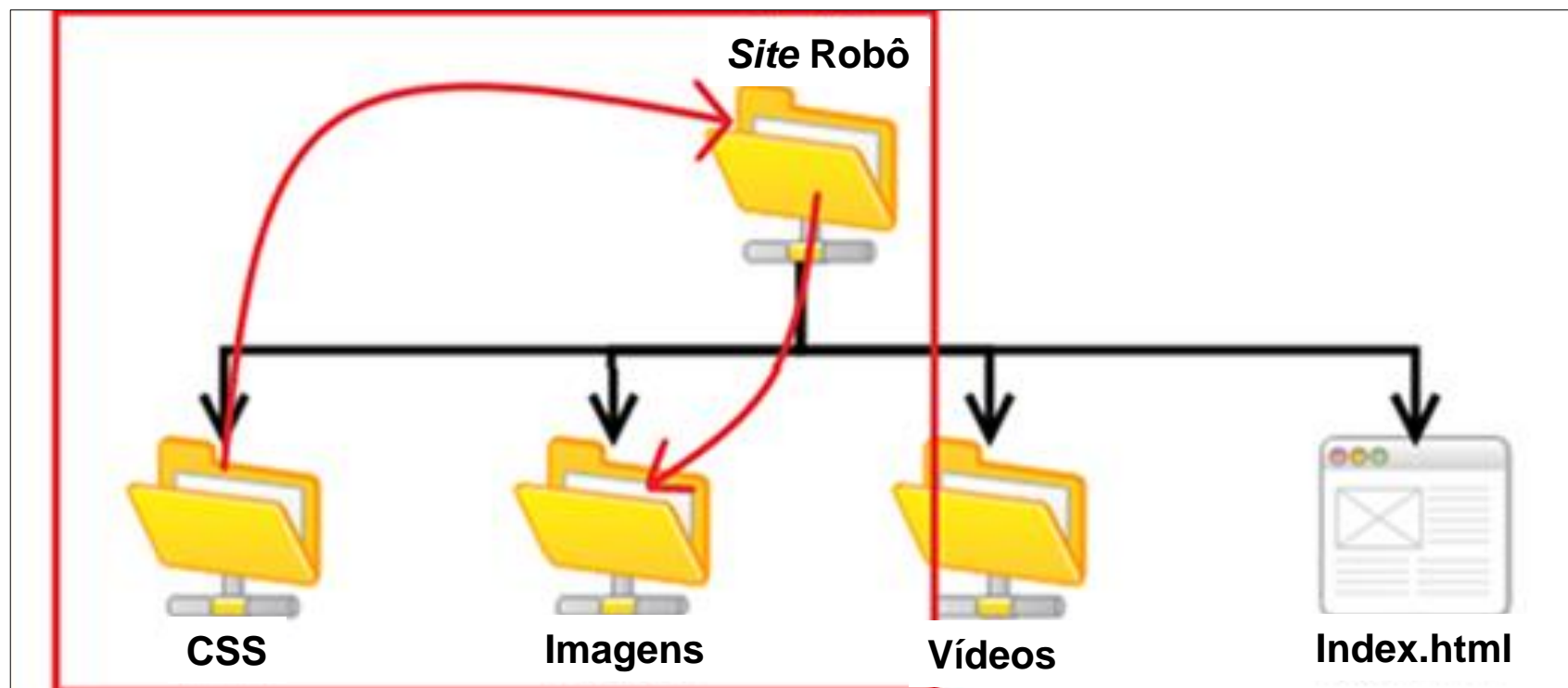
```
1  body {  
2      background-image: url("../imagens/robos.png");  
3  }
```



Fonte: autoria própria.

Navegação entre as pastas

- A página onde está o arquivo .html, que estamos editando, está dentro da pasta “páginas” e a imagem está dentro da pasta “imagens”; ambas as pastas estão na raiz do projeto, portanto o ../ faz voltarmos à raiz do projeto e, assim, colocamos o endereço da pasta de “imagens”.

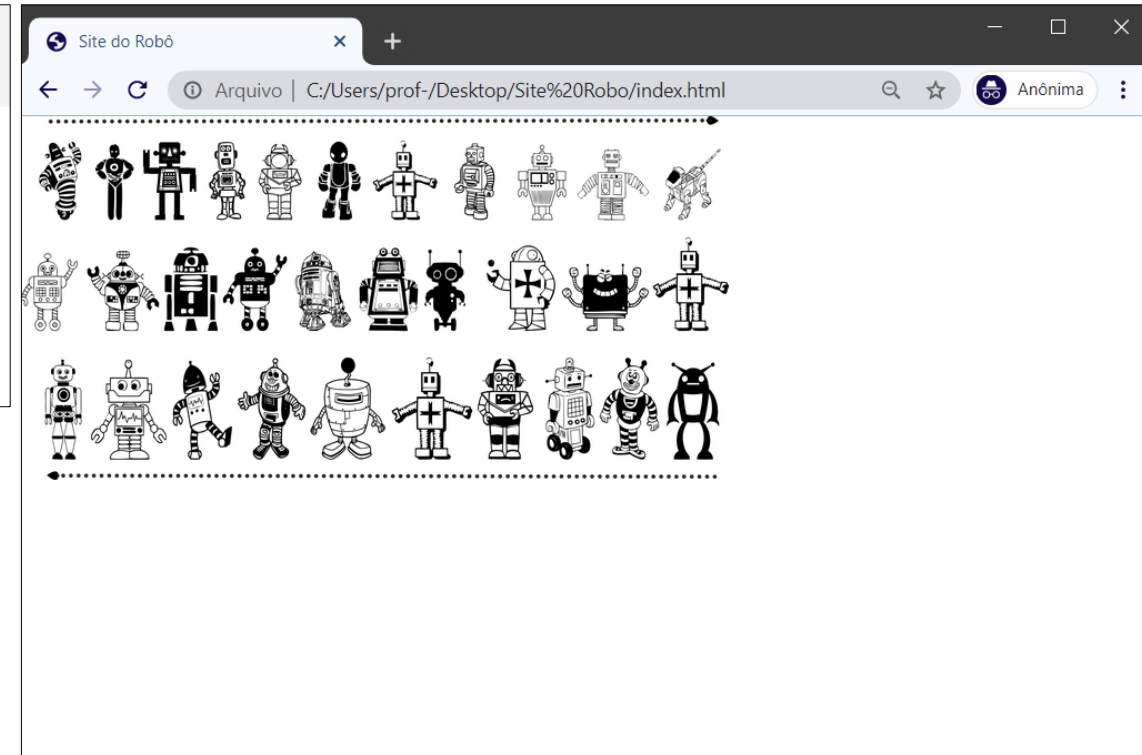


Fonte: autoria própria.

Retirando a repetição

Note, também, que a imagem por padrão está se repetindo quando executamos a página `index.html` no navegador; caso não seja essa a intenção existe a propriedade *background-repeat*.

```
# styles.css  X
c: > Users > prof- > Desktop > Site Robo > css > # styles.css > ...
1  body {
2      background-image: url("../imagens/robos.png");
3      background-repeat: no-repeat;
4  }
```



Fonte: autoria própria.

Trabalhando com duas imagens de fundo

Podemos também trabalhar com duas imagens de fundo, ao mesmo tempo, utilizando, também, a propriedade *background-repeat*. Para isso, iremos fazer o *download* de outra imagem qualquer e inseri-la na pasta “imagens” do nosso projeto. A imagem escolhida para os nossos exemplos foi a imagem a seguir; ela possui o nome `robos02.png`:

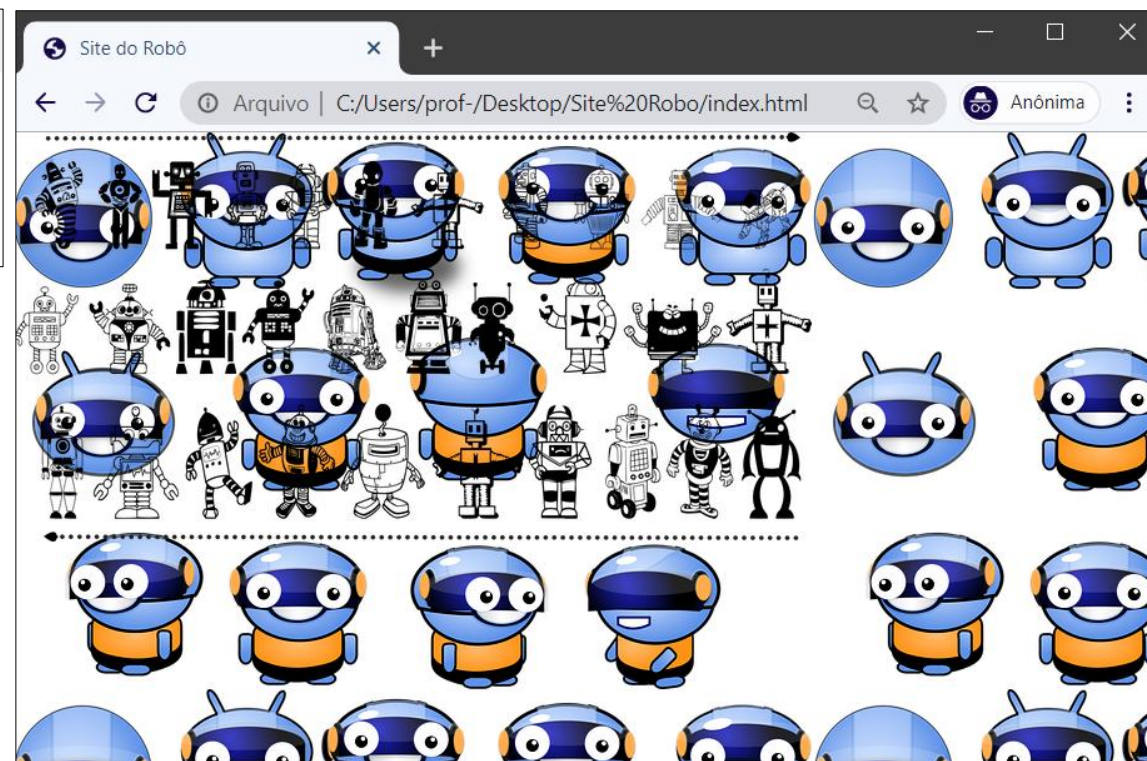


Fonte: Pixabay.

Trabalhando com duas imagens de fundo

Iremos implementar o seguinte código dentro do arquivo styles.css:

```
# styles.css  X
c: > Users > prof- > Desktop > Site Robo > css > # styles.css > ...
1  body {
2      background-image: url("../imagens/robos.png"), url("../imagens/robos02.png");
3      background-repeat: no-repeat, repeat;
4  }
```



Fonte: autoria própria.

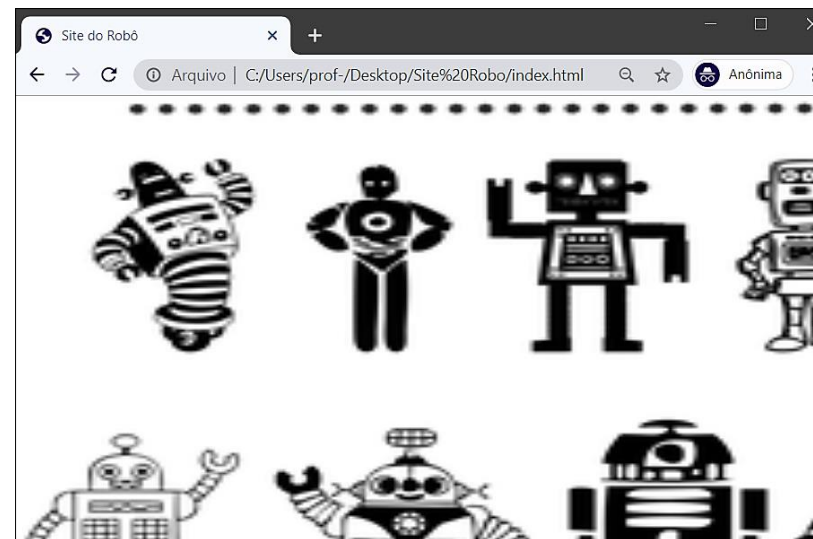
Definindo o tamanho das imagens

Podemos, também, escolher o tamanho da imagem; vamos continuar utilizando o *no-repeat* para facilitar o entendimento e vamos deixar, apenas, uma imagem como *background*:

styles.css X

c: > Users > prof- > Desktop > Site Robo > css > # styles.css > ...

```
1 body {  
2     background-image: url("../imagens/robos.png");  
3     background-repeat: no-repeat;  
4     background-size: 3000px 1000px;  
5 }
```



Fonte: autoria própria.

- Como resultado teremos uma imagem gigante e estourada, pois não fazemos ideia de quantos *pixels* têm o monitor da pessoa que acessará o nosso *site*; isso sem contar a perda de qualidade na imagem.

Definindo o tamanho das imagens

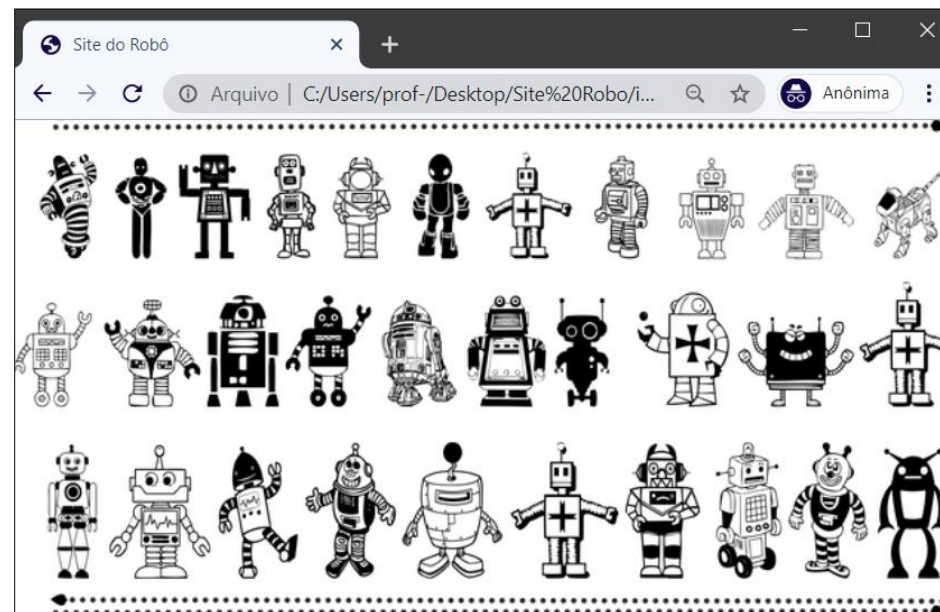
- Para conseguirmos colocar a imagem no plano de fundo inteiro e distorcê-la o menos possível, podemos utilizar, no lugar de valores em *pixels*, valores em porcentagem, e agora ativando o *repeat* novamente.

styles.css

c: > Users > prof- > Desktop > Site Robo > css > # styles.css > ...

```
1  body {  
2      background-image: url("../imagens/robos.png");  
3      background-repeat: repeat;  
4      background-size: 100%;  
5  }
```

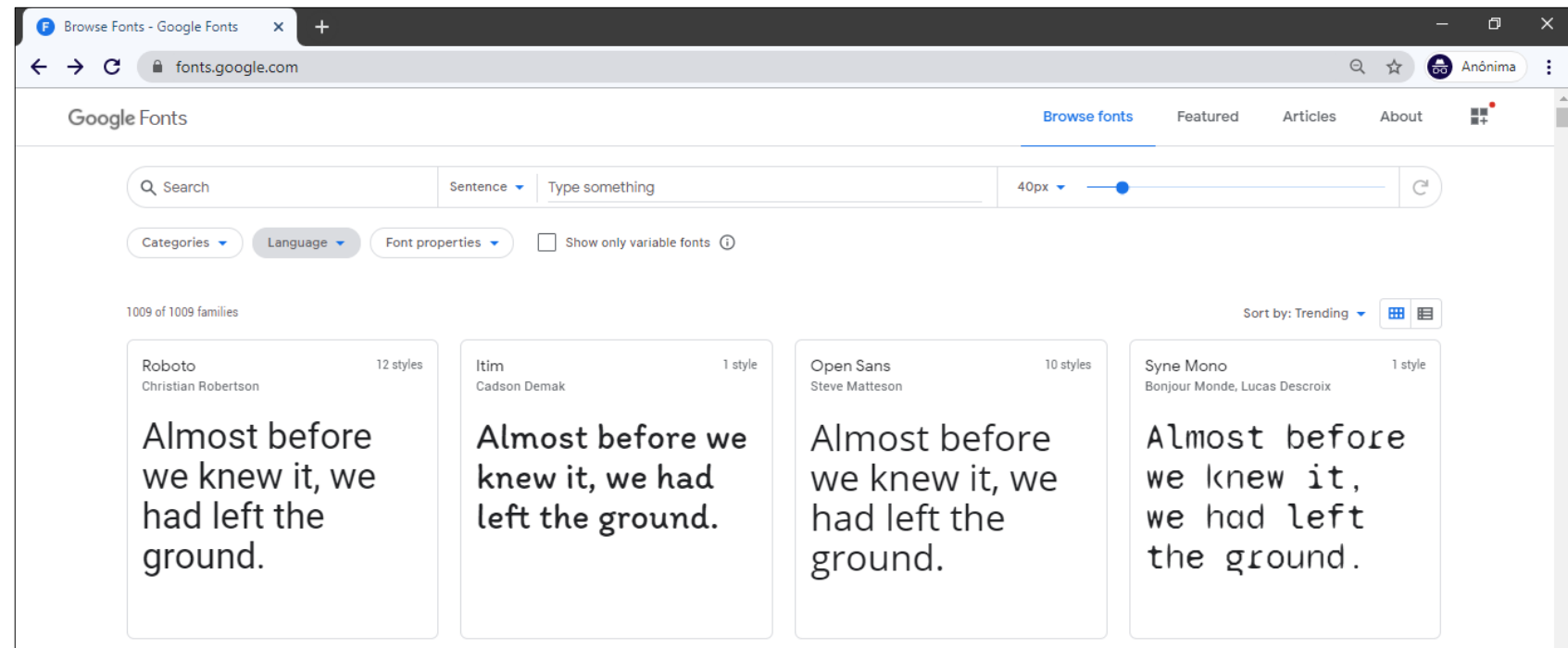
Fonte: autoria própria.



- Note que, no caso, a imagem ocupará 100% da página, não importando o tamanho do *browser*. Agora que já aprendemos um pouco sobre o *background-size*, vamos adotar essa configuração para continuar o nosso projeto.

Utilizando fontes de terceiros

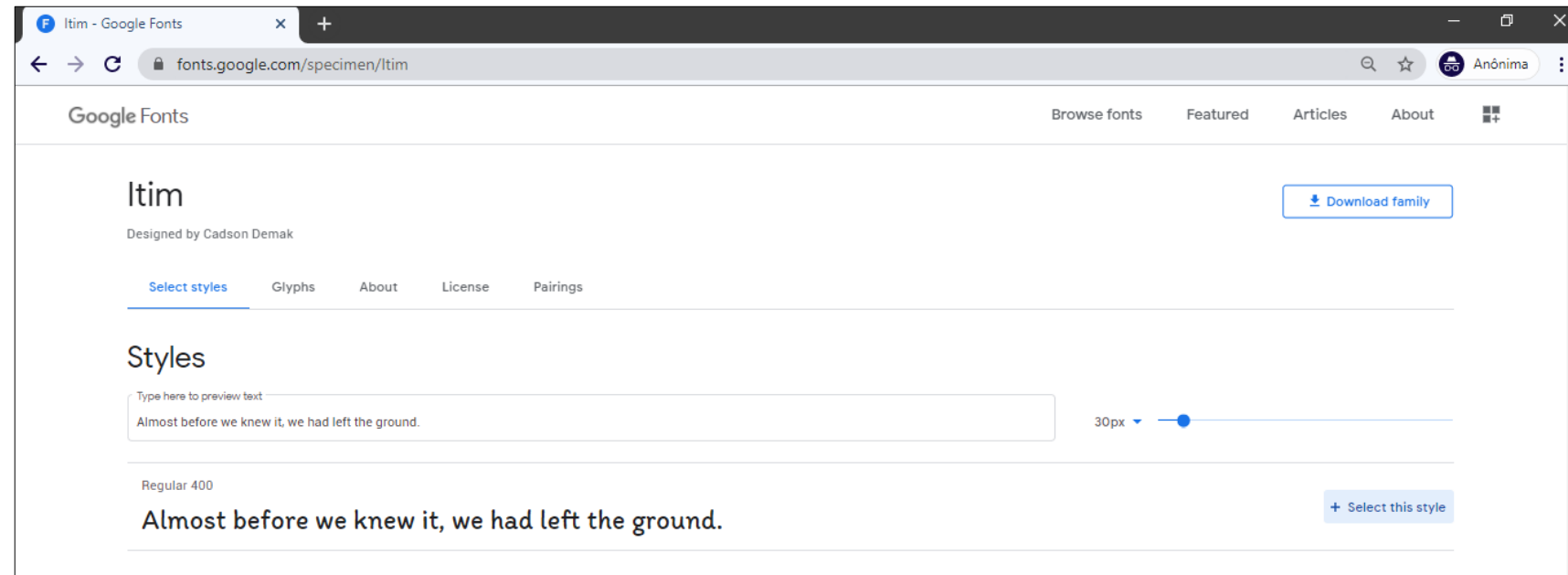
- Para melhorar o nosso *site*, precisamos de fontes novas e mais legais; para isso, entraremos no *site* <https://fonts.google.com/> e procuraremos alguma fonte bacana.



Fonte: <https://fontes.google.com>

Utilizando fontes de terceiros

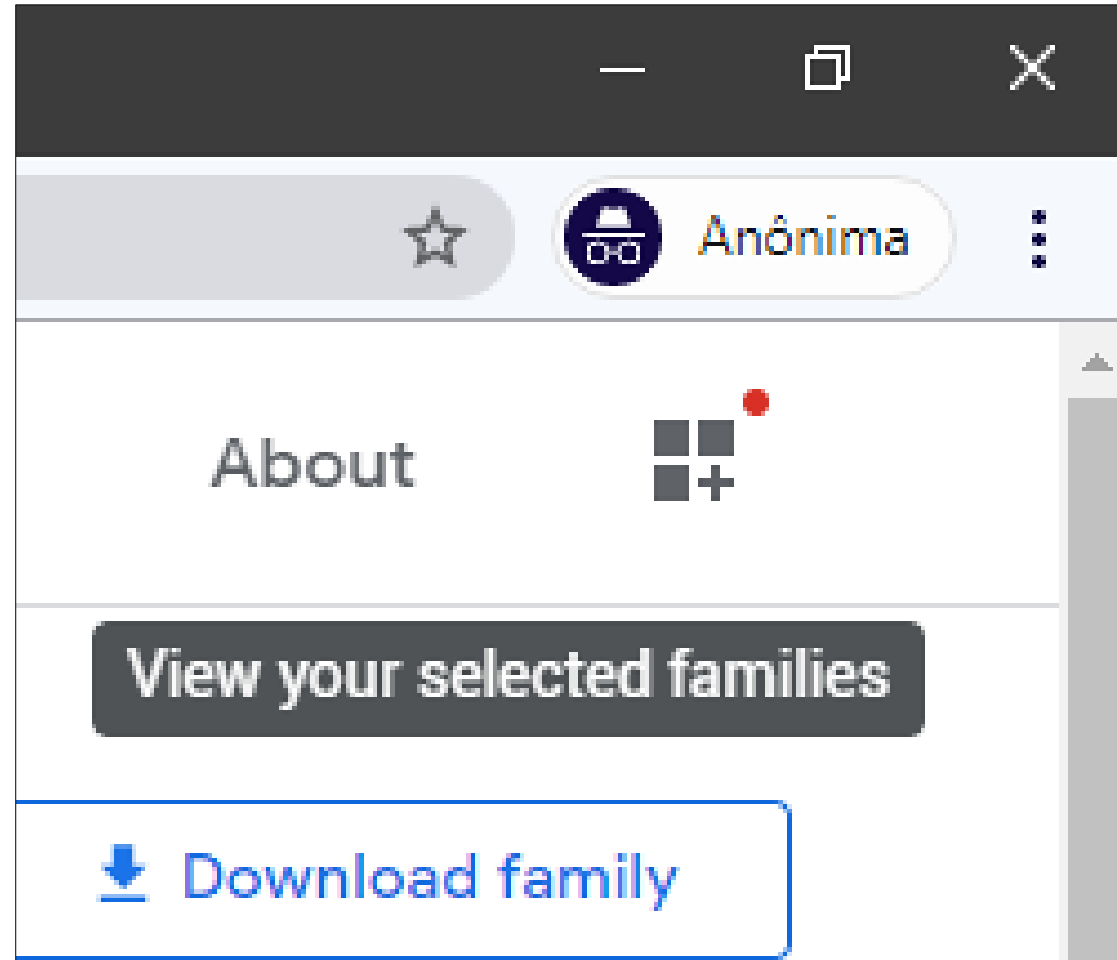
Ao escolhermos a nossa fonte nós clicaremos no símbolo de + *Selected this type*, que aparece logo ao lado direito do nome dela:



Fonte: <https://fontes.google.com>

Utilizando fontes de terceiros

Você irá perceber um ponto vermelho no logo, no canto direito superior. Se parar o *mouse* em cima do logo, irá ver a opção *View your selected families*:



Fonte: <https://fontes.google.com>

Utilizando fontes de terceiros

Clicando na opção, abrirá uma pequena janela com diversas opções para escolhermos, referentes à fonte que selecionamos. Note que na opção <link> poderemos copiar um bloco de código para inserir em nossa página. É exatamente isso que iremos fazer:

Selected family

Review

Itim

Use on the web

To embed a font, copy the code into the <head> of your html

☒ <link> ☐ @import

```
<link href="https://fonts.googleapis.com/css2?family=Itim&display=swap" rel="stylesheet">
```

CSS rules to specify families

```
font-family: 'Itim', cursive;
```

[API docs](#) [Download all](#)

Fonte: <https://fontes.google.com>

Utilizando fontes de terceiros

Assim sendo, nós adicionaremos o *link* a nossa página HTML index dentro do *head*:

```
3  <head>
4      <title>Site do Robô</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="CSS/styles.css">
7      <link href="https://fonts.googleapis.com/css2?family=Itim&display=swap" rel="stylesheet">
8  </head>
```

Fonte: autoria própria.

- Dessa forma, teremos o acesso a essa fonte durante a confecção de nossa página. O nome da fonte que importamos é *Itim*; mais adiante, esse nome será importante.

Interatividade

Ao criarmos um arquivo CSS dentro de uma subpasta de nosso projeto principal nós acabamos alterando o caminho que o mesmo se encontrava anteriormente; para isso, usamos o comando `../` dentro do caminho, para que:

- a) Seja possível excluir uma pasta antes do caminho apontado.
- b) Seja possível retornar uma pasta antes do caminho apontado.
- c) Seja possível avançar uma pasta depois do caminho apontado.
- d) Seja possível excluir um arquivo antes do caminho apontado.
- e) Seja possível retornar um arquivo antes do caminho apontado.

Resposta

Ao criarmos um arquivo CSS dentro de uma subpasta de nosso projeto principal nós acabamos alterando o caminho que o mesmo se encontrava anteriormente; para isso, usamos o comando `../` dentro do caminho, para que:

- a) Seja possível excluir uma pasta antes do caminho apontado.
- b) Seja possível retornar uma pasta antes do caminho apontado.**
- c) Seja possível avançar uma pasta depois do caminho apontado.
- d) Seja possível excluir um arquivo antes do caminho apontado.
- e) Seja possível retornar um arquivo antes do caminho apontado.

Estilizando a *tag* <main>

- Apenas o ato de abrir e fechar a *tag* já é suficiente para que possamos trabalhar o CSS dela; então, vamos lá: no nosso arquivo CSS digite o seguinte código:

```
7   main {  
8       background-color: ■ rgba(0, 0, 0, 0.6);  
9       padding: 5px 40px;  
10      margin-top: 20px;  
11      border-radius: 10px;  
12  }
```

Fonte: autoria própria.

Estilizando a *tag* <main>

Vamos, agora, entender cada parte que será feita nesta estilização. Como seletor temos a *tag main*, ou seja, tudo o que estiver dentro da *tag main* receberá estes estilos, que são:

8

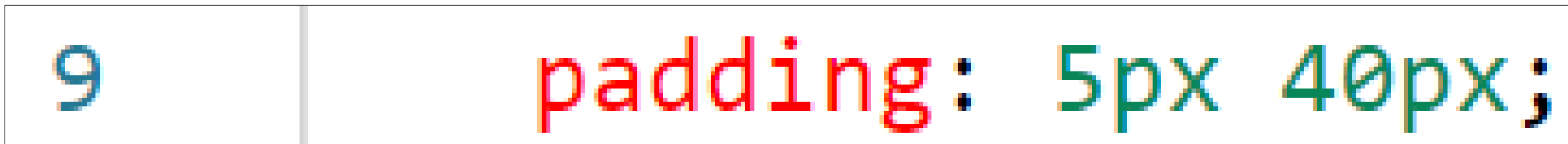
```
background-color:  rgba(0, 0, 0, 0.6);
```

Fonte: autoria própria.

- Nós já aprendemos a como escrever cores em RGB, mas, desta vez, colocamos RGBA; isso significa que o canal *alpha* está presente. Daí, você me pergunta: “o que é o canal *alpha*?” E a resposta é simples: canal *alpha* é aonde se encontra a transparência.

Estilizando a *tag* <main>

- O *padding* existe para nos ajudar com o espaçamento entre o conteúdo e as bordas; diferentemente da margem que dá um espaçamento externo, o *padding* dá um espaçamento interno.



Fonte: autoria própria.

Estilizando a *tag* <main>

- O *margin-top* cria uma margem com o topo do *site*, nós colocamos em *pixels*, mas, também, pode ser colocado em porcentagem.

10

margin-top: 20px;

Fonte: autoria própria.

Estilizando a *tag* <main>

- E, por fim, temos o *border-radius* que define o arredondamento das esquinas do *main* utilizando, como referência, o raio das bordas.

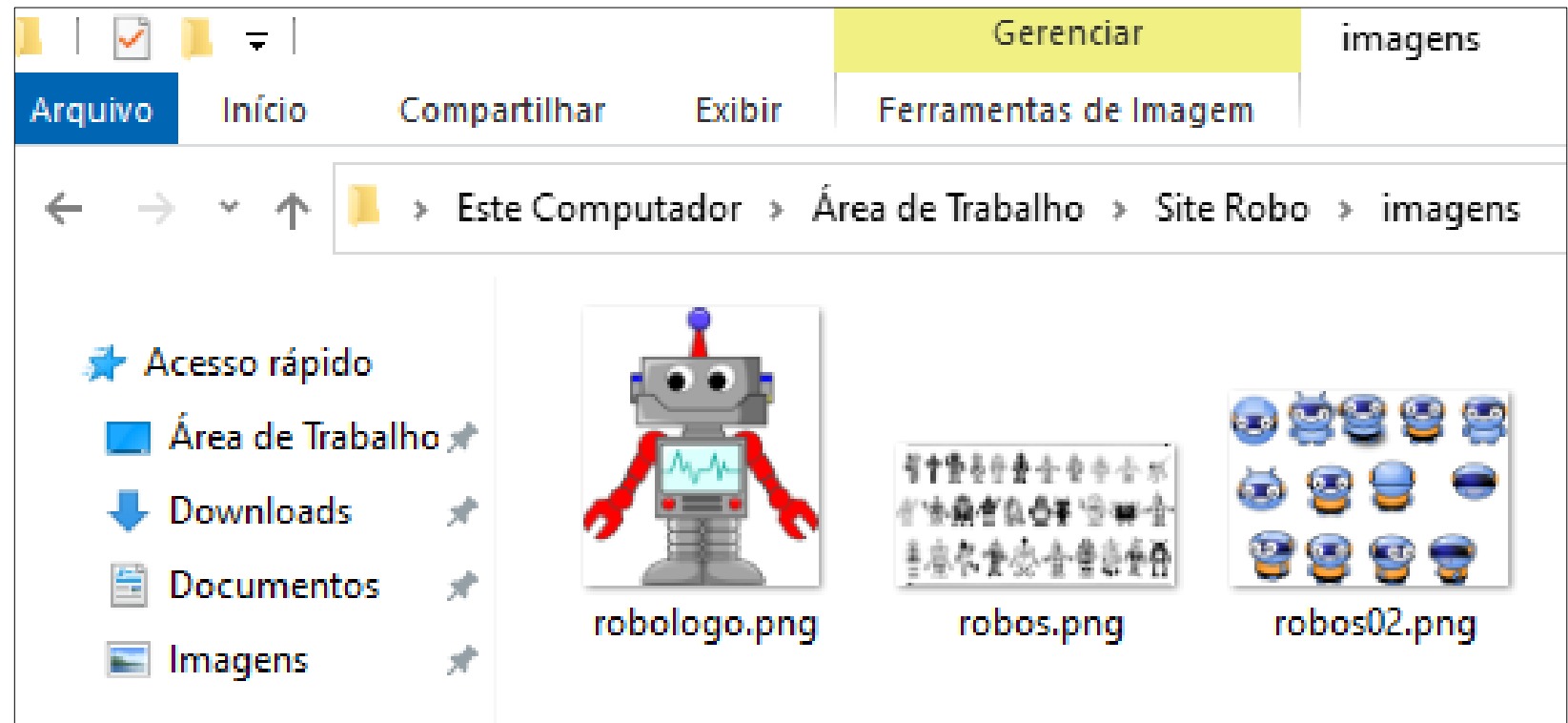
11

```
border-radius: 10px;
```

Fonte: autoria própria.

Containers <div> e a classificação de tag

Vamos, agora, dividir o nosso conteúdo em pequenos *containers* que, como o nome já sugere, são espaços separados onde o conteúdo ficará guardado. Usaremos o logo a seguir; será necessário que você faça o *download* de um logo de sua escolha e salve na pasta “imagens”:



Fonte: autoria própria.

Containers <div> e a classificação de tag

Agora que já possuímos a imagem para o nosso logo, vamos digitar o seguinte código na nossa página index:

10		<code><main></code>
11		<code> <div class="logo"></code>
12		<code> </code>
13		<code> </div></code>
14		<code></main></code>

Fonte: autoria própria.

Containers <div> e a classificação de *tag*

Essa <div> que colocamos foi classificada como “logo”; o nome “logo” foi dado por nós, se colocássemos qualquer nome iria funcionar da mesma forma. Dentro dessa <div> colocamos a imagem referente ao logo da nossa página e, então, editaremos o seu estilo no CSS:

```
14      .logo{  
15          |      margin: 0;  
16          |      text-align: center;  
17      }
```

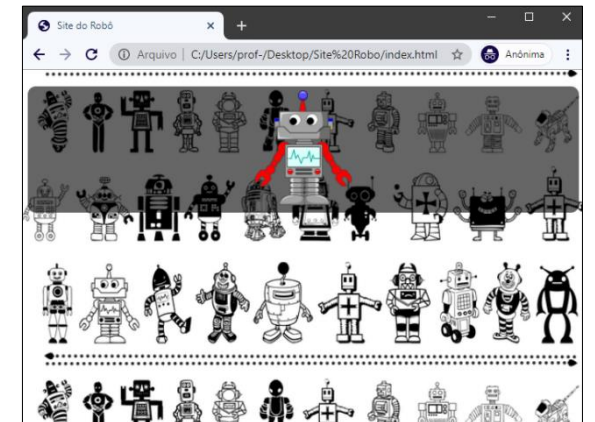
Fonte: autoria própria.

Containers <div> e a classificação de tag

- O item *margin* define o espaçamento entre os elementos, enquanto o item *text-align* faz o alinhamento do texto, que, no caso, queremos que fique no centro. Independente de colocarmos uma imagem ou um texto, o alinhamento será feito da mesma maneira.

15	<code>margin: 0;</code>
16	<code>text-align: center;</code>

Fonte: autoria própria.



Estilização de componentes

- Note que, dessa vez, ao invés de usar como seletor a *tag* que vamos modificar, nós utilizamos a classe que criamos, pois, na mesma página, teremos várias `<div>` que podem ou não ser diferentes; sendo assim, o mais prudente é colocar a classe como seletor.

Interatividade

Nós já aprendemos como escrever cores em_____, mas, desta vez, colocamos RGBA, isso significa que o canal _____ está presente, que é aonde se encontra a _____. O valor dela é medido na _____ entre 0 e 1, sendo 0, totalmente visível, e 1, totalmente invisível.

Assinale a alternativa que preenche corretamente as lacunas do parágrafo:

- a) HTML, *foxtrot*, transparência, variação.
- b) HTML, beta, displicência, continuação.
- c) RGB, *alpha*, transparência, variação.
- d) RGB, gama, displicência, continuação.
- e) PDF, beta, transparência, variação.

Resposta

Nós já aprendemos como escrever cores em_____, mas, desta vez, colocamos RGBA, isso significa que o canal _____ está presente, que é aonde se encontra a _____. O valor dela é medido na _____ entre 0 e 1, sendo 0, totalmente visível, e 1, totalmente invisível.

Assinale a alternativa que preenche corretamente as lacunas do parágrafo:

- a) HTML, *foxtrot*, transparência, variação.
- b) HTML, beta, displicência, continuação.
- c) **RGB, *alpha*, transparência, variação.**
- d) RGB, gama, displicência, continuação.
- e) PDF, beta, transparência, variação.

Estilizando as *tags* de título

Vamos, agora, voltar a trabalhar com os títulos, mudando as suas propriedades com classes que nós já aprendemos a fazer; insira o seguinte código na página index:

```
10      <main>
11      |      <div class="logo">
12      |      |      
13      |      </div>
14      |      <h1 class="manchete">Robô</h1>
15      |      <h2 class="headline">O que ele consegue fazer?</h2>
16      </main>
```

Fonte: autoria própria.

Estilizando as *tags* de título

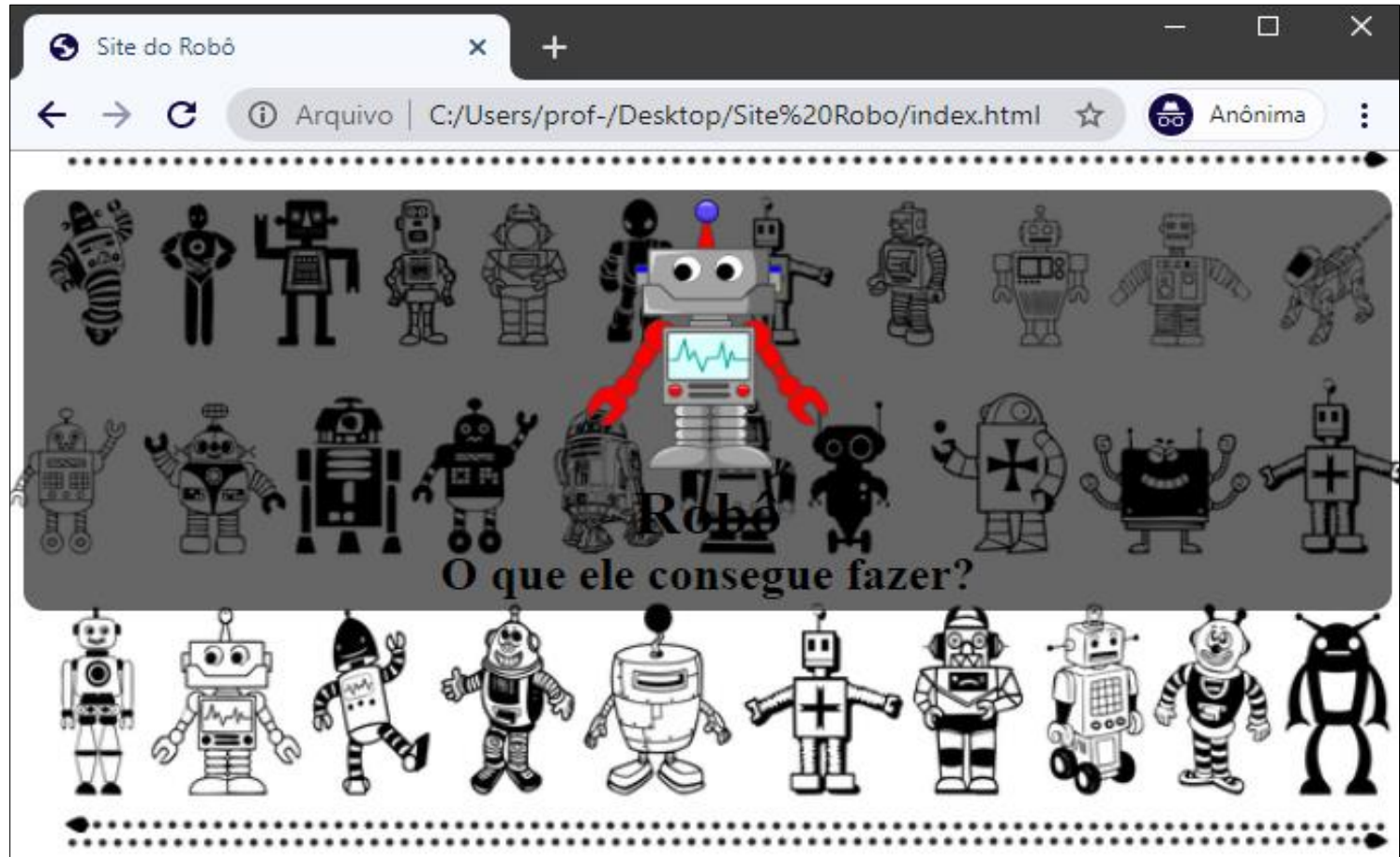
- Podemos colocar essas classificações com as mesmas características, então, nós só iremos adicionar mais classes na mesma declaração no CSS.

```
19  .logo, .manchete, .headline{  
20      margin: 0;  
21      text-align: center;  
22  }
```

Fonte: autoria própria.

Estilizando as *tags* de título

- Mas note que, no navegador, é possível perceber que a cor da fonte pode apresentar um conflito com a cor do fundo; sendo assim, vamos mudá-la para facilitar a visualização do texto.



Fonte: autoria própria.

Estilizando as *tags* de título

- Como o problema é generalizado para todas as classes, o mais sensato é colocarmos a estilização dentro do <body>, para que todas as alterações a seguir recebam os estilos herdados dela.

```
1  body {  
2      background-image: url("../imagens/robos.png");  
3      background-repeat: repeat;  
4      background-size: 100%;  
5      font-family: 'Itim';  
6      color: white;;  
7  }
```

Fonte: autoria própria.

Estilizando as *tags* de título

- Note que estamos usando, agora, a fonte que baixamos com o elemento *font-family* e também o *color* para pintarmos a nossa fonte de branco.

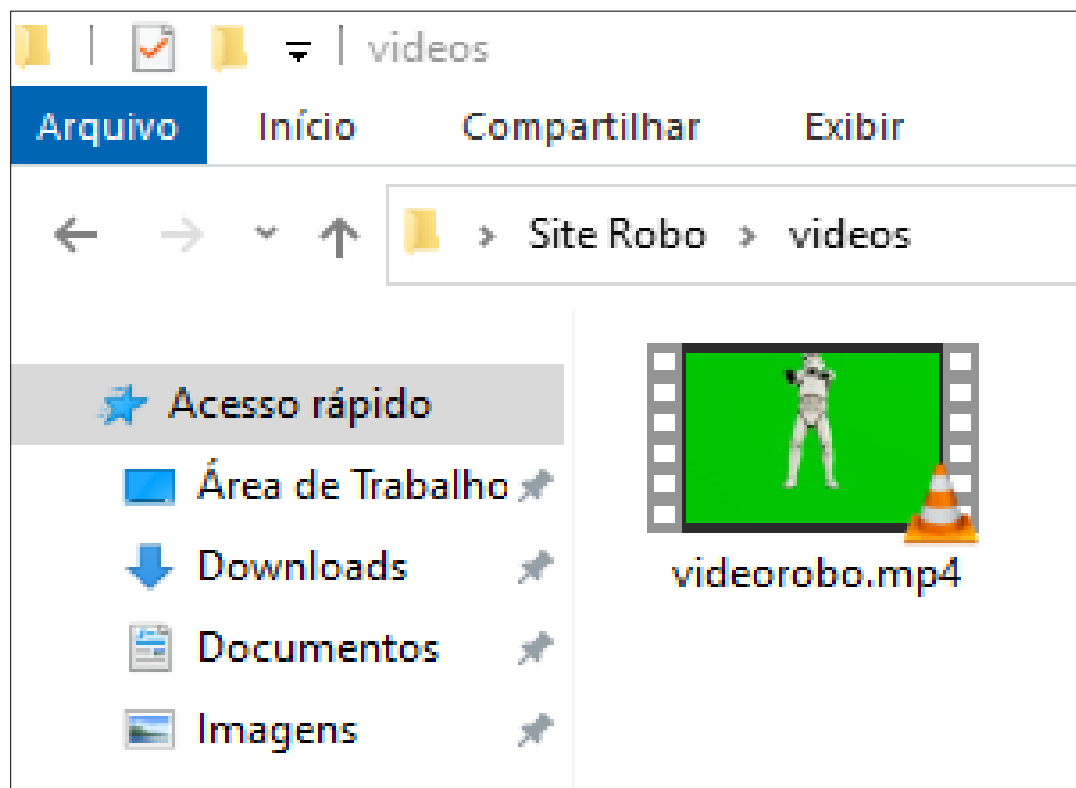


Fonte: autoria própria.

Inserindo os vídeos na página

- Vamos, agora, colocar vídeos na nossa página html; para isso, você precisará encontrar um vídeo para o *site*.

Em nosso caso, a nossa pasta de vídeos ficou da seguinte maneira:



Fonte: autoria própria.

Inserindo os vídeos na página

Agora que já possuímos o vídeo do robô, vamos inserir o seguinte código na página index.html na sequência da *tag* <main>:

18			<div class="video">
19			<video width="320" height="240" controls="controls">
20			<source src="videos/videorobo.mp4" type="video/mp4"/>
21			</video>
22			</div>

Fonte: autoria própria.

Inserindo os vídeos na página

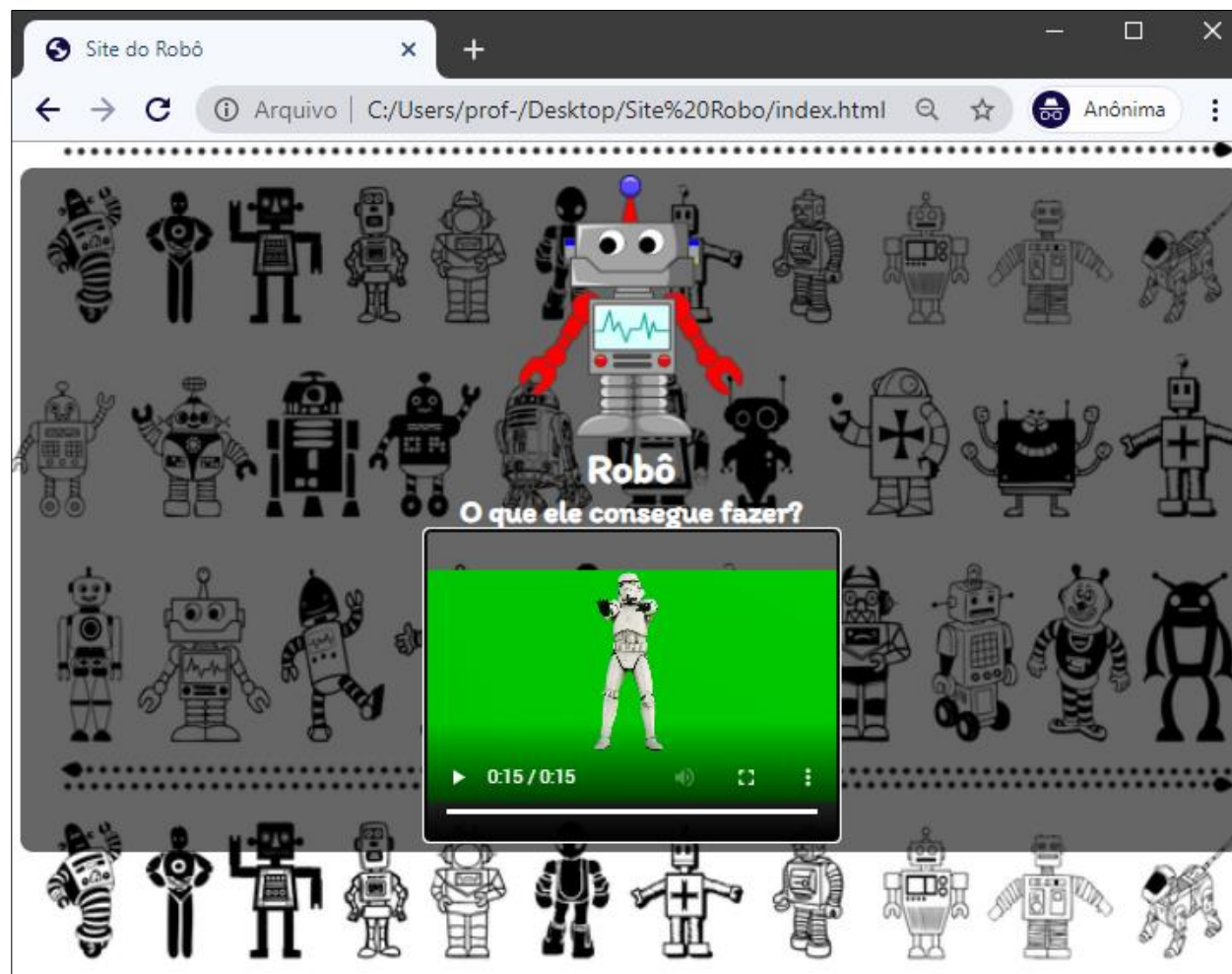
- Assim como a *tag* busca uma imagem, a *tag* <source> busca um recurso, que, no caso, é um vídeo. Declaramos no CSS separadamente das outras de preferência, pois pode ser que seja necessário aplicar algumas mudanças com o tempo, então, vamos deixá-la com as mesmas características do logo, mas em uma declaração diferente.

```
26     .video{  
27         |     margin:0;  
28         |     text-align: center;  
29     }
```

Fonte: autoria própria.

Inserindo os vídeos na página

O resultado disso é o nosso *site*, agora com o recurso de vídeo:



Fonte: autoria própria.

Estilizando a *tag* <article>

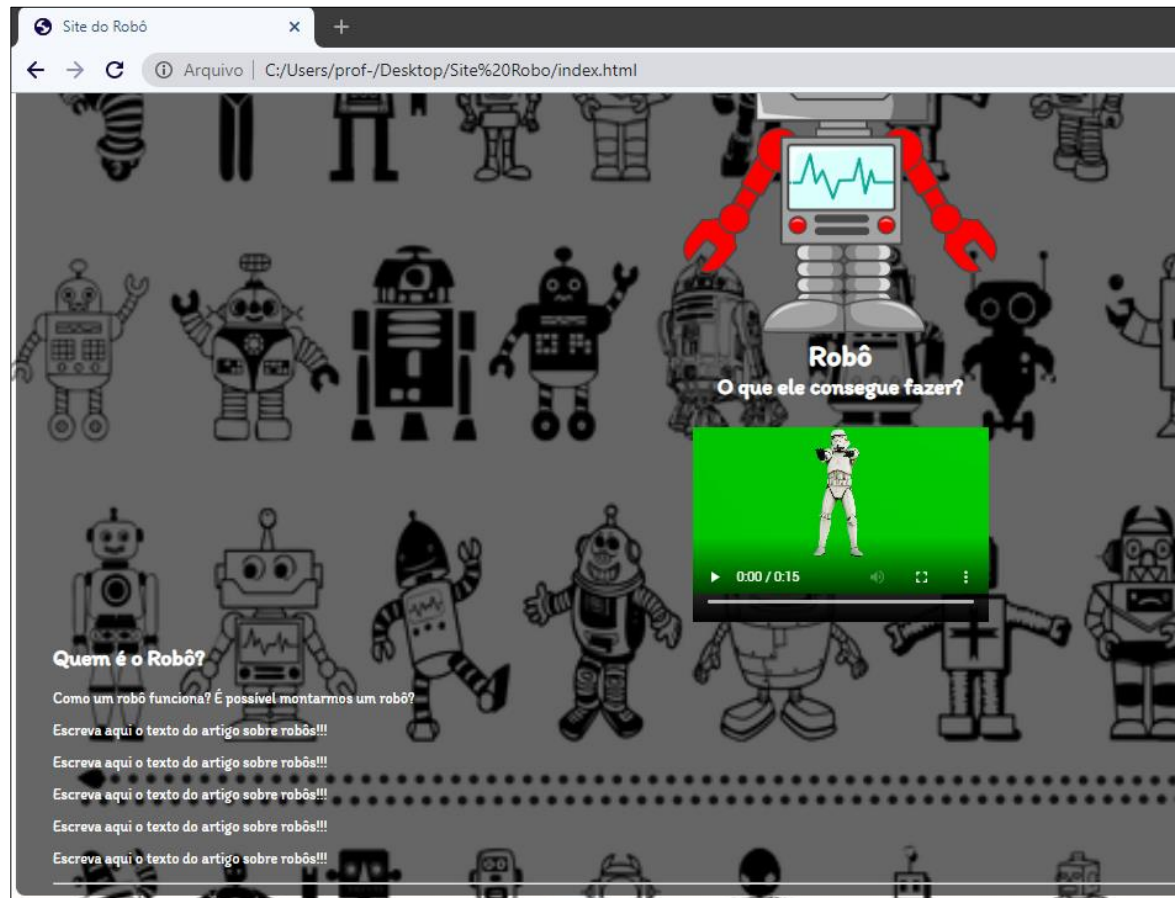
Agora, vamos fazer o nosso primeiro artigo do *site*, contendo os parágrafos e a imagem; para isso, insira o seguinte código na sequência da *tag* <main>, na página *index*:

24				<article>
25				<h1>Quem é o Robô?</h1>
26				<p>Como um robô funciona? É possível montarmos um robô?</p>
27				<p>Escreva aqui o texto do artigo sobre robôs!!!</p>
28				<p>Escreva aqui o texto do artigo sobre robôs!!!</p>
29				<p>Escreva aqui o texto do artigo sobre robôs!!!</p>
30				<p>Escreva aqui o texto do artigo sobre robôs!!!</p>
31				<p>Escreva aqui o texto do artigo sobre robôs!!!</p>
32				</article>
33				<hr>

Fonte: autoria própria.

Estilizando a *tag* <article>

Mesmo sem colocar nenhum estilo na *tag* <article>, podemos ver que ela já está herdando estilos do <body>, utilizamos, também, a *tag* <hr> que faz uma linha entre um conteúdo e outro:



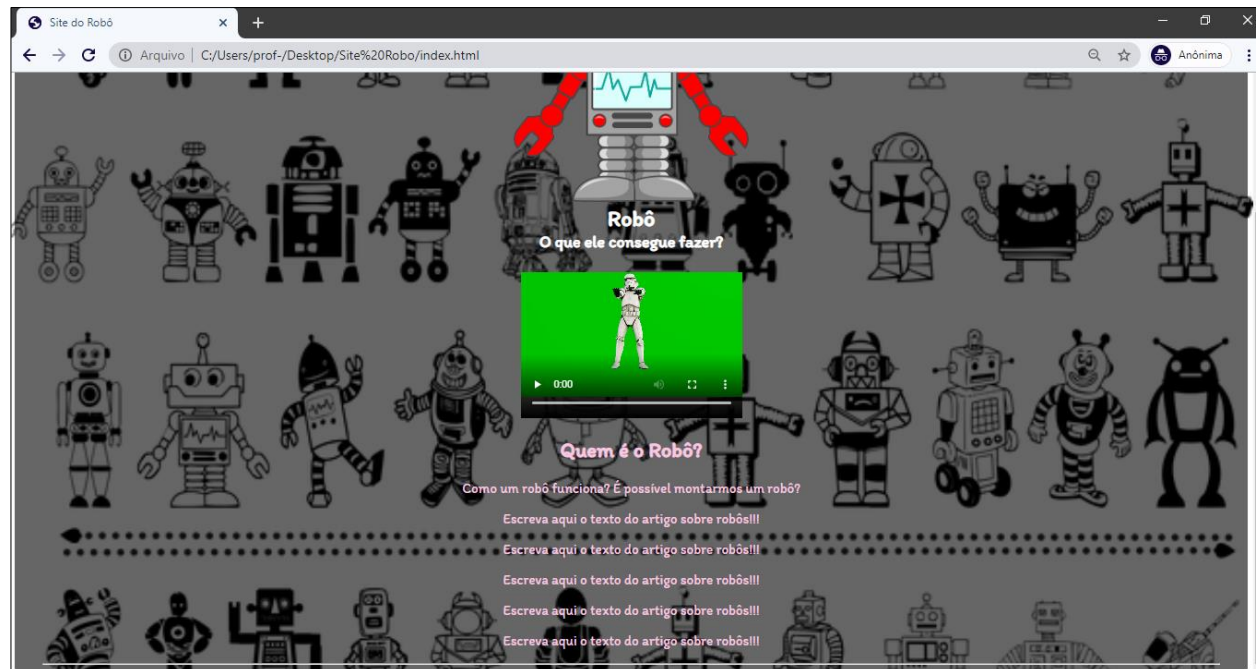
Fonte: autoria própria.

Estilizando a *tag* <article>

Para mudar isso, precisamos sobrepor os estilos:

- Nós alinhamos o texto no centro, mudamos o estilo da fonte, deixamos o tamanho da fonte como 20 *pixels*, além de deixarmos a fonte em uma cor de tom rosado.

```
31 article{  
32     text-align: center;  
33     font-style: normal;  
34     font-size: 20px;  
35     color: ■ rgb(250,200,225);  
36 }
```



Fonte: autoria própria.

Interatividade

Como qualquer tipo de linguagem de marcação, o CSS possui uma sintaxe própria e bem definida, sendo possível identificar, exatamente, como ela funciona com as suas propriedades. Com base no texto, assinale como verdadeiro (V) ou falso (F) as afirmações a seguir:

- () `h1` é um exemplo de seletor.
- () `{color:blue}` é um exemplo de declaração.
- () `12px` é um exemplo de valor.
- () *font-size* é um exemplo de propriedade.

Agora, assinale a alternativa que apresenta a sequência correta:

- a) V, V, V, F.
- b) V, V, F, F.
- c) V, V, V, V.
- d) F, F, F, V.
- e) F, F, V, V.

Resposta

Como qualquer tipo de linguagem de marcação, o CSS possui uma sintaxe própria e bem definida, sendo possível identificar, exatamente, como ela funciona com as suas propriedades. Com base no texto, assinale como verdadeiro (V) ou falso (F) as afirmações a seguir:

- (V) `h1` é um exemplo de seletor.
- (V) `{color:blue}` é um exemplo de declaração.
- (V) `12px` é um exemplo de valor.
- (V) `font-size` é um exemplo de propriedade.

Agora, assinale a alternativa que apresenta a sequência correta:

- a) V, V, V, F.
- b) V, V, F, F.
- c) V, V, V, V.
- d) F, F, F, V.
- e) F, F, V, V.

ATÉ A PRÓXIMA!