

## AULA 12 – GIT BRANCHES DE FORMA FÁCIL E COM EXEMPLO

→ Branch significa ramo

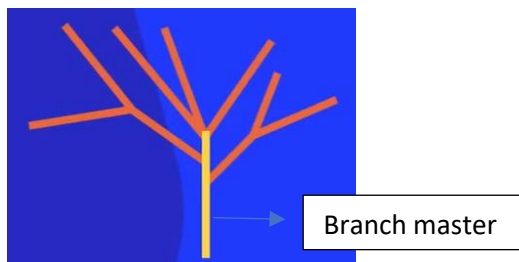
→ Temos o ramo principal, que se chama **branch master**. Esse é o ramo de produção, é o produto final, a versão corrente

→ Se meu software está na versão 4.0, por exemplo, é essa versão que é meu branch master

→ Ao mesmo tempo que meu software está na versão 4.0, minha equipe já pode estar desenvolvendo a versão 4.5 por exemplo, no caso ela seria uma nova branch, que posteriormente possa virar a branch master

→ Outro exemplo: meu software se desenvolveu e se descobriu um problema nele: posso criar uma branch só para resolver esse problema, fazendo um **merge**

Representação:



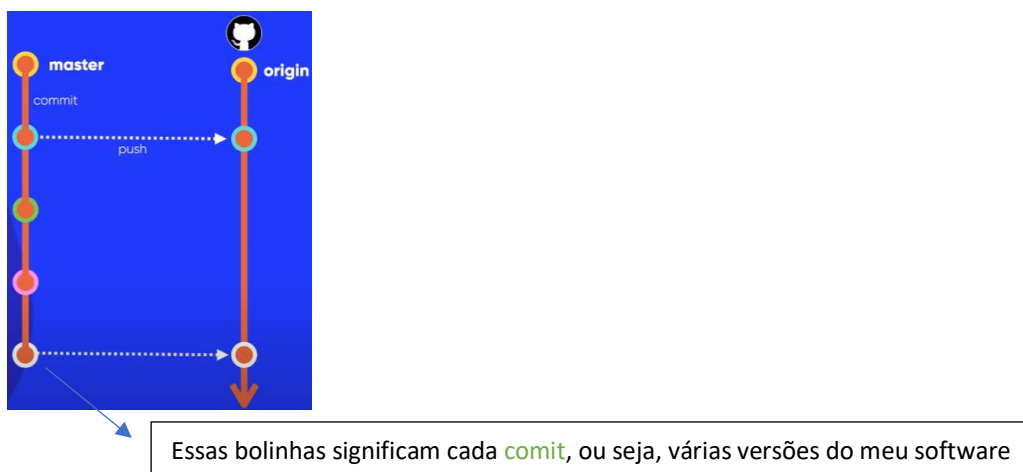
→ Sempre que eu crio um repositório novo, esse ramo master ele é obrigatório. Sempre quando escrevemos algo novo no VS Code ou alteramos ele, devemos fazer um **commit** essa alteração

→ Quando eu comito, eu gero uma nova versão

→ A qualquer momento, eu posso jogar esse conteúdo da **branch master** para a minha **origin**. Origem é tirar do meu **repositório local** e colocar no **repositório remoto** (ou seja, do Git para o GitHub)

→ Essa origin é basicamente um backup, estou colocando meu software de maneira remota

→ Ou seja, quando eu **comito**, estou salvando na minha máquina local. Quando dou um **push**, eu jogo esse software diretamente para o meu **repositório local**, criando uma cópia dele

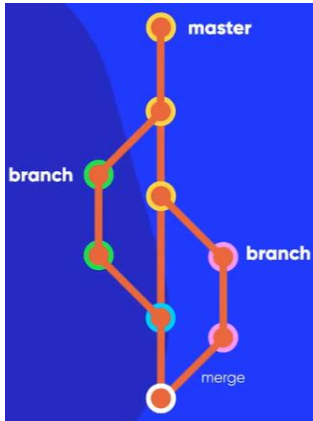


**ADENDO: evitar comitar tudo na branch master!**

→ Quando estou fazendo uma **feature** nova, eu não vou ficar desenvolvendo e comitando na master

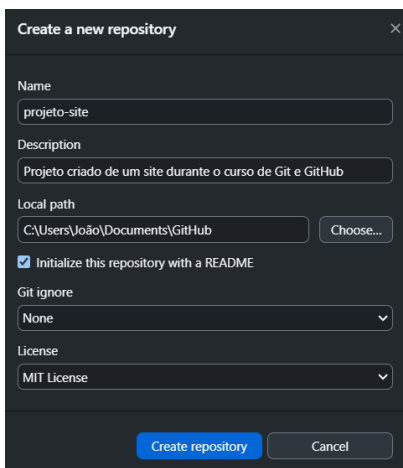
→ Se eu fizer uma cagada gigante, por exemplo, estou fazendo ela no meu ramo principal!

- Para grandes projetos, o ideal é criar uma ramificação, ou seja, uma **nova branch**
- No momento que eu finalizo essa nova branch, eu faço um **merge**, que ele joga essa feature que eu desenvolvi e joga no meu ramo principal
- Posso desenvolver, ao mesmo tempo, várias branches... Exemplo: uma equipe trabalhando no design, outra equipe trabalhando no conteúdo
- Se acontecer algum tipo de conflito, o próprio Git avisa!

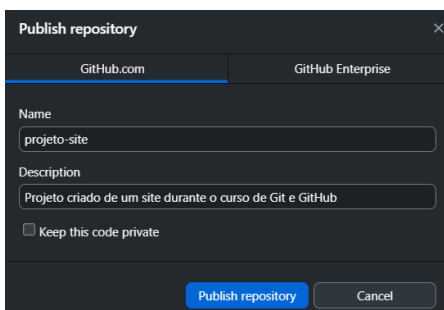


### Exemplo prático:

- Primeiramente, criando um novo repositório no **GitHub Desktop**

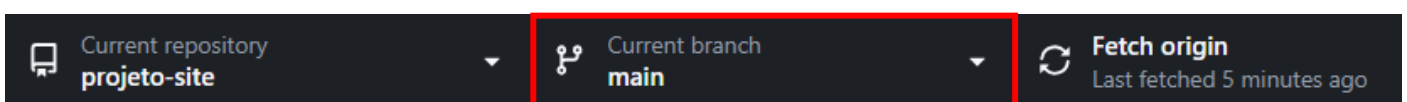


- Publicar esse repositório



- Abrir o projeto no VS Code

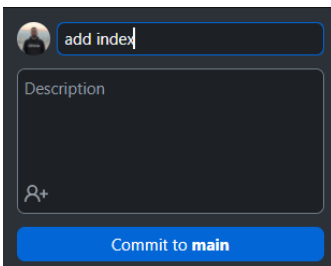
OBS: no meu GitHub Desktop, eu consigo ver a minha branch atual



Base do VS Code:

```
index.html > html > head
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Título do site</title>
7 </head>
8 <body>
9   <main>
10    <header>
11      <h1>Título do site</h1>
12    </header>
13    <article>
14      <h2>Artigo 001</h2>
15      <p>Texto do Artigo 001</p>
16    </article>
17    <article>
18      <h2>Artigo 002</h2>
19      <p>Texto do Artigo 002</p>
20    </article>
21  </main>
22 </body>
23 </html>
```

→ Com isso, vou comitar para minha branch master no meu **GitHub Desktop**

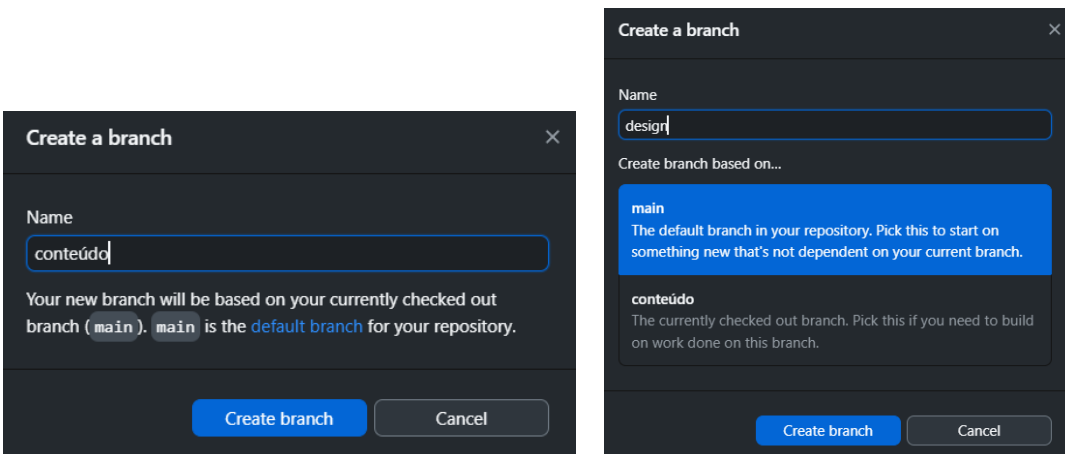


Main é a mesma coisa que master!

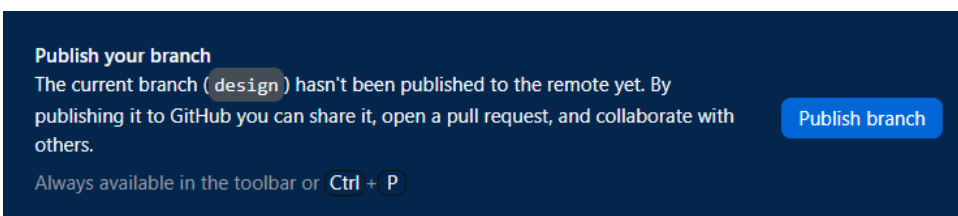
→ Fazer o **push**

→ Com tudo isso feito, eu vou criar duas branches: uma para a equipe que estiver cuidando do conteúdo (HTML) e outra equipe para o design (CSS)

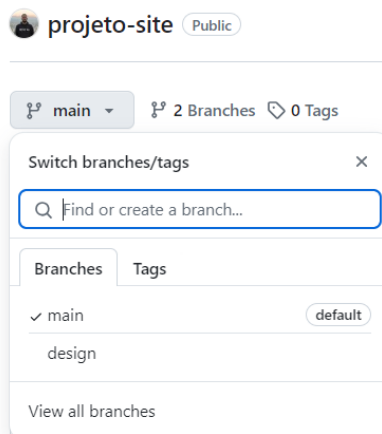
→ Para isso, no GitHub Desktop, é só ir em **branch** → **new branch**



→ Com isso, o próprio **GitHub Desktop** já sugere que eu publique essas branches



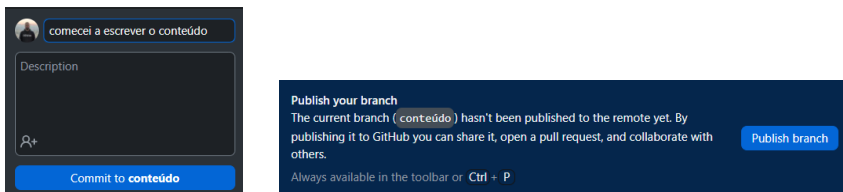
→ Com elas publicadas, podemos visualizar no meu GitHub as branches



→ Para começar a trabalhar essa nova branch, no meu GitHub Desktop, eu devo selecionar a branch desejada e abrir ela no meu VS Code

```
index.html > html > body > main > article > p
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Título do site</title>
7  </head>
8  <body>
9    <main>
10     <header>
11       <h1>Cursos Grátis</h1>
12     </header>
13     <article>
14       <h2>Curso de HTML5 com CSS3</h2>
15       <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Vero illum, rem nulla dolore
16         nobis excepturi aut assumenda in suscipit mollitia placeat possimus praesentium,
17         repudiandae voluptates earum dicta, optio quos corporis!</p>
18     </article>
19     <article>
20       <h2>Curso de JavaScript</h2>
21       <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nam voluptatibus culpa
22         inventore, modi provident sed voluptas explicabo et voluptatum id dolorum eligendi fugiat
23         eaque omnis, autem pariatur saepe reiciendis maxime.</p>
24     </article>
25   </main>
26 </body>
27 </html>
```

→ Com isso, eu **comito** para a minha **branch de conteúdo**, e logo após eu dou um **push**

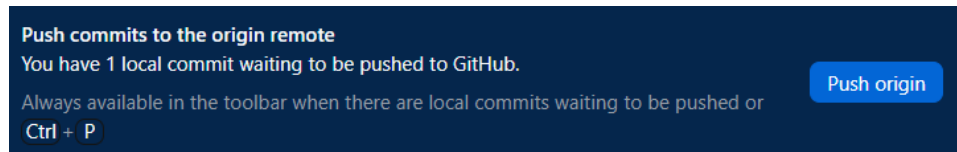
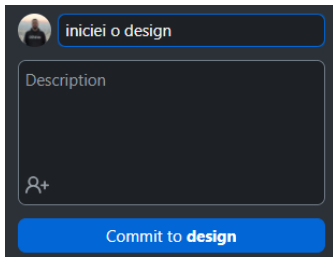


→ Agora, eu vou mexer na minha **branch de design**

```
index.html > html
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Título do site</title>
7    <link rel="stylesheet" href="estilos/style.css">
8  </head>
9  <body>
10   <main>
11     <header>
12       <h1>Título do site</h1>
13     </header>
14     <article>
15       <h2>Artigo 001</h2>
16       <p>Texto do Artigo 001</p>
17     </article>
18     <article>
19       <h2>Artigo 002</h2>
20       <p>Texto do Artigo 002</p>
21     </article>
22   </main>
23 </body>
24 </html>
```

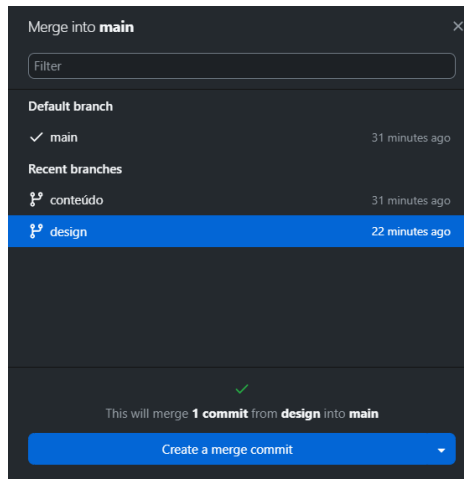
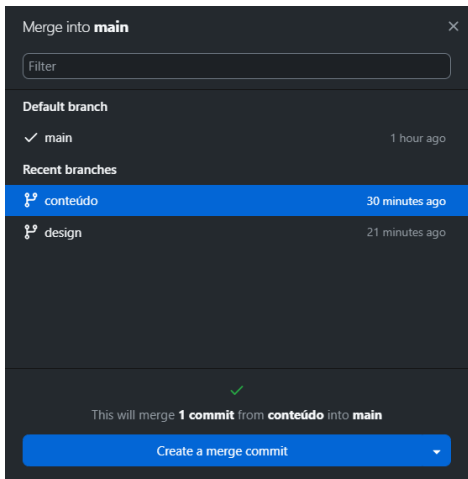
```
estilos > style.css > main
1  * {
2    font-family: Arial, Helvetica, sans-serif;
3    font-size: 16px;
4  }
5
6  body {
7    background-color: #burlywood;
8  }
9
10 main {
11   background-color: #white;
12   width: 500px;
13   margin: auto;
14   padding: 10px;
15   box-shadow: 3px 3px 5px #rgba(0, 0, 0, 0.448);
16 }
```

→ **Comito** e dou um **push**



→ Para fazer um **merge**, primeiro eu devo estar com a **branch master selecionada**, logo após eu vou em **branch** → **merge into current branch**

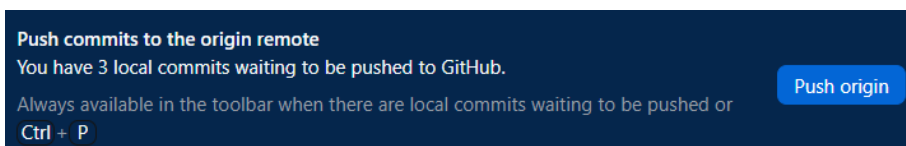
→ Devo fazer isso tanto para a **branch de conteúdo** como para a **branch de design**



OBS: se por acaso eu alterei uma mesma linha nas minhas duas ramificações ao mesmo tempo, pode gerar um **conflito!**

O próprio **GitHub Desktop** manda abrir no VS Code e já mostra o erro para corrigir

→ Após o **merge**, o **GitHub Desktop** já sugere que eu faça um **push**



→ Com isso, está feito o projeto!

