

Tutorial de GitHub

Sumário:

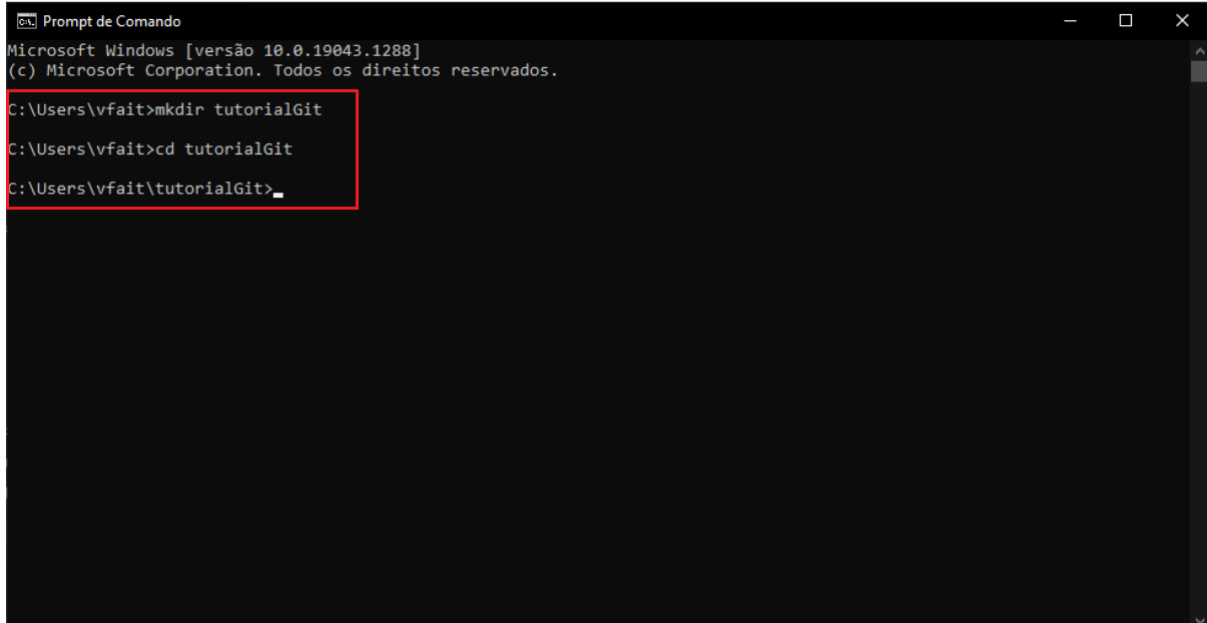
Funções e seus respectivos funcionamentos

1. Criando Repositório
 - 1.1. Local
 - 1.1.1. Commit
 - 1.2. Remoto
 - 1.2.1. Push
2. Clonando Repositório Remoto
3. Criando Branches
4. Criando Pull Request
5. Fazendo Merge

1 - Criando Repositório / 1.1 - Local

Para criar um novo repositório local siga as etapas a seguir:

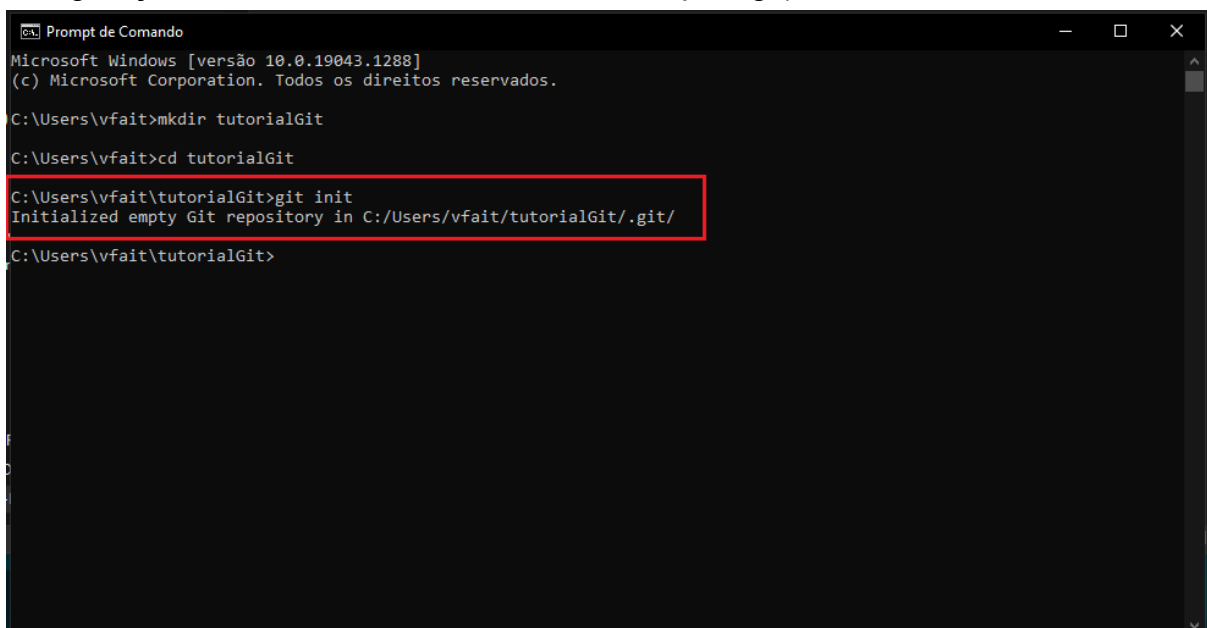
1 - Abra o CMD em sua máquina, digite **mkdir nomeDaPasta** (Este comando irá criar uma nova pasta), após digite **cd nomeDaPasta** (Este comando irá mover a edição de pasta para a nova pasta que criamos).



```
Microsoft Windows [versão 10.0.19043.1288]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\vfait>mkdir tutorialGit
C:\Users\vfait>cd tutorialGit
C:\Users\vfait\tutorialGit>_
```

2 - Digite **git init** (Este comando irá iniciar um arquivo git na sua pasta com as configurações iniciais criadas automaticamente pelo git)

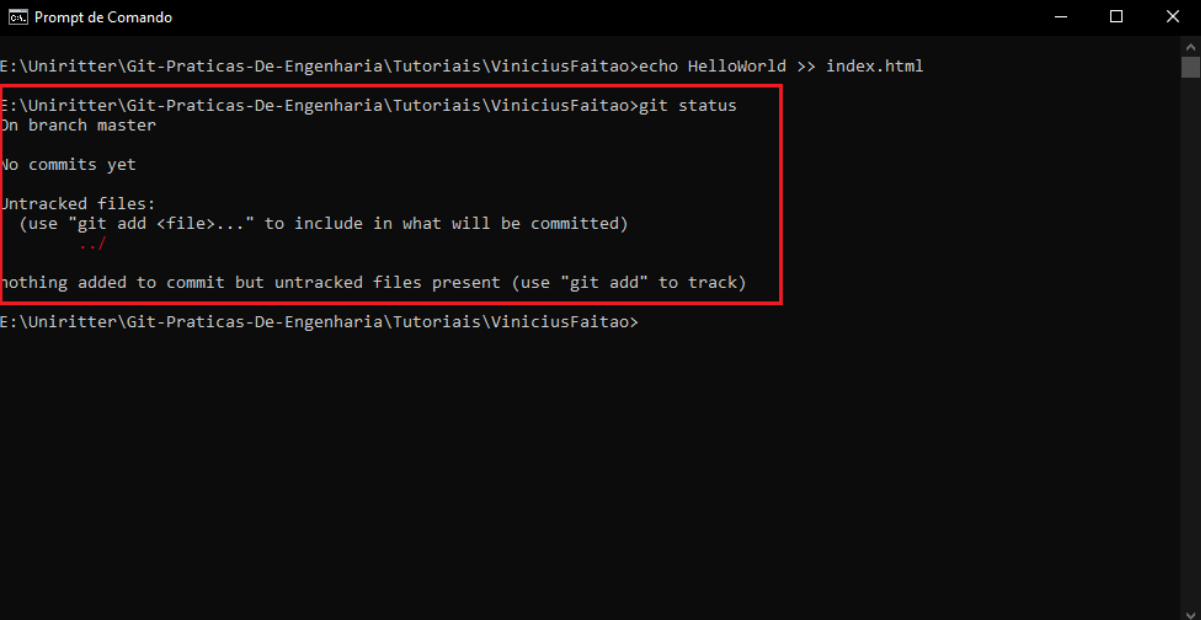


```
Microsoft Windows [versão 10.0.19043.1288]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\vfait>mkdir tutorialGit
C:\Users\vfait>cd tutorialGit
C:\Users\vfait\tutorialGit>git init
Initialized empty Git repository in C:/Users/vfait/tutorialGit/.git/
C:\Users\vfait\tutorialGit>
```

“Um repositório vazio do Git foi inicializado na pasta...”

3 - Para verificarmos a funcionalidade, podemos executar o comando **git status** (Este comando irá fornecer informações do git, como: Branch atual; Se contém algum commit; E se existe algum arquivo novo ou editado). Para esta etapa irei criar um arquivo com o comando **echo HelloWorld >> index.html** (Este comando irá criar um arquivo html escrito HelloWorld dentro), dentro da nossa pasta chamado "index.html" e após irei executar o comando **git status** novamente.



```
Prompt de Comando

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>echo HelloWorld >> index.html

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>git status
On branch master

No commits yet

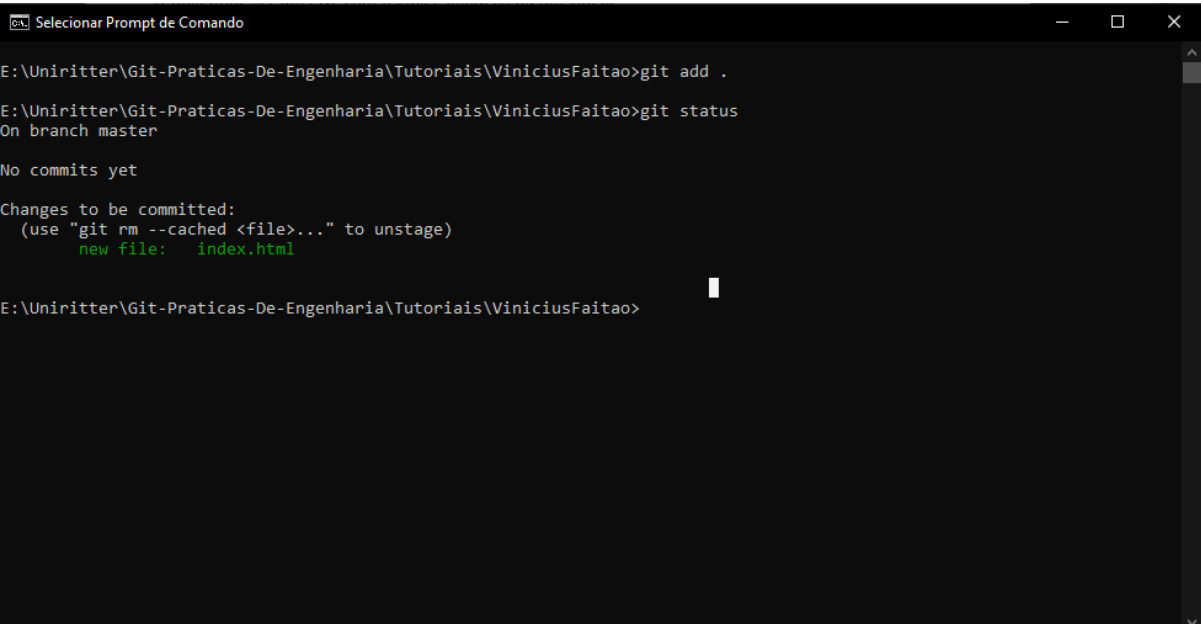
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ../

nothing added to commit but untracked files present (use "git add" to track)

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>
```

“Veja que temos **../**, o que significa que o git viu que tem uma modificação na pasta principal e instrui para que seja feito o git add”

4 - Agora digite **git add .** (Este comando irá organizar os novos arquivos e modificações, **sem exclusões**, no diretório atual e seus subdiretórios).



```
Selecionar Prompt de Comando

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>git add .

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>git status
On branch master

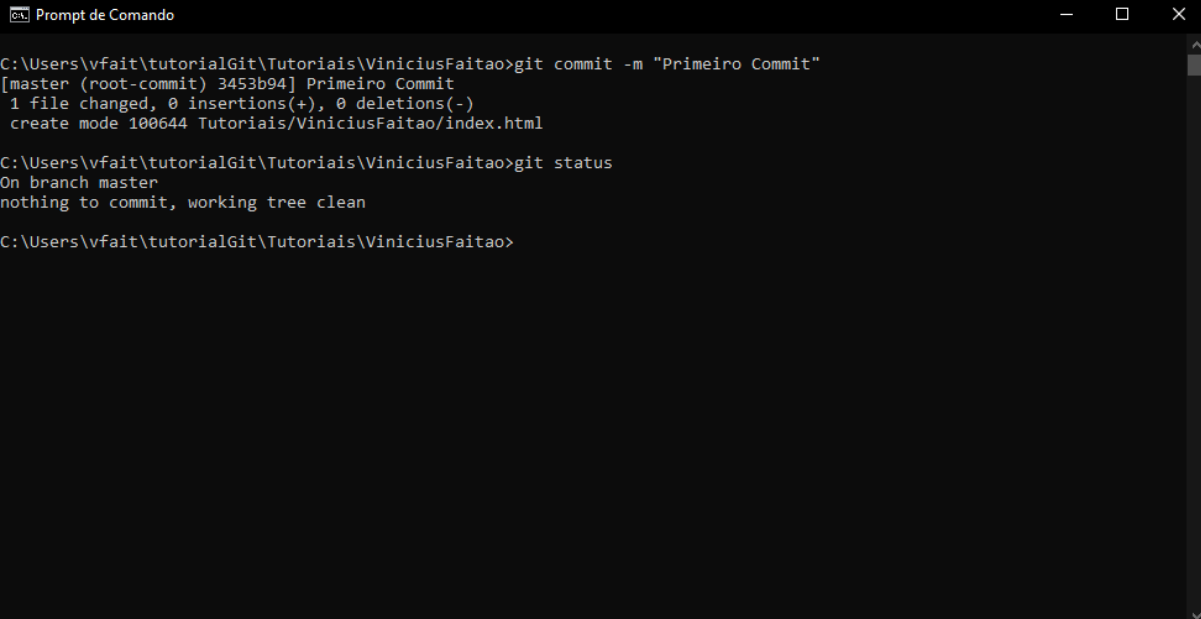
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html

E:\Uniritter\Git-Praticas-De-Engenharia\Tutoriais\ViniciusFaitao>
```

“Após efetuar o comando add . e status, vemos que o novo arquivo foi organizado e aguarda um commit”

5 - Nesta etapa iremos apenas fazer commit **git commit -m “mensagem”** (Este comando irá commitar os arquivos localmente e aguardar que seja feito **push** para o repositório remoto do git. “Veremos de push na parte 4 do tutorial”.



```
Prompt de Comando

C:\Users\vfait\tutorialGit\Tutoriais\ViniciusFaitao>git commit -m "Primeiro Commit"
[master (root-commit) 3453b94] Primeiro Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Tutoriais/ViniciusFaitao/index.html

C:\Users\vfait\tutorialGit\Tutoriais\ViniciusFaitao>git status
On branch master
nothing to commit, working tree clean

C:\Users\vfait\tutorialGit\Tutoriais\ViniciusFaitao>
```


“Após efetuar o comando commit e status, vemos que 1 arquivo foi alterado e as alterações salvas localmente, também vemos que não temos mais nenhuma alteração para commitar”

1 - Criando Repositório / 1.2 - Remoto

1 - Abra o [GitHub](#), digite o nome que deseja para o repositório, selecione se deseja que o projeto seja Public ou Private e clique em **Create repository**.

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner *  viniciusfaitao

Repository name * trabalhos-PESF-N ✓

Great repository names are short and memorable. Need inspiration? How about [sturdy-palm-tree?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

Create repository

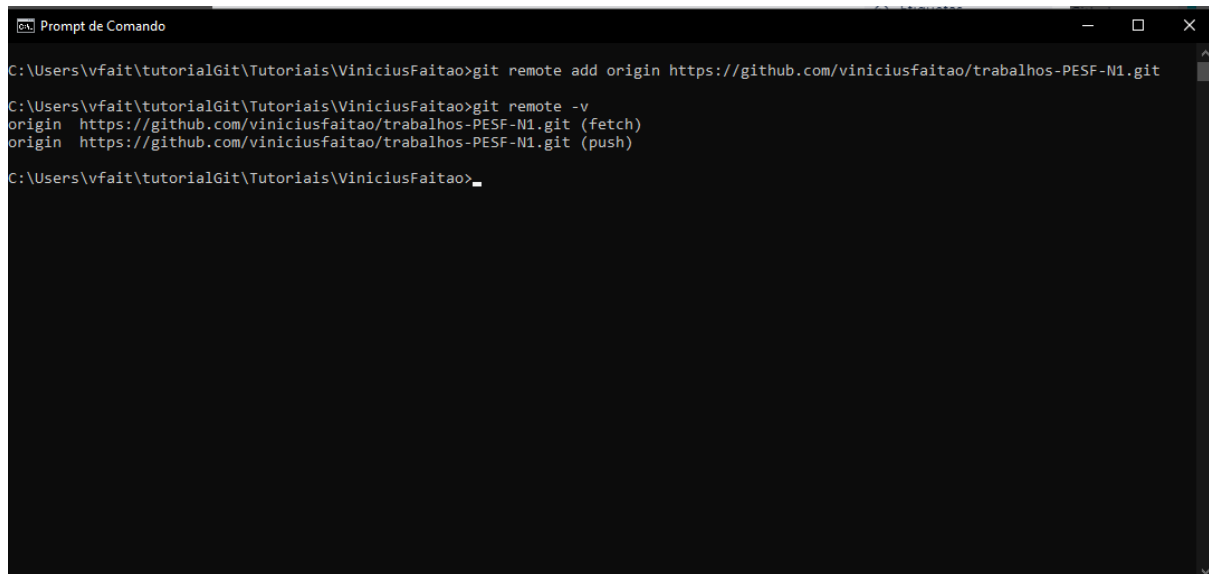
“Note alguma coisas: Por escolha própria, selecionei o repositório como Public, mas fica a seu critério e como boa prática mandei adicionar um README file (Este arquivo irá conter tutoriais de uso do seu projeto no github ou anotações importantes para si ou quer for visualizar)”

2 - Após criarmos o projeto iremos configurar nosso git remoto para receber o commit que fizemos localmente na etapa 1 do tutorial. Na página inicial do seu repositório, clique com **Code** e copie a URL **HTTPS**.



3 - Agora volte para eu CMD e digite o seguinte comando:

git remote add origin https://github.com/sua_URL_copiada_do_Github



“Após adicionar a url na minha origin remota, já efetuei conexão da minha pasta local com o repositório que criamos no github. Escrevendo o comando **git remote -v** podemos visualizar qual repositório origin irá nosso futuro push”

4 - Agora com a origin configurada iremos realizar nosso primeiro push. Vá para o CMD e digite **git push --set-upstream origin master**. Fique calmo, você não precisa decorar este comando, pois ele é solicitado sempre que você cria uma nova branca e mesmo que esqueça ele irá te avisar. Após a primeira vez, só será necessário escrever, **git push**.

```
Prompt de Comando

C:\Users\vfait\tutorialGit\tutoriais\ViniciusFaitao>git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

C:\Users\vfait\tutorialGit\tutoriais\ViniciusFaitao>
```

5 - Nesta etapa, provavelmente teremos um erro, pois como adicionamos um README file no repositório do github, teremos que fazer pull (pegar todos os arquivos existentes no repositório) e somente após iremos fazer push.

```
C:\Users\vfait\tutorialGit>git pull origin master --allow-unrelated-histories
From https://github.com/viniciusfaitao/trabalhos-PESF-N1
 * branch                master       -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 README.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 README.md

C:\Users\vfait\tutorialGit>git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
C:\Users\vfait\tutorialGit>
```

“Note que eu escrevi (**git pull origin master --allow-unrelated-histories**) pois como já fizemos um commit local antes mesmo de fazermos o remoto, o git precisa fazer MERGE dos arquivos existentes no repositório com os nossos. Agora nós podemos fazer git push, como o próprio git já nos instruiu após o comando git status”

6 - Execute o comando **git push**.

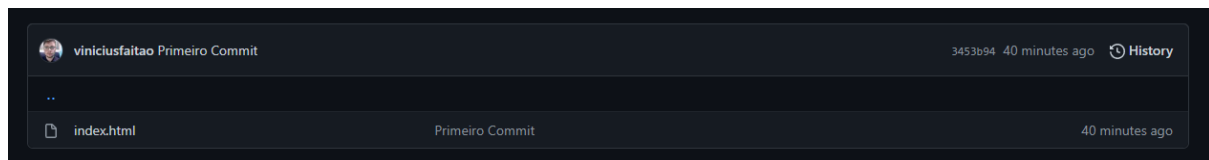
```
C:\Users\vfait\tutorialGit>git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (7/7), 621 bytes | 621.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/viniciusfaitao/trabalhos-PESF-N1.git
 7f4103e..b7c87d4  master -> master

C:\Users\vfait\tutorialGit>
```

7 - Pronto, agora vá ao seu repositório no github e atualize a página, logo já verá nossos arquivos lá.



Entrando na pasta, vemos o arquivo index.html que criamos no tutorial 1 e estava salvo no repositório local.

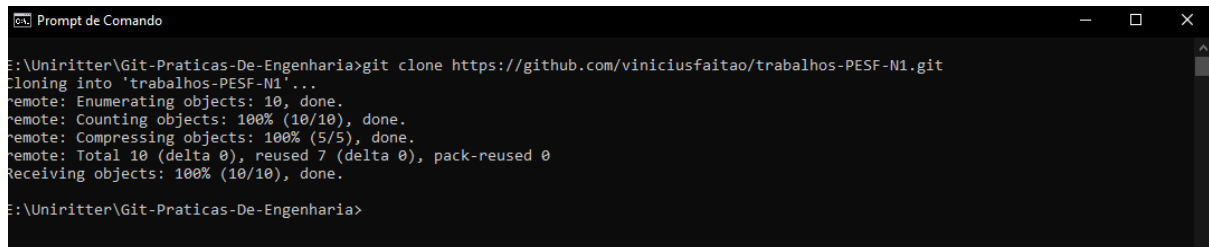


2 - Clonando Repositório Remoto

1 - Para começar, abra o CMD, crie uma nova pasta **“Não pode ser a mesma dos tutoriais anteriores”** e entre na pasta (parte 1.1 etapa 1 do tutorial).

2 - Após vá no seu github e copie a URL **HTTPS** (parte 1.2 etapa 2 do tutorial).
“https://github.com/sua_URL_copiada_do_Github”

3 - Digite no CMD **git clone https://github.com/sua_URL_copiada_do_Github**

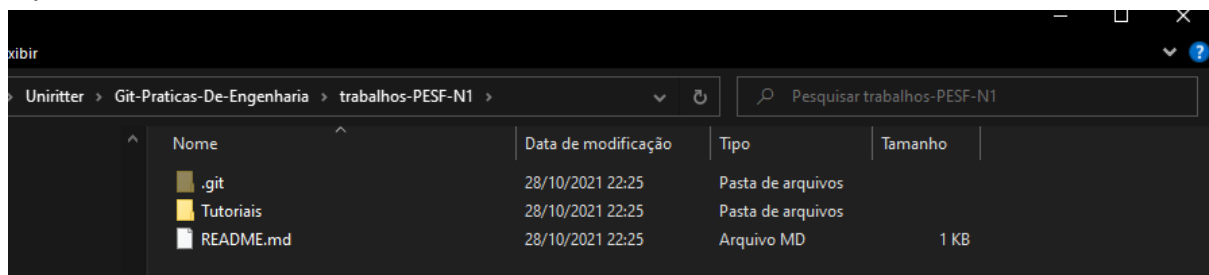


```

E:\Uniritter\Git-Praticas-De-Engenharia>git clone https://github.com/viniciusfaitao/trabalhos-PESF-N1.git
Cloning into 'trabalhos-PESF-N1'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 10 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
E:\Uniritter\Git-Praticas-De-Engenharia>
```

“Podemos ver que o git clone foi efetuado com sucesso e o repositório está na nova pasta que criamos.”

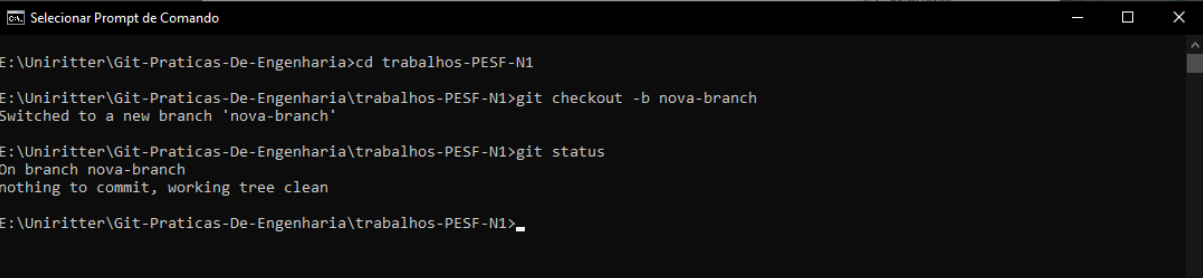
4 - Agora para validar, entre na pasta criada e verifique que seus arquivos do repositório já estão na pasta. Lembrando que como foi feito um git clone, não é necessário mais nenhuma configuração extra, todos arquivos modificados dentro desta nova pasta, poderá ser feito um **git add**, **git commit** e **git push** para o repositório remoto.



“Agora todo arquivo pode ser alterado pela pasta e feito push por cmd, inclusive o readme.md”

3 - Criando Branches

1 - No mesmo CMD e na mesma pasta criaremos uma branch, um novo arquivo e enviaremos para o repositório remoto. Vá para o CMD e digite **git checkout -b nova-branch** (Este comando irá criar uma nova branch e por ter escrito **-b** já iremos mover da master para a branch nova automaticamente)



```
Selecionar Prompt de Comando

E:\Uniritter\Git-Praticas-De-Engenharia>cd trabalhos-PESF-N1

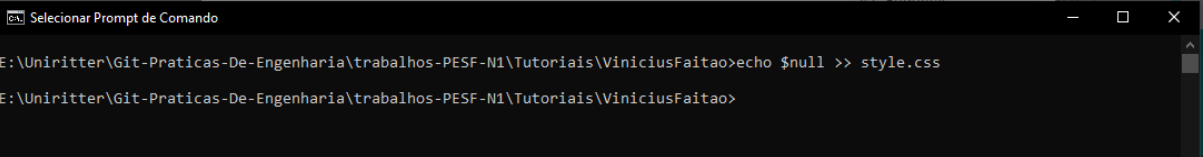
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1>git checkout -b nova-branch
Switched to a new branch 'nova-branch'

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1>git status
On branch nova-branch
nothing to commit, working tree clean

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1>
```

“Podemos ver ao dar git status que está escrito On branch nova-branch, ou seja, já estamos fazendo uso da mesma e podemos modificar o sistema sem medo que nada irá alterar na master por descuido.”

2 - Vamos criar um novo arquivo chamado style.css **echo \$null >> style.css**.



```
Selecionar Prompt de Comando

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>echo $null >> style.css

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>
```

“Pronto, se for verificado na pasta, terá um novo arquivo chamado (style.css).”

3 - Agora faça git add, git commit e git push (parte 1.1 etapa 4 e 5 do tutorial, parte 1.2 etapa 4).

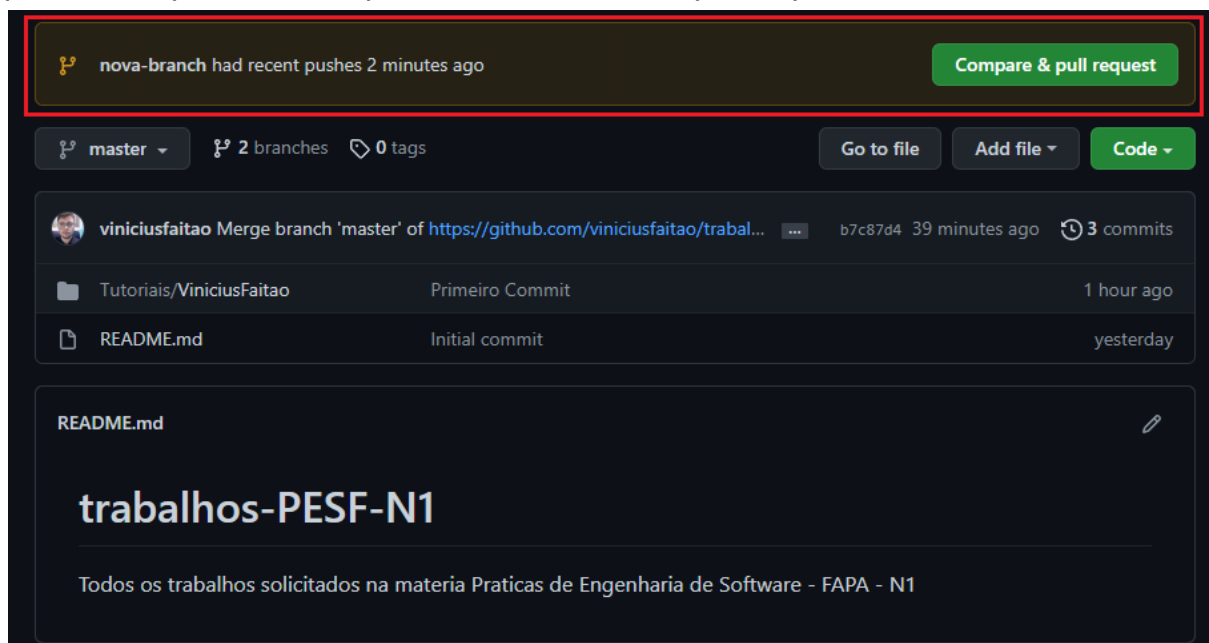
```
Prompt de Comando

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>echo $null >> style.css
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git add .
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git commit -m "Adicionando style.css"
[nova-branch 5d8ed63] Adicionando style.css
1 file changed, 1 insertion(+)
 create mode 100644 Tutoriais/ViniciusFaitao/style.css
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git push
fatal: The current branch nova-branch has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin nova-branch
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git push --set-upstream origin nova-branch
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 426 bytes | 426.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'nova-branch' on GitHub by visiting:
remote:   https://github.com/viniciusfaitao/trabalhos-PESF-N1/pull/new/nova-branch
remote:
To https://github.com/viniciusfaitao/trabalhos-PESF-N1.git
 * [new branch]      nova-branch -> nova-branch
Branch 'nova-branch' set up to track remote branch 'nova-branch' from 'origin'.
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>
```

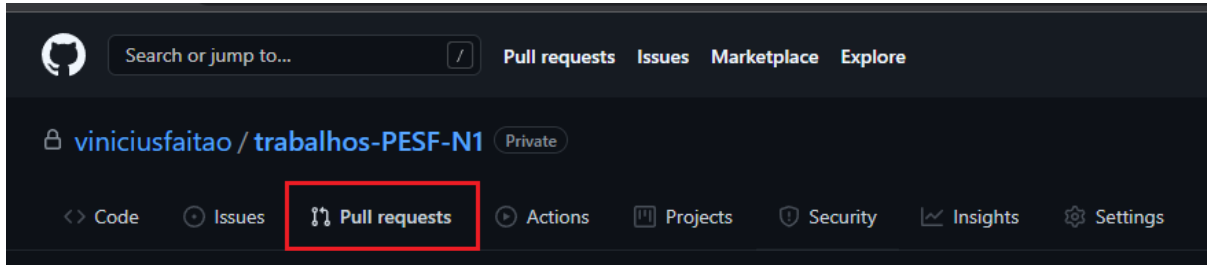
“Repare que quando estamos em uma branch, ao fazer git push, não escrevemos master, pois no final do command line, tu diz em qual branch esse código deve ser feito o push, assim quando estivermos no github remoto, veremos nas branches essa nova branch criada.”

4 - Pronto, no Github Remoto logo na página inicial, teremos uma notificação que foi feito recentemente um push de uma nova branch, a branch que criamos.“Veja a próxima etapa do tutorial para saber como fazer pull request”.

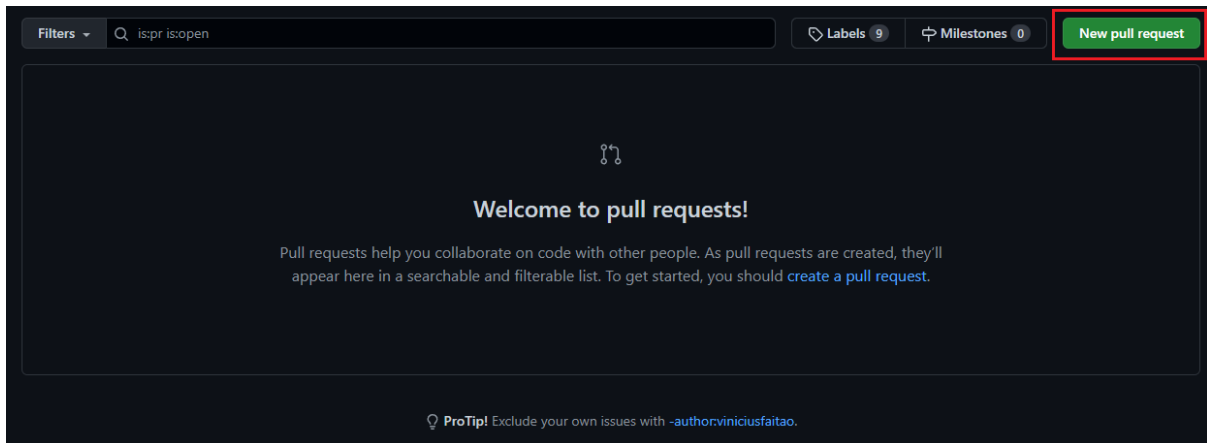


4 - Criando Pull Request

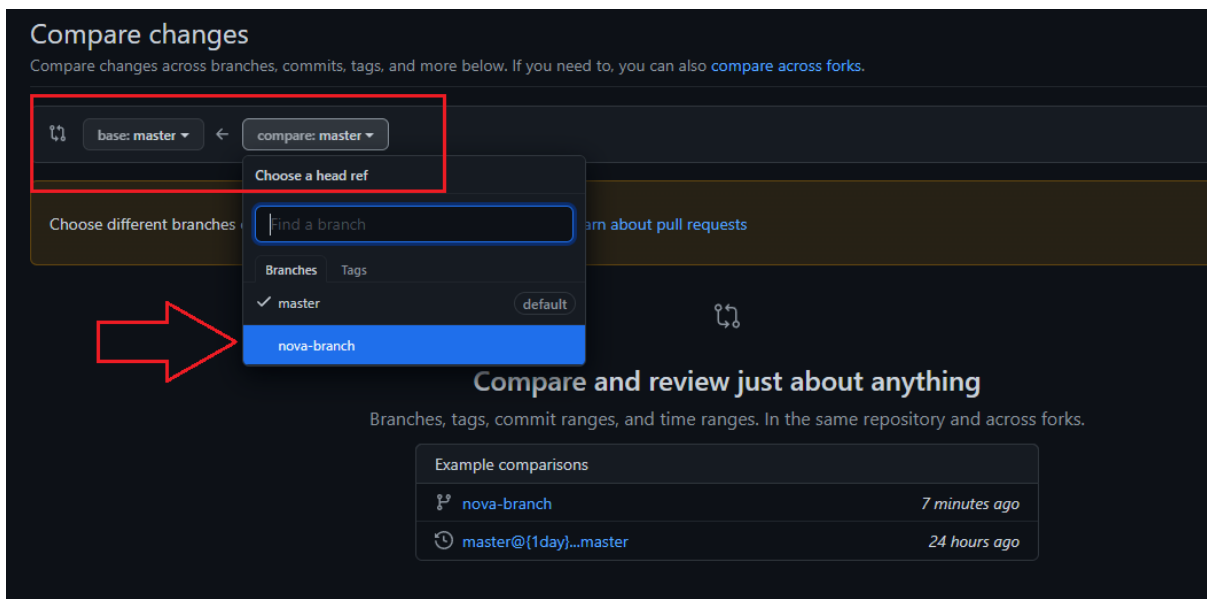
1 - No repositório do github remoto, clique no menu **Pull requests**.



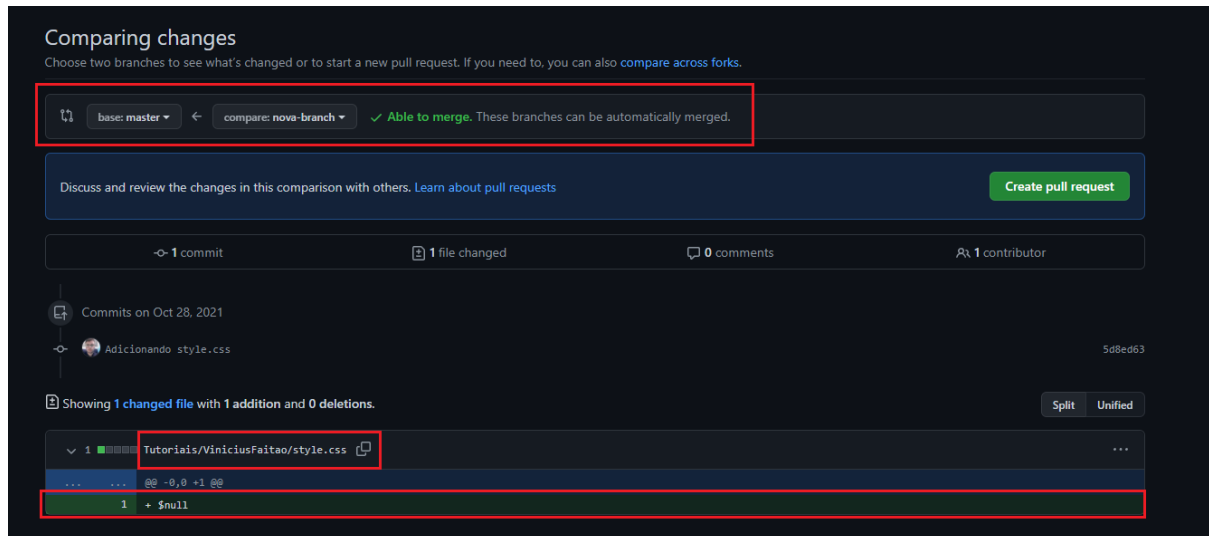
2 - Já na nova página, clique no botão **New pull request**.



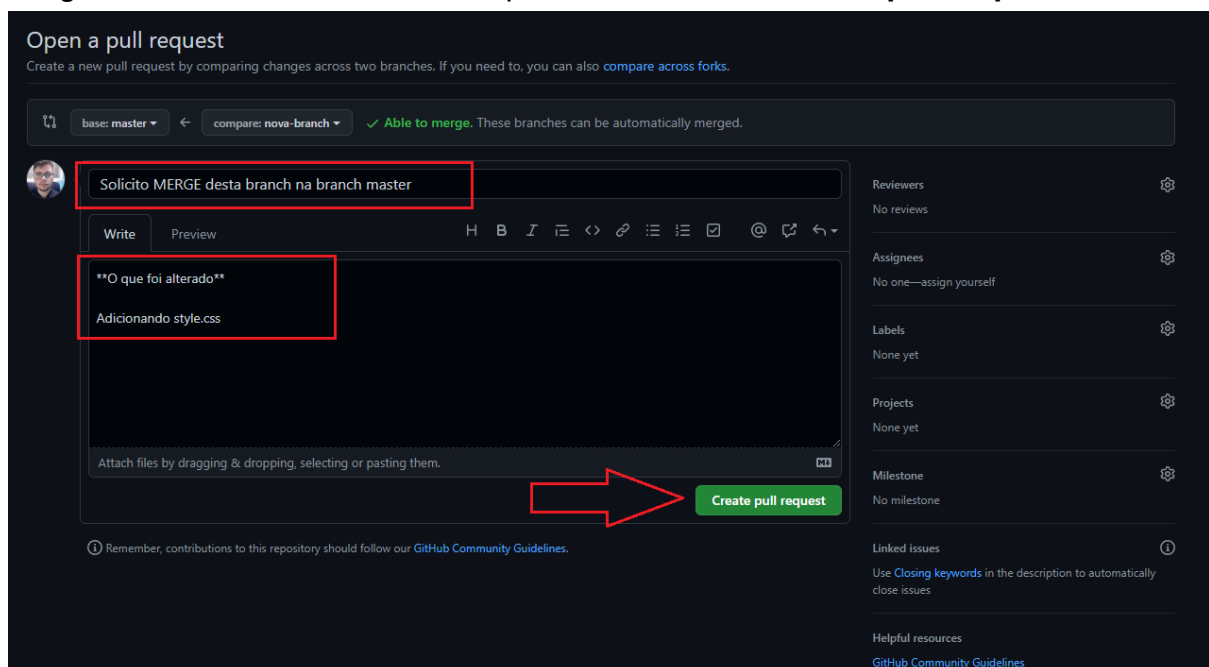
3 - No select **base**: deixe sua branch **DEFAULT** selecionado e no select **compare**: selecione a nova branch que fizemos commit "**nova-branch**".



4 - Após selecionarmos a branch, terá um histórico de edições na parte de baixo. Onde podemos ver marcado que estamos adicionando um novo arquivo no repositório chamado **style.css** e na linha verde vemos o que tem dentro deste arquivo que é novo também, no código de criação, colocamos null, então o conteúdo está como null.



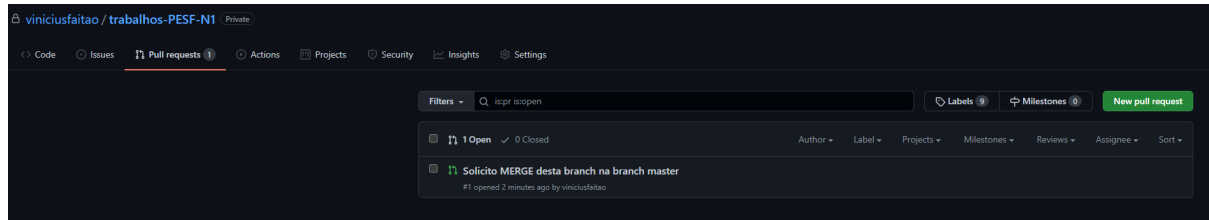
5 - Agora clique no botão Create pull request e iremos para esta etapa abaixo, nela você irá por o título da sua pull request e no campo textual, uma breve descrição do que foi feito nesta nova branch. Isso é necessário quando outras pessoas irão avaliar seu PR para verificar a integridade do arquivo e poder permitir que seja feito Merge na master. Para finalizar, clique novamente em **Create pull request**.



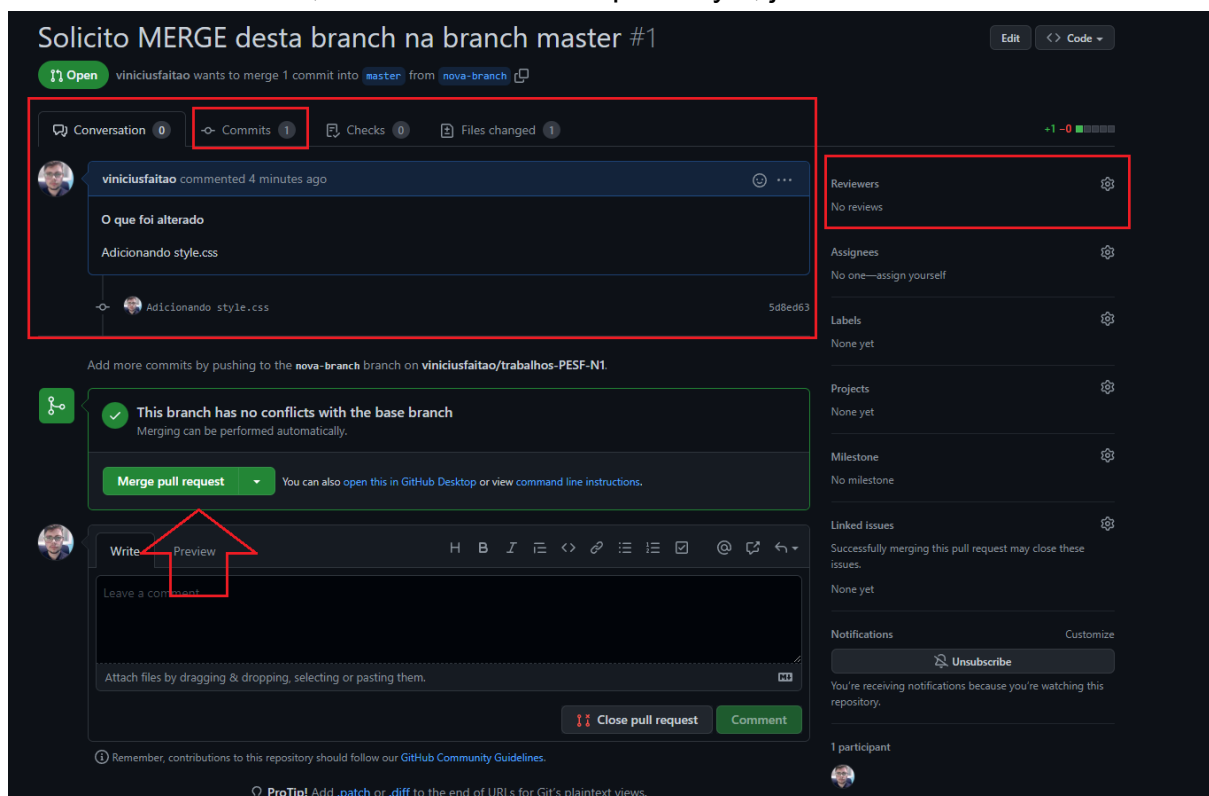
1 - Fazendo Merge

Agora é a etapa final do nosso tutorial. O desejado MERGE!!!!

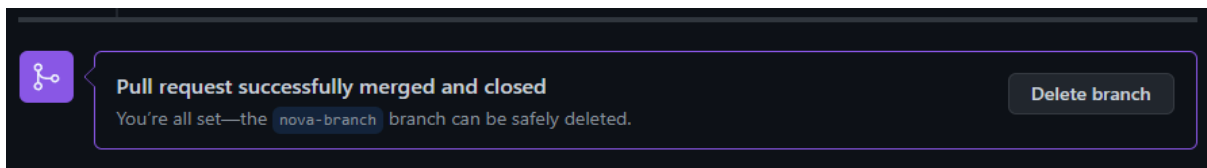
1 - Na aba Pull requests, iremos ver uma lista de todos os pull requests criados.



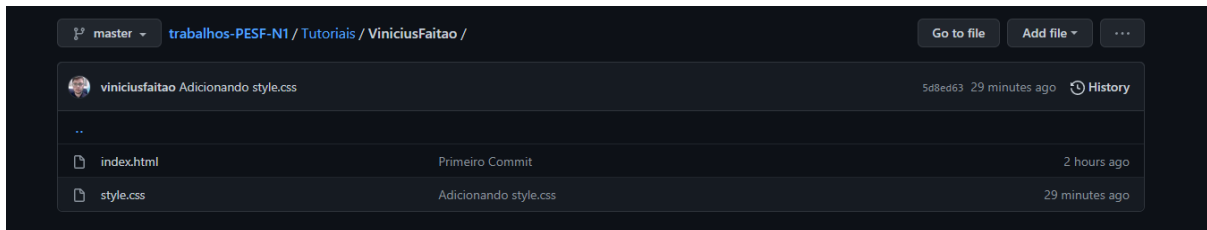
2 - Entre no PR desejado e veremos uma série de informações importantes. Podemos visualizar quem criou o PR acompanhado da descrição feita explicando o que foi feito. Podemos visualizar a aba **Commits** e nela veremos o que o usuário modificou, criou ou editou em cada commit. Podemos ver quem são os **Reviewers**, esta opção normalmente é muito utilizada em projetos compartilhados, para que nada vá de uma branch para a master sem uma análise do superior. Então por último o mais desejado, clique no botão **Merge pull request** e assim fará merge da sua branch na master, enviando o novo arquivo style, junto com o index.html.



3 - Verificando o resultado.



“Mensagem após Merge realizado.”



“Ir até a pasta modificada e verificar se o novo arquivo style.css consta no local.”

Extra

1 - Altere para a branch master **git checkout master** e faça **git pull** para atualizar seu repositório local novamente.

```
E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 659 bytes | 11.00 KiB/s, done.
From https://github.com/viniciusfaitao/trabalhos-PESF-N1
   b7c87d4..6a414cd master    -> origin/master
Updating b7c87d4..6a414cd
Fast-forward
   Tutoriais/ViniciusFaitao/style.css | 1 +
   1 file changed, 1 insertion(+)
   create mode 100644 Tutoriais/ViniciusFaitao/style.css

E:\Uniritter\Git-Praticas-De-Engenharia\trabalhos-PESF-N1\Tutoriais\ViniciusFaitao>
```