

03 - Criando classes em HTML e melhorando a disposição dos elementos

Relembrando

Na aula anterior eu havia começado a desenvolver o site de receitas da Dona Maria e você devia continuar o processo.

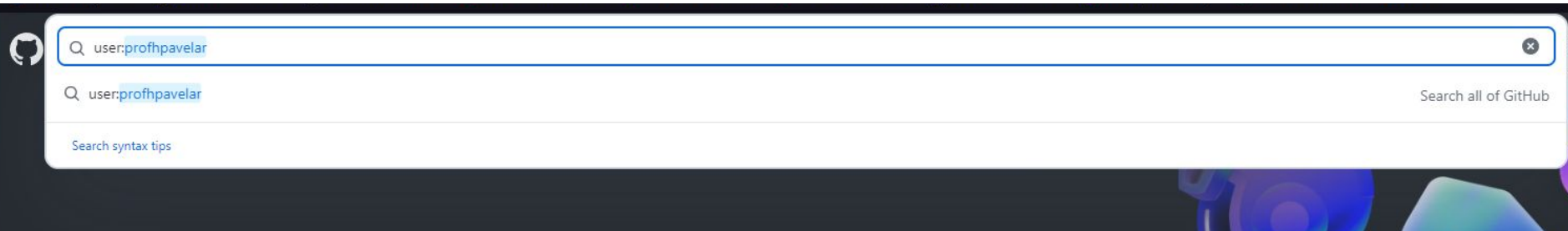
No final, já havia uma evolução considerável, mas ainda existiam problemas claros como, por exemplo, o tamanho da imagem utilizada.

Nesta aula vamos melhorar a disposição dos elementos na página. Iremos abordar propriedades como *display*, *justify-content*, *margin* e *padding*. Também criaremos algumas classes nos elementos HTML para poder trabalhar de forma mais eficiente no CSS.

1. Baixando os arquivos do GitHub

Se você conseguiu *commitar* os arquivos da aula anterior no GitHub, poderá baixá-los da sua própria conta. Caso tenha me enviado por e-mail, poderá procurar pelos arquivos na sua aba de e-mails enviados.

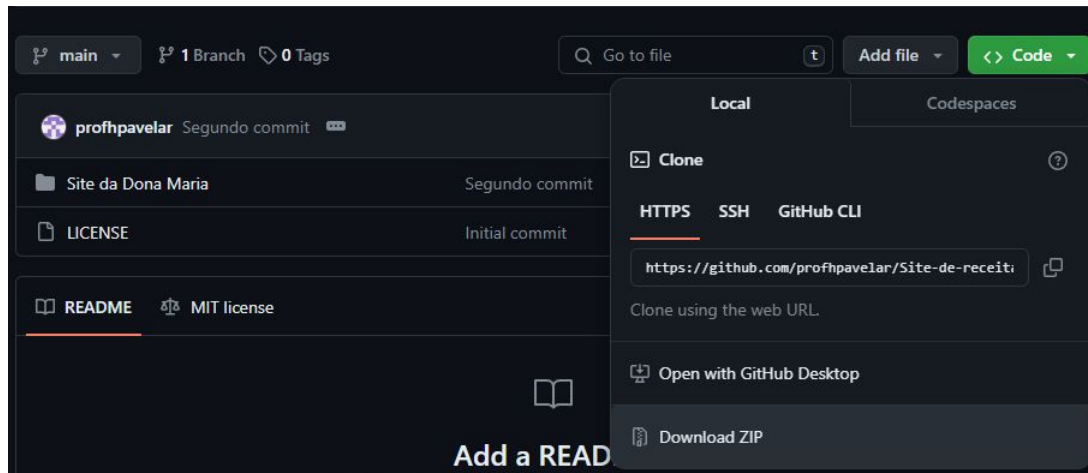
Outra opção é baixar os arquivos do meu repositório no GitHub. VOCÊ NÃO PRECISA FAZER LOGIN POR ENQUANTO. Basta entrar no site do GitHub e, na barra de pesquisa, digitar: user:profhpavelar para entrar no meu perfil.



Na página que aparecer, entre no repositório Site-de-receitas-da-Dona-Maria

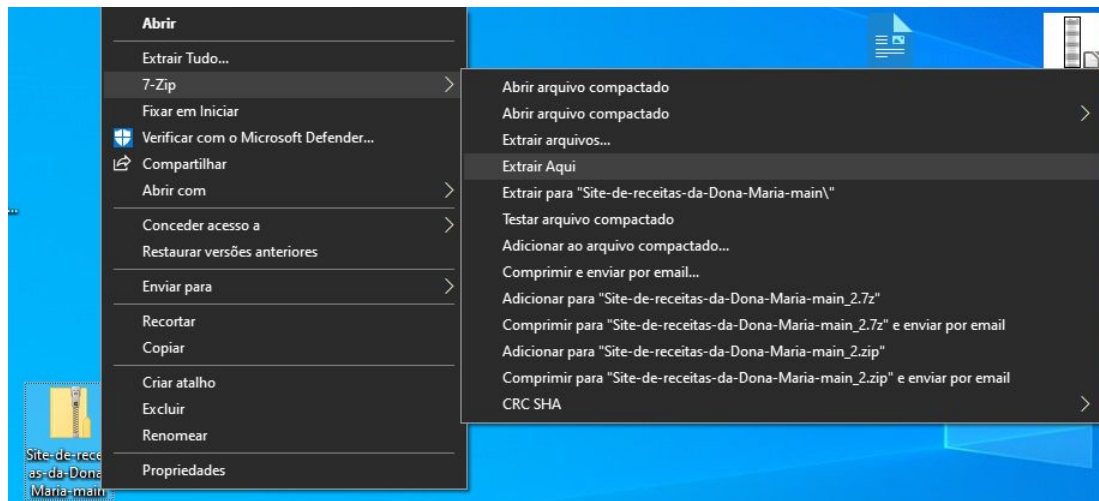
[illegible]

Clique em “Code” e, em seguida, em “Download ZIP”.



Uma pasta compactada será baixada no seu computador. Leve ela até a Área de Trabalho.

Clique com o botão direito do *mouse* sobre essa pasta e procure por alguma opção semelhante a “Extrair Aqui”. Clicando nessa opção, a pasta será descompactada.



Abrindo o VS Code e clicando em *File* e, depois, em *Open Folder*, você conseguirá editar os arquivos normalmente.

2. Continuando os trabalhos

2.1 Criando classes

Até agora, sempre que precisávamos alterar a aparência de alguma coisa, usávamos os seletores. Por exemplo, se quiséssemos alterar para branco a cor da fonte das listas não ordenadas de uma página, escreveríamos o seguinte no arquivo CSS:

```
ul {  
  color: white;  
}
```

2.1 Criando classes

O problema disso é que, em alguns casos, podemos ter mais de uma lista não ordenada e, por algum motivo, desejar mudar a aparência de apenas uma delas. Como resolver essa situação? Uma possibilidade é usando classes.

Nesta aula utilizaremos o atributo *class* da linguagem HTML. Com esse atributo, podemos, de certa forma, agrupar os elementos aos quais desejamos aplicar uma determinada mudança via CSS. No caso das listas não ordenadas, imagine que temos as listas LISTA1, LISTA2 e LISTA3 e queremos mudar a cor da fonte da LISTA1 e da LISTA2. Podemos criar uma classe usando *class* para a LISTA 1 e a LISTA2 e, no CSS, aplicar a mudança para essa classe. Dessa forma, a LISTA3 permanece inalterada.

2.2 class="receita-foto"

Na aula anterior já tínhamos atribuído à imagem do ovo frito a classe receita-foto.

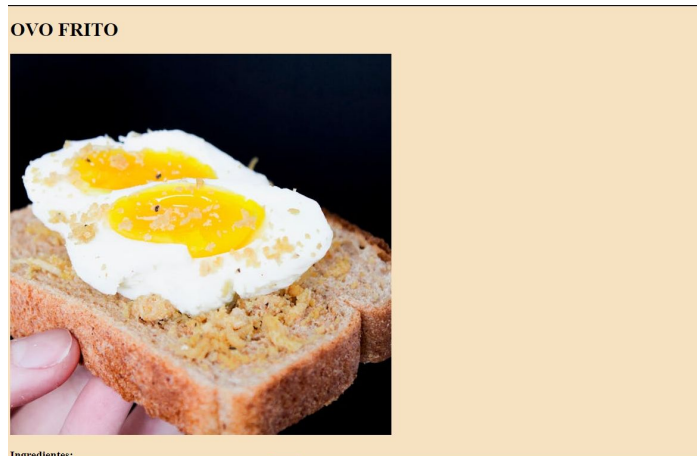
```
13 <h1>OVO FRITO</h1>
14 
```

Vamos fazer a primeira alteração na imagem utilizando CSS. Note que ao invés de utilizar o seletor *img* para mudar as características de todas as imagens da página, vamos usar *.receita-foto* (repare no ponto final antes do nome da classe) para fazer alterações só nas imagens que pertencerem a essa classe.

```
<> index_receitas.html # style_receitas.css 1 ●
# style_receitas.css > body
1 body {
2   background-color: #F6E2C1;
3 }
4
5 .receita-foto {
6
7 }
```

2.3 Propriedade *width*

Não queremos que a imagem ocupe quase a tela toda como está acontecendo.



Então vamos utilizar a propriedade *width* do CSS com o valor 20%, que vai fazer com que a imagem ocupe 20% da largura total da tela.

```
5  .receita-foto {  
6    width: 20%;  
7  }  
8
```

NÃO ESQUEÇA DE
SALVAR O ARQUIVO
DEPOIS DA ALTERAÇÃO!

2.3 Propriedade *width*

Aquela imagem gigantesca que tínhamos antes agora está bem menor e deixa espaço para que vejamos o texto abaixo dela:

OVO FRITO



Ingredientes:

- 1 ovo
- 1 colher (chá) de azeite (ou óleo)
- Sal e pimenta-do-reino moída na hora a gosto

Modo de preparo:

1. Numa tigela pequena quebre o ovo, com cuidado para não furar a gema.
2. Leve uma frigideira antiaderente ao fogo baixo e regue com 1 colher (chá) de azeite. Com cuidado, transfira o ovo para a frigideira e deixe cozinhar por cerca de 2 minutos – com uma espátula de silicone, vá puxando a borda da clara para perto da gema, para manter o formato arredondado do ovo.
3. Assim que a clara começar a firmar, tempere com sal e pimenta a gosto e tampe a frigideira para o ovo terminar de fritar mas a gema ainda ficar mole – o vapor permite que a clara cozinhe por igual sem a base do ovo queimar. Sirva a seguir.

2.4 Cabeçalho

A Dona Maria reclamou que não tem a identidade dela na página, então vamos criar um cabeçalho com a *tag* `<header>`.

A ideia é escrever “Dona Maria” com um título h1 e adicionar *links* para as páginas “Início”, “Sobre Nós”, “Receitas” e “Contato”. Além disso, vamos utilizar a *tag* `<p>` para colocar um parágrafo onde teremos o lema do site. Vamos criar esses elementos no arquivo HTML:

2.4 Cabeçalho

Note que colocamos os links como elementos de uma lista não ordenada. Além disso, eles estão apenas indicados, mas ainda não funcionam.

```
9  <body>
10  <header>
11    <h1>Dona Maria</h1>
12    <p>Açúcar faz mal, mas a gente gosta</p>
13    <ul>
14      <li>Início</li>
15      <li>Sobre Nós</li>
16      <li>Receitas</li>
17      <li>Contato</li>
18    </ul>
19  </header>
20  <!--Essa receita foi copiada de https://paneli
21    Aliás, chamamos o conteúdo dentro dessa ta
22    mas não são exibidos na página quando vist
23  <h1>OVO FRITO</h1>
```

DICA: na tag <p>, entre “gosta” e “</p>” adicione “🤠”. O que muda?

2.4 Cabeçalho

E, depois de salvar os arquivos e recarregar a página, temos essa bagunça:

Dona Maria

Açúcar faz mal, mas a gente gosta

- Início
- Sobre Nós
- Receitas
- Contato

OVO FRITO



Ingredientes:

- 1 ovo
- 1 colher (chá) de azeite (ou óleo)
- Sal e pimenta-do-reino moída na hora a gosto

Modo de preparo:

1. Numa frigideira aqueça o azeite e com uma espátula vire o ovo e deixe

2.4 Cabeçalho

Vamos deixar o título “OVO FRITO”, que estava entre `<h1></h1>` entre `<h2></h2>`. Dessa forma o tamanho desse título vai diminuir e haverá maior destaque no nome da Dona Maria.

```
22      mas não são exibidos na pá
23      <h2>OVO FRITO</h2>
24      <img class="receita-foto" src=
25      <h3>Ingredientes:</h3>
```

2.4 Cabeçalho

Vamos criar classes para os elementos do cabeçalho para que tenhamos mais facilidade de estilizá-los:

```
9    <body>
10      <header>
11        <h1 class="titulo-cabecalho">Dona Maria</h1>
12        <p class="lema-do-site">Açúcar faz mal, mas a gente gosta</p>
13        <ul class="links-do-topo">
14          <li class="links-do-topo-item">Início</li>
15          <li class="links-do-topo-item">Sobre Nós</li>
16          <li class="links-do-topo-item">Receitas</li>
17          <li class="links-do-topo-item">Contato</li>
18        </ul>
19      </header>
```

2.4 Cabeçalho

No arquivo CSS, vamos mudar a cor do fundo do cabeçalho utilizando o seletor *header*. Poderíamos atribuir uma classe para ele, mas acredito que não será necessário. Caso precise, depois atribuímos.

Além disso, por uma questão de organização, vou adicionar as mudanças do *header* acima das mudanças da classe *receita-foto*, pois o *header* fica mais acima na página. Esse tipo de escolha é arbitrária e visa uma melhor organização dos códigos.

2.4 Cabeçalho

```
1  body {  
2    background-color: #F6E2C1;  
3  }  
4  
5  header {  
6    background-color: #FFDA9E;  
7  }  
8  
9  .receita-foto {  
10   width: 20%;  
11 }
```

2.4 Cabeçalho

O resultado é que a área delimitada pela tag <header> está, agora em uma cor levemente diferente. Eu escolhi essa cor, mas você pode escolher outra se desejar.

Dona Maria

Açúcar faz mal, mas a gente gosta

- [Início](#)
- [Sobre Nós](#)
- [Receitas](#)
- [Contato](#)

OVO FRITO



Ingredientes:

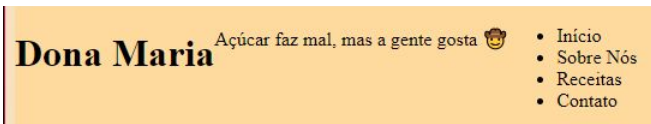
- 1 ovo
- 1 colher (chá) de azeite (ou óleo)
- Sal e pimenta-do-reino moída na hora a gosto

2.4 Cabeçalho

Os itens do cabeçalho um abaixo do outro estão deixando a Dona Maria irritada, então vamos melhorar a disposição deles. Vamos, primeiramente, utilizar a propriedade *display* com o valor *flex* para que os elementos sejam posicionados um ao lado do outro dentro do cabeçalho.

```
5  header {  
6      background-color: #FFDA9E;  
7      display: flex;  
8  }
```

2.4 Cabeçalho



Note que os links ainda estão um em cima do outro, pois o *display: flex* foi adicionado ao header, mas não à lista. Vamos adicionar *display:inline-block* à classe dos itens da lista dos links, que é a *links-do-topo-item*. O valor *inline-block* fará com que os itens fiquem todos em uma mesma linha, cada um ocupando apenas o espaço necessário para que seja exibido.

```
.links-do-topo-item {  
  display: inline-block;  
}
```

2.4 Cabeçalho

Início Sobre Nós Receitas Contato

A Dona Maria gostaria que o lema do site ficasse abaixo do nome dela, e não do lado. Então vamos fazer uma alteração no código: colocar o nome dela e o lema como dois itens de uma lista não ordenada:

```
9  <body>
10  <header>
11    <ul class="titulo-lema">
12      <li><h1 class="titulo-cabecalho">Dona Maria</h1></li>
13      <li><p class="lema-do-site">Açúcar faz mal, mas a gente gosta &#129312</p></li>
14    </ul>
15    <ul class="links-do-topo">
16      <li class="links-do-topo-item">Início</li>
17      <li class="links-do-topo-item">Sobre Nós</li>
18      <li class="links-do-topo-item">Receitas</li>
19      <li class="links-do-topo-item">Contato</li>
20    </ul>
21  </header>
```


2.4 Cabeçalho

Ficou assim:

- [Início](#) [Sobre Nós](#) [Receitas](#) [Contato](#)
- **Dona Maria**
- Açúcar faz mal, mas a gente gosta 🍪

Para que os links e o título deixem de estar tão próximos, usaremos a propriedade *justify-content* com o valor *space-around*, o que vai fazer com que os elementos que estão numa mesma linha tenham margens iguais. Lembre que, neste caso existem dois elementos dentro da tag `<header>`: uma lista com o título e o lema do site e uma outra lista com os links.

2.4 Cabeçalho

```
1  body {  
2    background-color: #F6E2C1;  
3  }  
4  
5  header {  
6    background-color: #FFDA9E;  
7    display: flex;  
8    justify-content: space-around;  
9  }  
10  
11  .links-do-topo-item {  
12    display: inline-block;  
13  }  
14  
15  .receita-foto {  
16    width: 20%;  
17  }  
18
```

2.4 Cabeçalho

Ficou assim:

. **Dona Maria**

- Açúcar faz mal, mas a gente gosta 🍬

[Início](#) [Sobre Nós](#) [Receitas](#) [Contato](#)

A Dona Maria disse que duvida que vamos conseguir tirar aqueles marcadores (“bolinhas”) do lado do nome do site e do lema. Esses marcadores aparecem porque colocamos o título e o lema como elementos de uma lista não ordenada. Vamos eliminá-los com a linha *list-style-type: none* no código CSS:

2.4 Cabeçalho

Não esqueça que a classe da lista que contém o título e o lema é a *titulo-lema*.

```
1  body {
2    background-color: #F6E2C1;
3  }
4
5  header {
6    background-color: #FFDA9E;
7    display: flex;
8    justify-content: space-around;
9  }
10
11  .titulo-lema {
12    list-style-type: none;
13  }
14
15  .links-do-topo-item {
16    display: inline-block;
17  }
18
19  .receita-foto {
20    width: 20%;
21  }
```

RESULTADO: Sem marcadores



2.5 Margin e padding

A propriedade *margin* no CSS refere-se ao espaçamento externo. Por exemplo, imagine uma imagem ao lado esquerdo da página. Quando maior o valor inserido para *margin*, mais longe essa imagem ficará da borda esquerda da página.

Já a propriedade *padding* se refere ao espaçamento interno. Por exemplo, imagine um livro. Quanto maior o valor atribuído ao *padding*, maior a distância entre uma linha e outra.

Difícil de entender? Com certeza, eu também tive dificuldades com essas propriedades, e parece que quem entende não consegue explicar direito. Vamos usar ambas na nossa página para tentar ver como funcionam.

2.5 Margin e padding

No documento CSS, vamos utilizar o seletor `*` para fazer uma alteração que vai servir para todos os elementos da página. Mais tarde vamos personalizando o *padding* e o *margin* de todos os elementos individualmente.

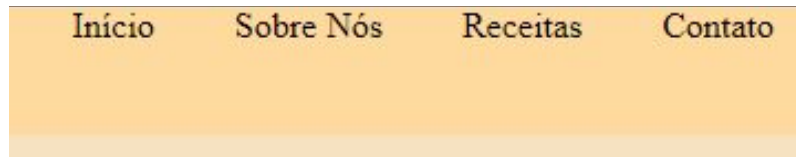
```
1  * {  
2      margin: 0;  
3      padding: 0;  
4  }  
5  
6  body {  
7      background-color: #F6E2C1;  
8  }  
9  
10 header {  
11     background-color: #FFDA9E;  
12     display: flex;  
13     justify-content: space-around;  
14 }  
15  
16 .titulo-lema {  
17     list-style-type: none;
```

2.5 Margin e padding

Como a Dona Maria não aguentava mais ver as marcações dos links coladas, vamos utilizar margin na classe links-do-topo-item. Utilizaremos os valores 0 16px. Dessa forma a distância entre os itens será 0 para cima e baixo e 16px para esquerda e direita.

```
19
20  .links-do-topo-item {
21      display: inline-block;
22      margin: 0 16px;
23  }
```

Resultado:



2.5 Margin e padding

Vamos editar a propriedade *padding* na tag `<header>` para adicionar um espaço interno verticalmente e, desta forma, melhorar essa sensação de “aperto” que ficou. 12px será o padding acima e abaixo e 0px será à esquerda e à direita.

```
10 header {  
11     background-color: #FFDA9E;  
12     display: flex;  
13     justify-content: space-around;  
14     padding: 12px 0px;  
15 }
```

Resultado:

Dona Maria

Açúcar faz mal, mas a gente gosta 🍪

[Início](#)

[Sobre Nós](#)

[Receitas](#)

[Contato](#)

3. Concluindo

3.1 Difícil?

Talvez o posicionamento dos itens seja uma das partes mais difíceis (ou, pelo menos, “chatas”) do desenvolvimento *web*. Demora para que entendamos o que cada propriedade faz.

Às vezes pensamos que entendemos tudo, mas quando vamos colocar em prática, percebemos que estávamos errados mais uma vez.

A forma de dispor os itens não é tão simples também. Por exemplo, o uso de listas às vezes será o mais indicado, em outras vezes utilizar tabelas será melhor. Tudo depende muito da situação e do gosto pessoal.

Por isso, sintam-se livres para testar outras formas de chegar no resultado que chegamos até aqui.

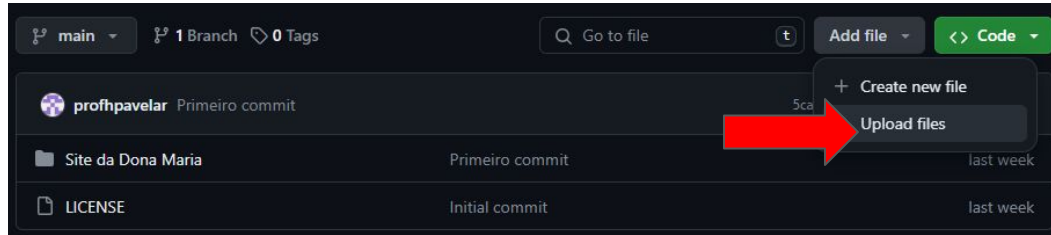
3.2 GitHub

Chegou a hora de atualizar o seu repositório no GitHub. Lembrando que estou atualizando o meu também com as alterações de cada aula, então, caso você não tenha conseguido fazer tudo, pode deixar para terminar em casa.

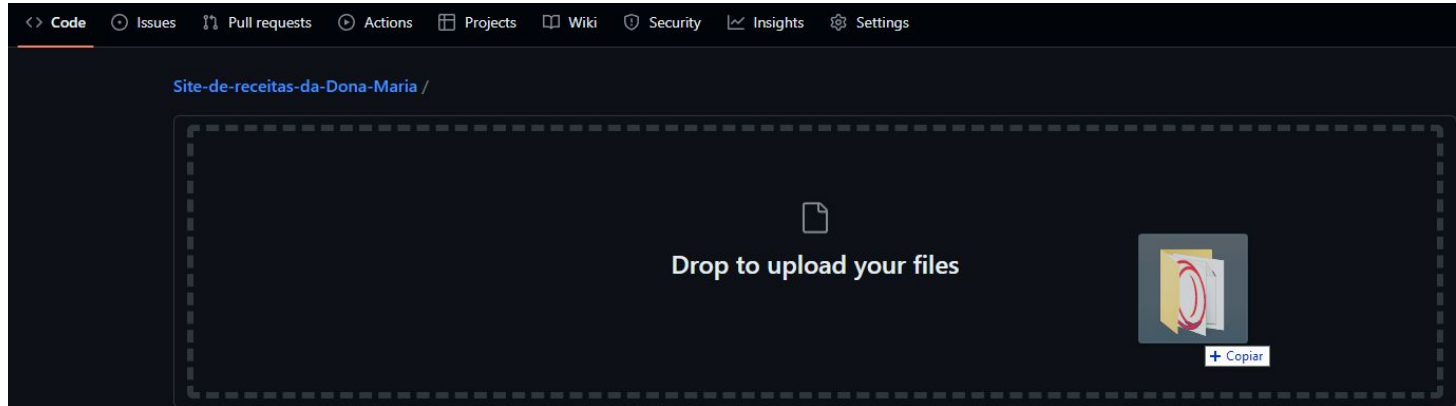
Caso não consiga fazer login no GitHub, você pode enviar seus arquivos da aula de hoje para o email avelar.henrique@escola.pr.gov.br.

3.2 GitHub

No seu repositório, clique em “Add File” e, depois, em “Upload Files”.



Depois, “arraste” a pasta com os arquivos para o espaço indicado



3.2 GitHub

Depois, adicione um título e uma descrição para o *commit* e clique no botão verde para enviar as alterações.

