

Aula 3

Finalizando o cabeçalho

Curso

**HTML e CSS: usando a
criatividade para criar o seu site**

O que vamos aprender?



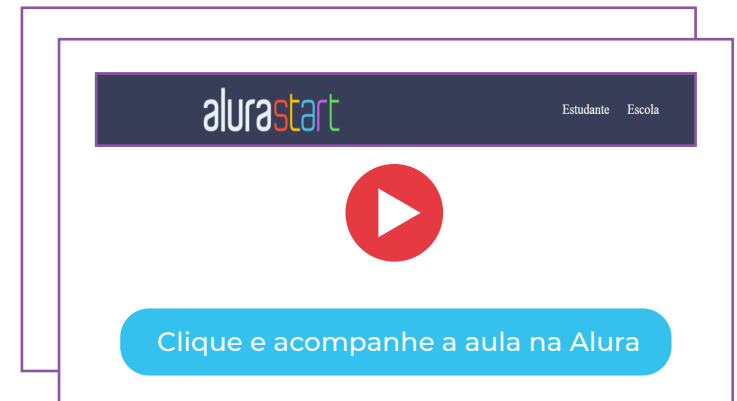
Conhecer espaçamentos internos e externos de uma página.



Entender e aplicar o seletor*.



Finalizar o cabeçalho da página.

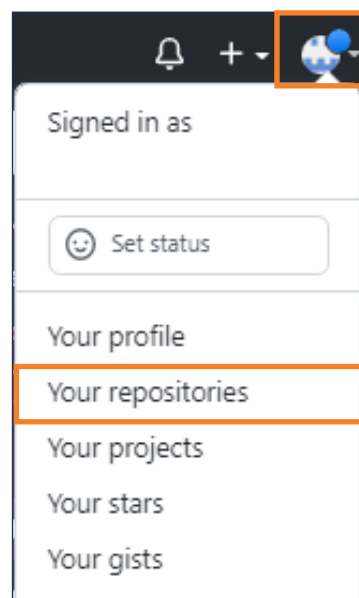


Finalizando o header

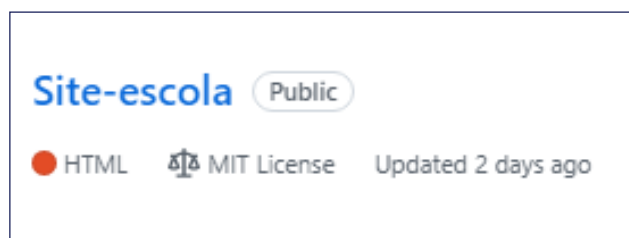
O objetivo deste curso é tornar o protótipo feito no Figma, ou em outra ferramenta, uma página HTML. Para dar continuidade ao nosso projeto, devemos acessar os arquivos criados na aula anterior, por meio do VSCode. Caso tenha mudado de computador ou não possua mais os arquivos, não se preocupe. Podemos baixá-los do GitHub, seguindo os passos abaixo:

Passo 1: Vá até o site github.com.

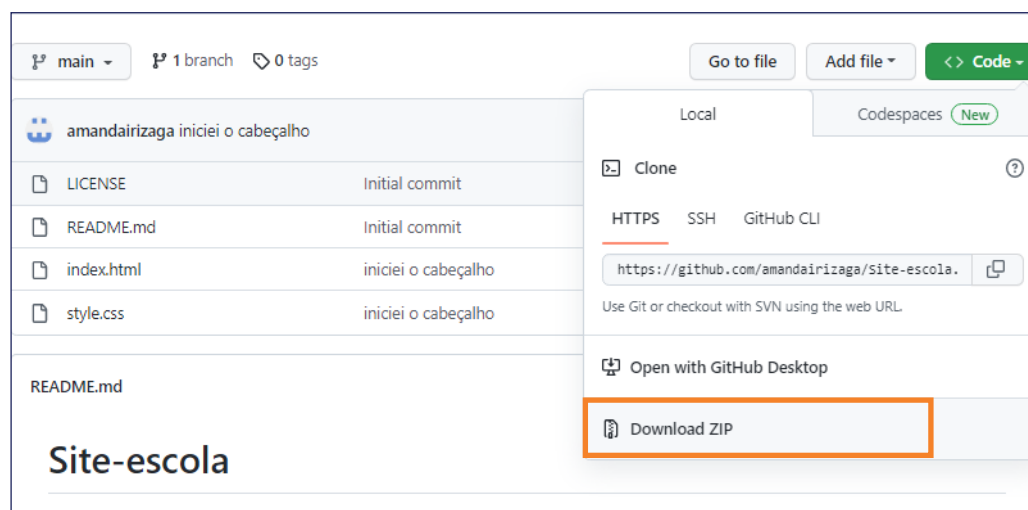
Passo 2: Vá até o canto superior direito da tela e clique em *Your repositories*, que significa “seus repositórios”.



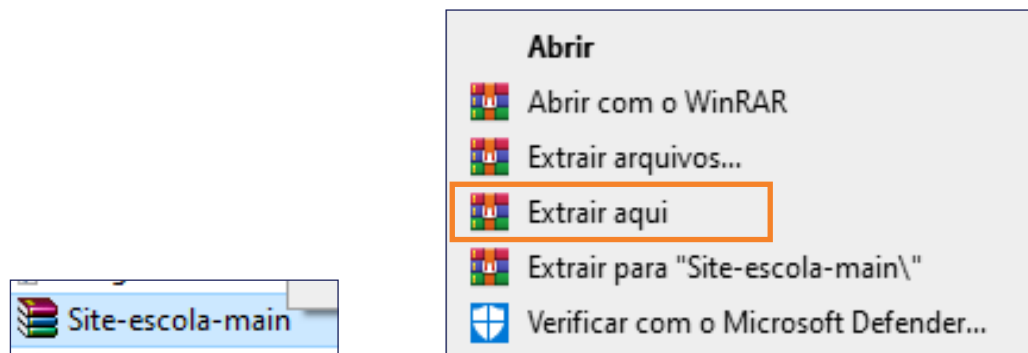
Passo 3: Uma lista com todos os seus repositórios será exibida; clique no repositório criado na aula anterior.



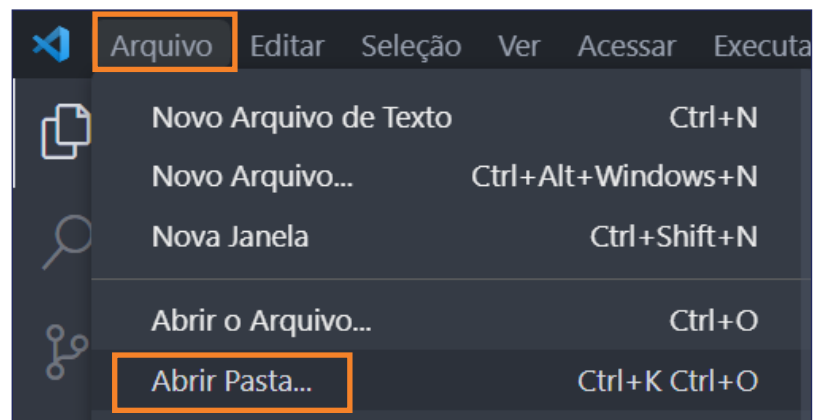
E então, baixe os arquivos deste repositório clicando em *Code* → *Download ZIP*.



Após baixar a pasta compactada, devemos descompactá-la para extrair os arquivos. Clique com o botão direito do mouse na pasta e então em *Extrair arquivos...*



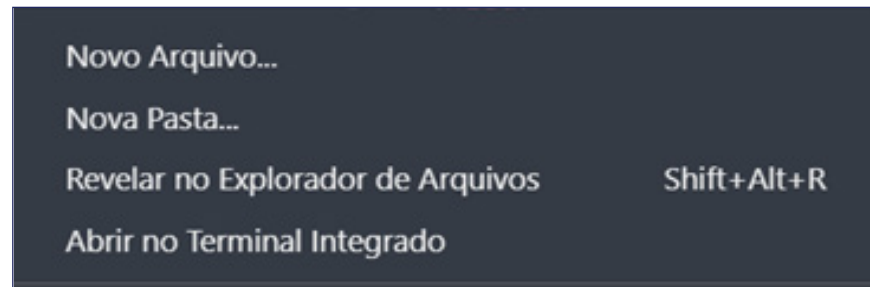
Agora que já possuímos em uma nova pasta os arquivos criados anteriormente, devemos ir ao VSCode e abrir a pasta clicando em *Arquivo* → *Abrir Pasta...*



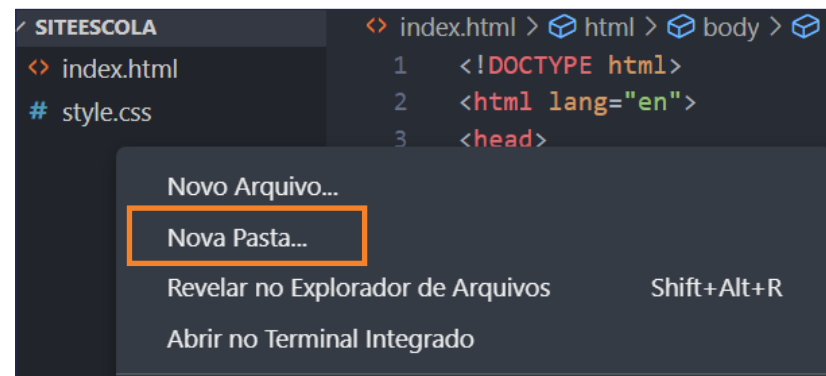
Feitos os passos anteriores, vamos iniciar a programação. Nesta aula, iremos adicionar algumas imagens ao site. Para manter nossa pasta organizada, vamos criar uma outra pasta que irá armazenar todas as imagens do nosso site. Para isso, vá até a área do projeto, à esquerda da tela:



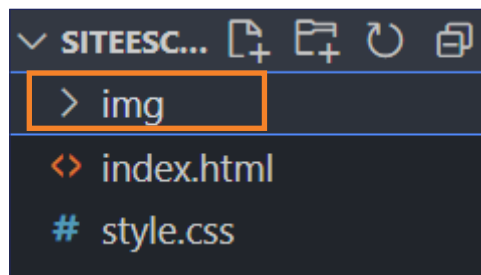
Clique com o botão direito logo abaixo do arquivo style.css. Deve abrir uma janela tal como a da imagem a seguir.



Clique em *Nova Pasta*.



E nomeie como *img*.



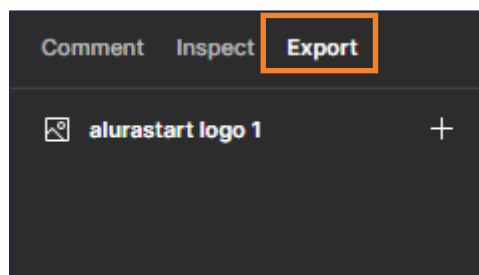
A página que está sendo criada fala sobre uma escola. O site está dividido em seções. A primeira seção é *Escola*, que tem uma pequena descrição sobre a escola e algumas imagens. Poderemos usar nossas próprias imagens para criar as seções ou as mesmas imagens do protótipo utilizado como exemplo. Para buscar por imagens podemos acessar os sites Pixabay e Freepik. É importante ter atenção ao utilizar imagens que possuem direitos autorais. Procure sempre colocar os créditos das imagens.



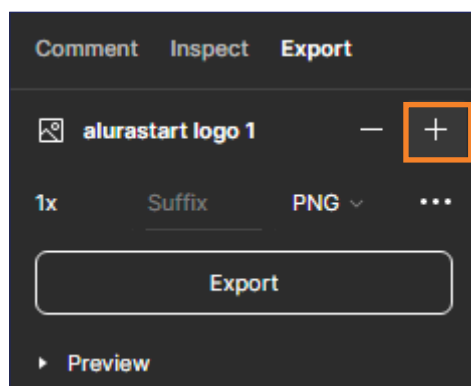
Neste material, optamos por utilizar as mesmas imagens do protótipo. Vamos utilizar a imagem em que está escrito “alurastart”. Para importar as imagens do Figma, clique duas vezes em cima da imagem:



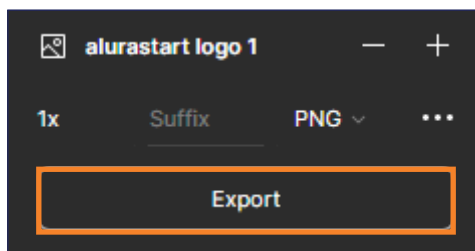
E então, vá até a coluna à direita da tela, na aba *Export*.



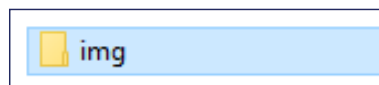
Clique no símbolo de sinal positivo “+”.



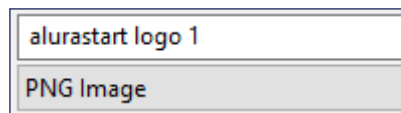
Clique na opção *Export*, que significa exportar.



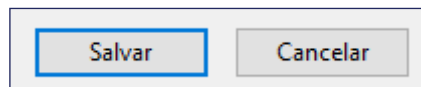
Procure pela pasta *img* dentro da pasta do projeto *SiteEscola*.



Uma vez encontrada, altere o nome da imagem se desejar.



E, então, clique em salvar.



Repita este processo para todas as imagens do Figma. Caso as imagens forem de sites, basta salvar a imagem diretamente dentro da pasta quando encontrar uma de que goste.

Vamos adicionar a primeira imagem ao cabeçalho do site. Para isso, devemos acessar o código do arquivo `index.html`. Usaremos a tag ``, que permite adicionarmos imagens em uma página HTML. Então, logo abaixo da tag `<header>`, acrescente o seguinte código:

```
<header>
  <img src="" alt="">
```

A propriedade **src** define qual imagem será adicionada. A propriedade **alt** é um recurso utilizado para descrever a imagem. A descrição da imagem é importante para pessoas com deficiência visual, pois possibilita que se compreenda a imagem por meio de sua descrição.

Dito isto, acrescentaremos o seguinte código que é o caminho da imagem, ou seja, a indicação para o navegador de onde ela está.

```

```

O caminho para encontrar a imagem deve iniciar com `.` e `/` pois há uma subpasta no projeto. Essa subpasta chama-se *img* e armazena as imagens.

Ao carregarmos a página no navegador, podemos observar que a imagem foi adicionada.



Perceba que o tamanho da imagem está inadequado, a imagem está muito junta às laterais da página. Para resolver essa questão vamos estilizar o tamanho dessa imagem.

Ao manipularmos elementos usando CSS, uma boa prática de programação é a criação de **classes**. Uma classe é uma estrutura que armazena um conjunto de características visuais de um elemento HTML. Então, vamos criar uma classe que estabeleça que o tamanho definido será adicionado somente à imagem do cabeçalho. Essa classe será chamada de **cabecalho-imagem**.

```
** vamos adicionar a classe junto com o seletor **.** (símbolo de ponto-final).

Ao criarmos uma classe no arquivo HTML, precisamos utilizar o seletor **.** (ponto-final) antes do nome da classe. Ao estilizarmos as tags já existentes no HTML, não é necessário adicionar o seletor.

```
.cabecalho-imagem{
}
```

Acrescente o valor 20% à propriedade **width**, que significa largura. Isso representará a ocupação de 20% da tela pela imagem adicionada.

```
.cabecalho-imagem{

 width: 20%;

}
```

Ao carregarmos a página, podemos observar uma mudança sutil no tamanho da imagem.



Retornando ao arquivo index.html, podemos excluir o texto “vai ser uma imagem aqui”, afinal, já adicionamos a imagem correta no lugar. Os demais elementos do cabeçalho também devem ser estilizados. Primeiramente, vamos criar mais duas classes: **cabecalho** e **cabecalho-lista-item**. Estas serão adicionadas às tags **<header>** e **<ul>** respectivamente.

```
<header class="cabecalho">

 <ul class="cabecalho-lista-item">
 Estudante
 Escola

```

Nenhuma modificação visual será implementada na página pois ainda não foram criadas estilizações no CSS. Para isso, vamos até o arquivo style.css. A primeira modificação na estrutura será a alteração da tag **<header>** pela classe **cabecalho**, desta forma fica fácil identificar a parte da página que estamos estilizando.

```
.cabecalho {
background-color: #383D58;
color: white;
}
```

Logo abaixo da propriedade **color**, adicione a propriedade **display** e o valor **flex** à classe **cabecalho**.

```
color: white;
display: flex;
}
```



Ao carregarmos a página, podemos observar que alguns elementos estão diferentes. Vamos analisar:

Antes:



Depois:



Ao adicionar a propriedade **display**, a lista de itens *Estudante* e *Escola* foi posicionada ao lado da imagem. Ou seja, essa propriedade organiza os elementos lado a lado e só deve trocar de linha quando o espaço ocupado pela linha atual terminar.

A próxima propriedade que devemos adicionar chama-se **justify-content**, que significa justificar o conteúdo. Existem muitos valores possíveis de serem adicionados a essa propriedade. Neste momento, usaremos o valor **space-around**.

```
display: flex;
justify-content: space-around;
}
```

Ao carregarmos a página, podemos observar que um novo formato de organização dos elementos foi exibido.



O que esse valor **space-around** faz? Ele determina margens iguais para os elementos de uma mesma linha criando um espaçamento entre eles, por isso o nome **space-around**, que significa, “espaço ao redor”.

O cabeçalho agora está um pouco mais parecido com o protótipo, mas há o que alterar. Vamos reorganizar os elementos da lista para que eles sejam distribuídos um ao lado do outro. Então, em index.html, adicione uma classe chamada **cabecalho-lista-item** aos elementos **<li>**.

```
<li class="cabecalho-lista-item">Estudante
<li class="cabecalho-lista-item">Escola
```

Crie a classe no arquivo CSS, adicionando a propriedade **display** e o valor **inline-block** à classe.

```
.cabecalho-lista-item{
 display: inline-block;
}
```

Ao carregarmos a página, o cabeçalho está assim:



A propriedade **display** junto do valor **inline-block** define que o elemento deve ocupar apenas o espaço suficiente para que ele seja exibido, e que, portanto, os próximos elementos em sequência devem ser adicionados logo após o último.

Os elementos do cabeçalho já possuem um posicionamento adequado, agora, é necessário adicionar os espaçamentos para que a visualização da página se torne mais agradável.

# Espaçamentos da página

No que se refere aos espaçamentos de uma página, eles se dividem em: internos e externos.

O espaçamento interno, também chamado de **padding**, cria um espaço considerando as bordas do próprio elemento, por exemplo, imagine um trecho de um livro. Quanto maior o espaçamento interno, maior será o espaço entre as linhas desse trecho.

O espaçamento externo, também chamado de **margin**, cria um espaço considerando as bordas da tela e as bordas externas dos demais elementos, por exemplo, imagine uma imagem de um site adicionada ao lado esquerdo da tela. Quanto maior o espaçamento externo, maior será a distância dessa imagem da borda à esquerda da página.

Vamos começar estilizando o espaçamento externo. Devemos adicionar à classe **classe-lista-item** a propriedade **margin**. Os valores podem ser adicionados de diferentes formas, vamos acrescentar **0** e **16px**, que correspondem a nenhum espaçamento nas margens de cima e de baixo (0) e 16 pixels de espaçamento na direita e na esquerda desse elemento (**16px**).

```
.cabecalho-lista-item{
 display: inline-block;
 margin: 0 16px;
}
```

Ao carregarmos a página, podemos observar que as margens foram aplicadas.



O cabeçalho ainda possui alguns espaçamentos indesejados; para alterarmos isto, vamos acrescentar o seletor **\*** no início do arquivo style.css. O seletor **\*** define para todo documento HTML as propriedades que forem adicionadas a ele. Portanto, vamos adicionar o valor **0** às propriedades **margin** e **padding**.

```
* {
 margin: 0;
 padding: 0;
}
```

Ao carregarmos a página, podemos observar que a disposição dos elementos foi alterada.

Antes:



Depois:



Altere o espaçamento interno para cima e para baixo adicionando à propriedade **padding** o valor **24px** para cada elemento da lista. Usaremos a classe **cabecalho-lista-item**.

```
margin: 0 16px;
padding: 24px 0;
}
```

Altere também o espaçamento interno do cabeçalho geral, através da classe **cabecalho**.

```
justify-content: space-around;
padding: 24px 0;
}
```

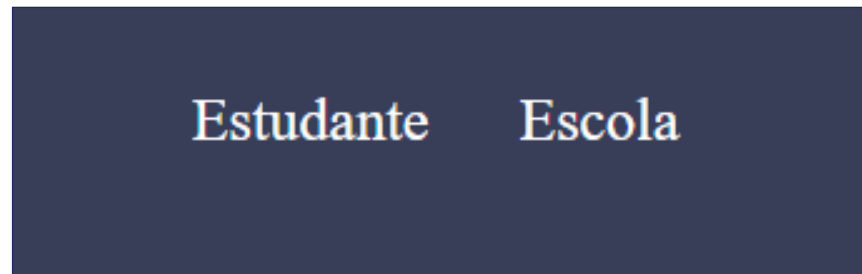
Ao carregarmos a página, podemos observar que os espaçamentos foram adicionados.



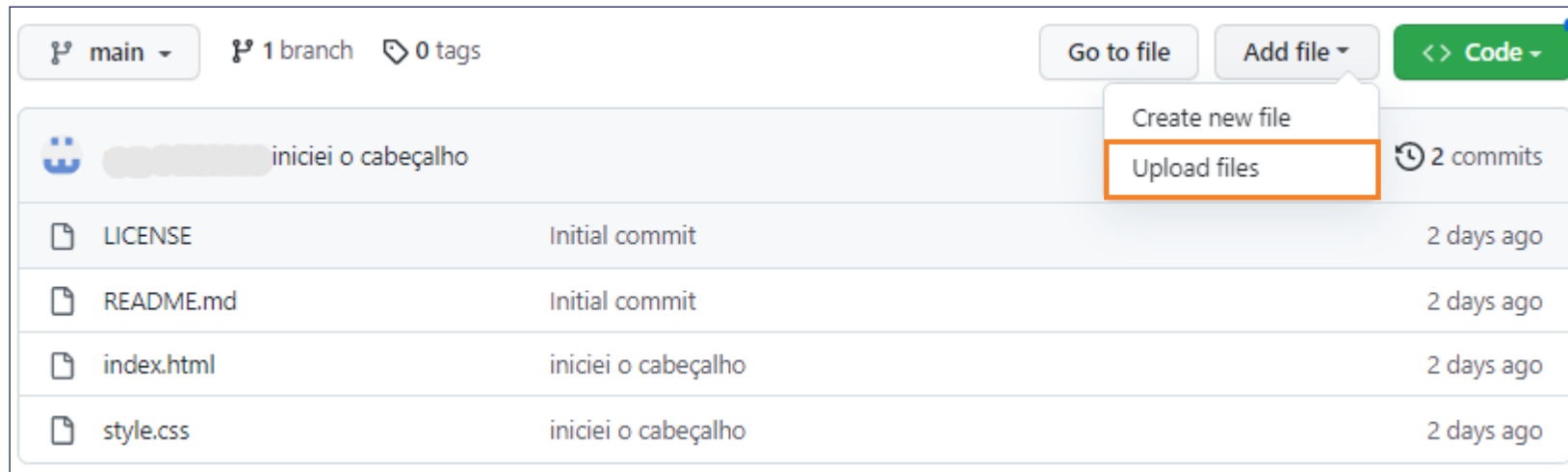
O tamanho da letra dos elementos da lista está pequeno. Então, vamos adicionar à propriedade **font-size**, que significa tamanho de fonte, o valor **24px**. Essa alteração deve ser feita na classe **cabecalho-lista-item**.

```
padding: 24px 0;
font-size: 24px;
}
```

Ao carregarmos a página, podemos observar que o tamanho da fonte foi alterado.

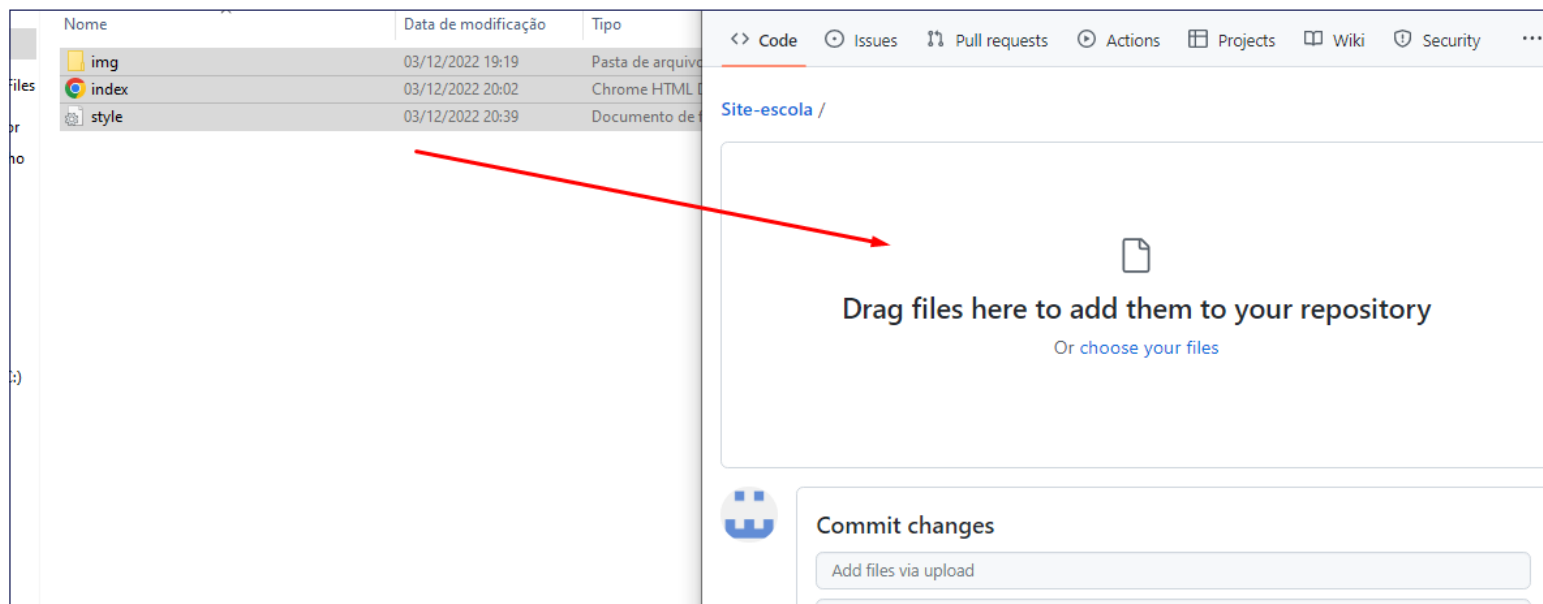


Finalizado o cabeçalho, devemos incorporar as alterações no repositório GitHub. Vamos abrir o GitHub e clicar em *Add file* → *Upload files*. O termo *upload* significa “carregar arquivos”.

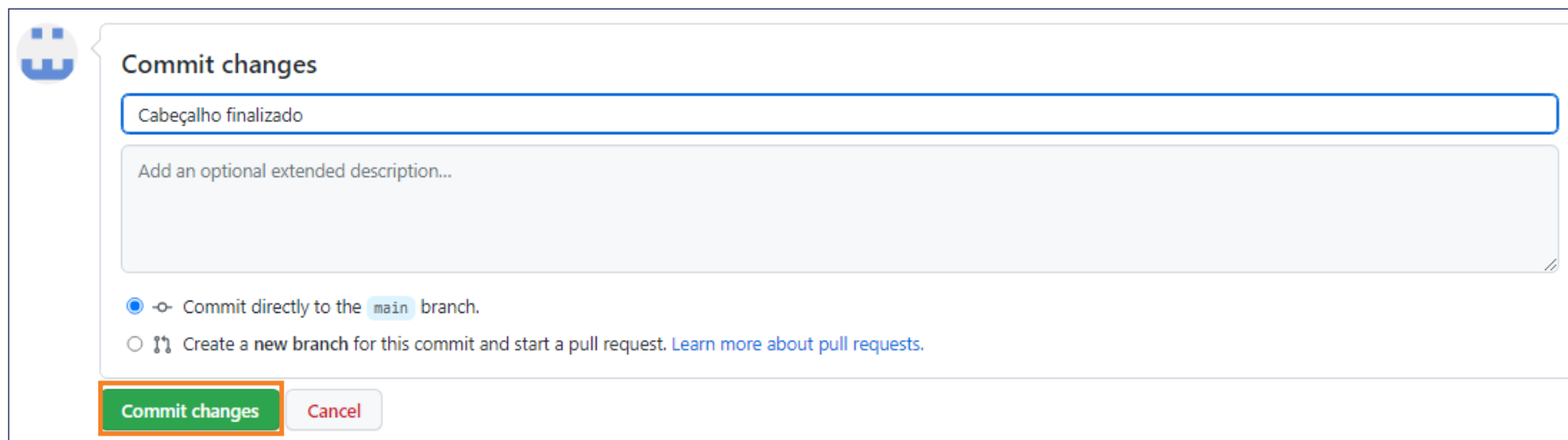




Então, abra a pasta com os arquivos, selecione e arraste para o espaço indicado pela seta na imagem a seguir. Inclua a pasta *img* na área de upload.



Após carregar os arquivos, sinalize a alteração feita e comente “Cabeçalho finalizado”; então, clique no botão *Commit changes* para salvar essas alterações no repositório.



The screenshot shows the GitHub 'Commit changes' dialog box. It features the GitHub logo in the top left corner. The title 'Commit changes' is at the top. Below it is a text input field containing 'Cabeçalho finalizado'. Underneath is a larger text area with the placeholder 'Add an optional extended description...'. At the bottom, there are two radio button options: the first is selected and labeled 'Commit directly to the main branch.', and the second is labeled 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the very bottom are two buttons: 'Commit changes' (highlighted with an orange border) and 'Cancel'.

Todas as atualizações produzidas já estão salvas no repositório.

# Desafio

Durante esta aula, você aprendeu dois tipos de seletores. O seletor `.` (ponto-final), que é utilizado quando uma nova classe é criada, e o seletor `*`, que é utilizado para definir um conjunto de propriedades para todos os elementos de uma página automaticamente. Como desafio, você deverá realizar uma pesquisa sobre os seletores do CSS e verificar se há mais tipos e quais são suas funções.