Introdução a Git e ao GitHub

MATEUS SANTOS VALENÇA

Introdução

O que é Git?

O Git é um sistema de controle de versão de projetos. Onde é possível várias pessoas contribuírem de forma que não seja necessário a modificação definitiva do projeto original.

No Git seus projetos ficam salvos localmente no computador.

Introdução

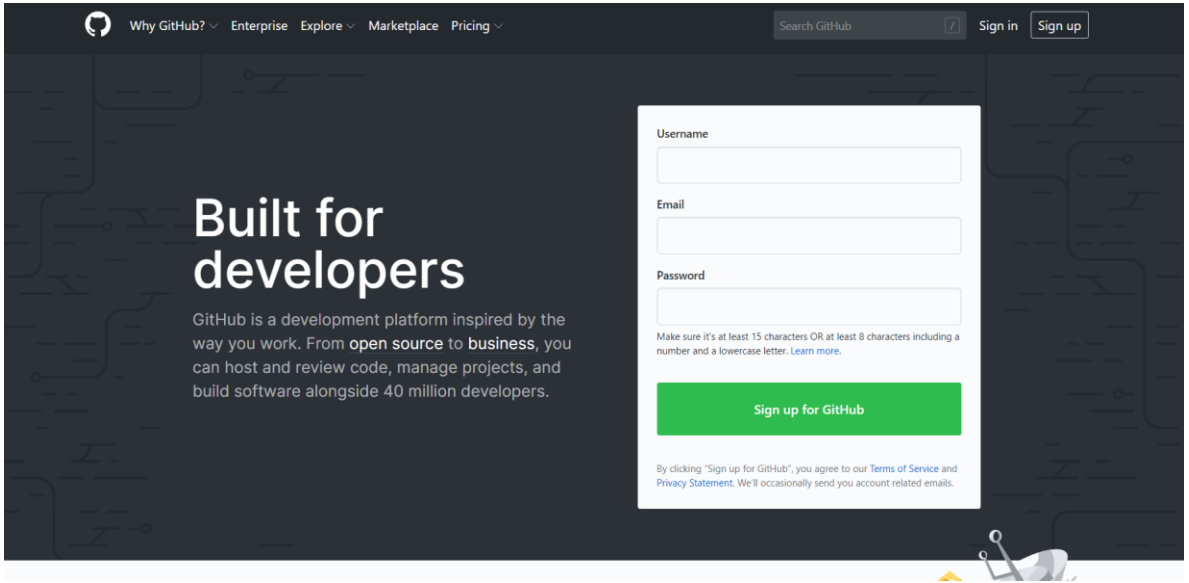
- **O que é GitHub?**

  - Ferramenta com Git
  - Gratuito
  - Backup na nuvem dos seus projetos
  - Compartilhamento prático

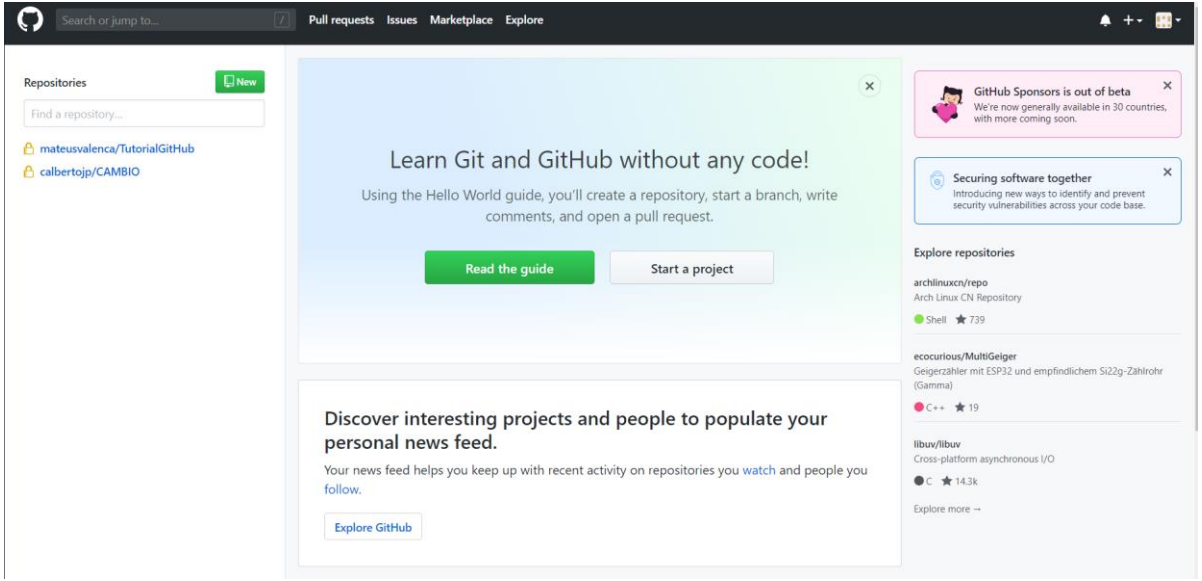
Primeiros Passos

Criar uma conta

Acessar github.com e criar uma conta



Tela Inicial



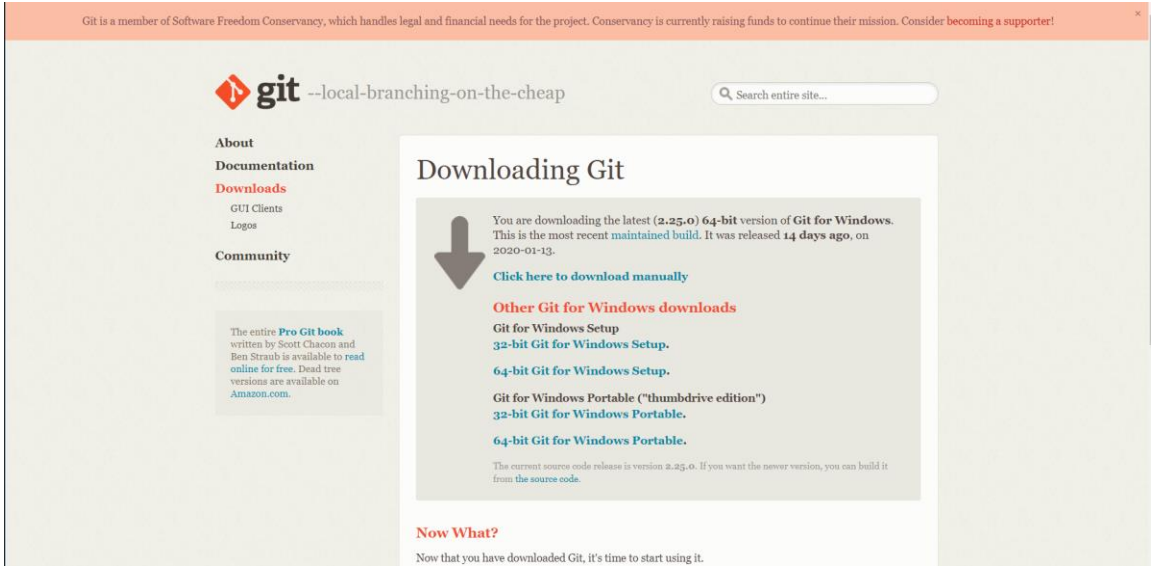
Instalado o Git Bash

Para usuários do Windows:

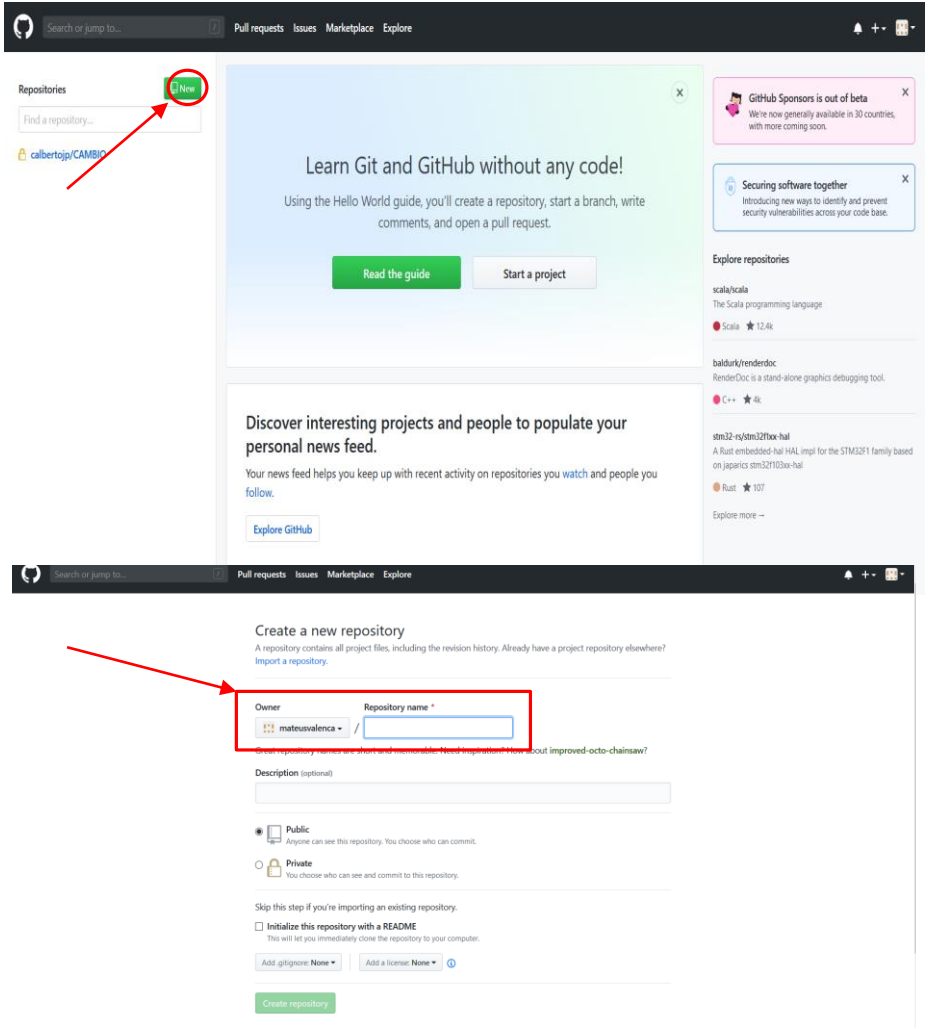
Acessar https://git-scm.com/download/win para baixar o git bash

Para usuários do Linux:

No prompt de comando digite:
“sudo apt-get install git”



Criando meu primeiro projeto

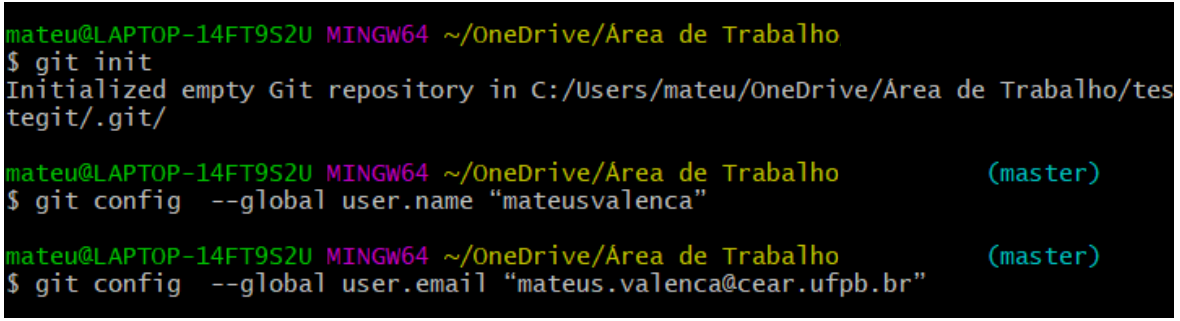


Salvando o seu usuário

Com o git bash aberto digite:
git init
git config --global user.name “nome de usuário”

git config –global user.email “e-mail cadastrado”

Essas configurações estarão salvas no seu pc no endereço ~\.gitconfig , onde ~ representa sua home, no meu caso:
C:\Users\mateu\.gitconfig

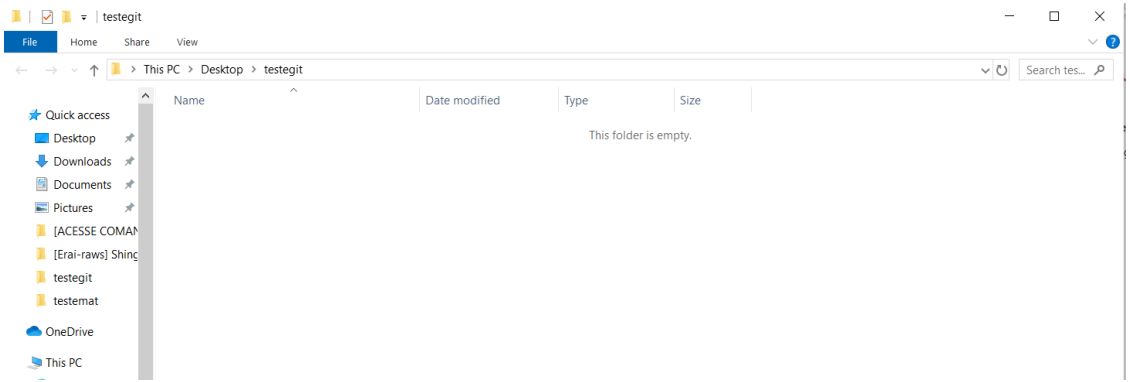


Clonando o projeto do GitHub

O processo de clonagem permite pegar aquele projeto existente no GitHub e trazer para o nosso git local dentro de uma pasta com aquele nome.
O comando usado é:
git clone link_do_projeto

Dica: no bash do Windows para colar algo copiado você usa o botão shift+Insert

```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho (master)
$ git clone https://github.com/mateusvalenca/testegit
Cloning into 'testegit'...
warning: You appear to have cloned an empty repository.
```

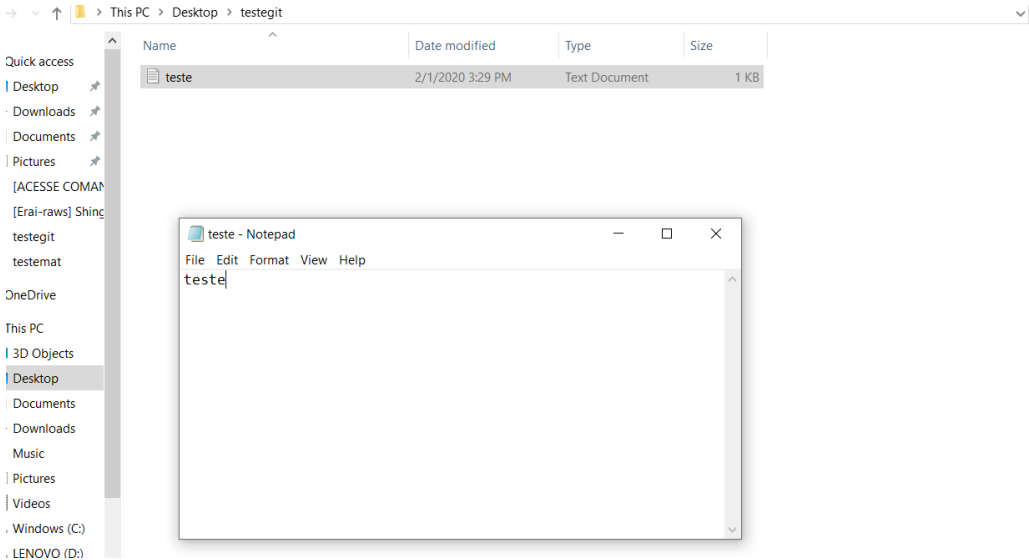


Comandos Iniciais

Git add nome_dos_arquivos ou git add . :

Esse comando vai pegar os arquivos e jogar num lugar chamado INDEX onde ele está sendo preparado para entrar no GitHub na próxima revisão feita.

Crie um arquivo .txt novo e digite algo nele.



Comandos Iniciais

Git commit –m “Mensagem”

Esse comando vai comentar todos os arquivos no INDEX com a mensagem escolhida.

Git push:

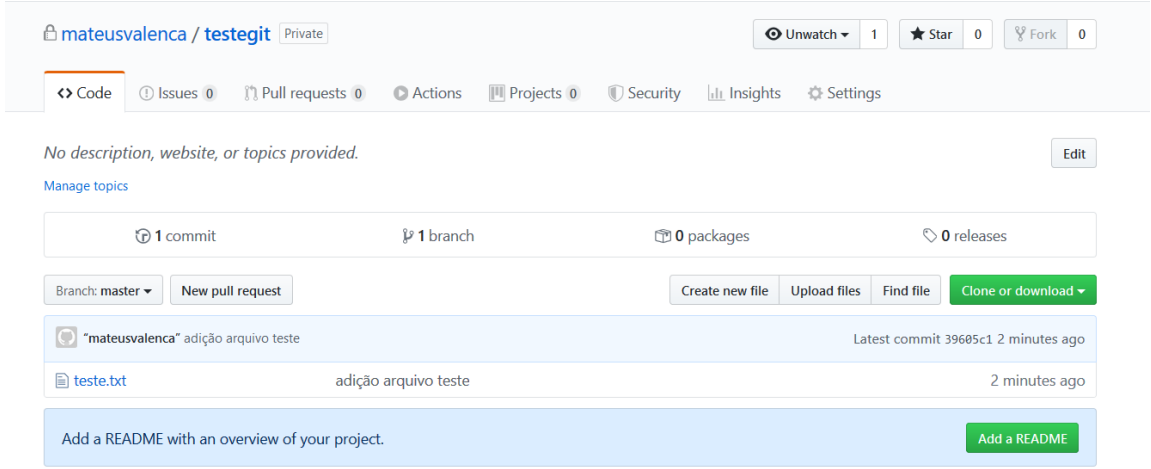
Insere os arquivos do INDEX no seu GitHub

```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git add .

mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git commit -m "adição arquivo teste"
[master (root-commit) 39605c1] adição arquivo teste
 1 file changed, 1 insertion(+)
 create mode 100644 teste.txt

mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 245 bytes | 81.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mateusvalenca/testegit
 * [new branch]      master -> master
```

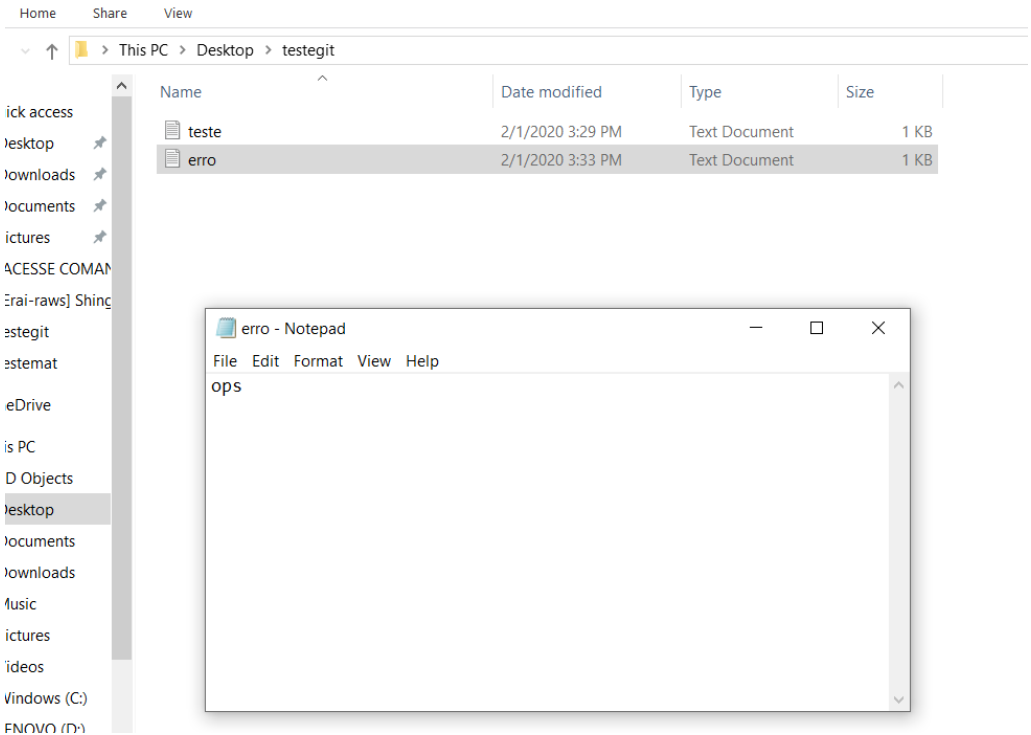
Mudanças feitas



E se a alteração está errada?

No seu git, faça um novo arquivo e repita o processo anterior para adicioná-lo ao seu git:

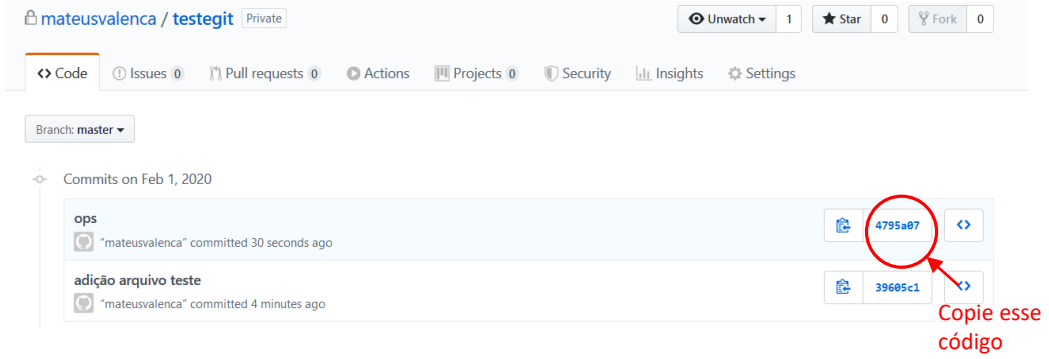
Dica:
1- Git add .
2- git commit –m “mensagem”
3- git push link



Revertendo o que foi feito!

No GitHub, vá na aba commits e veja o último commit feito.

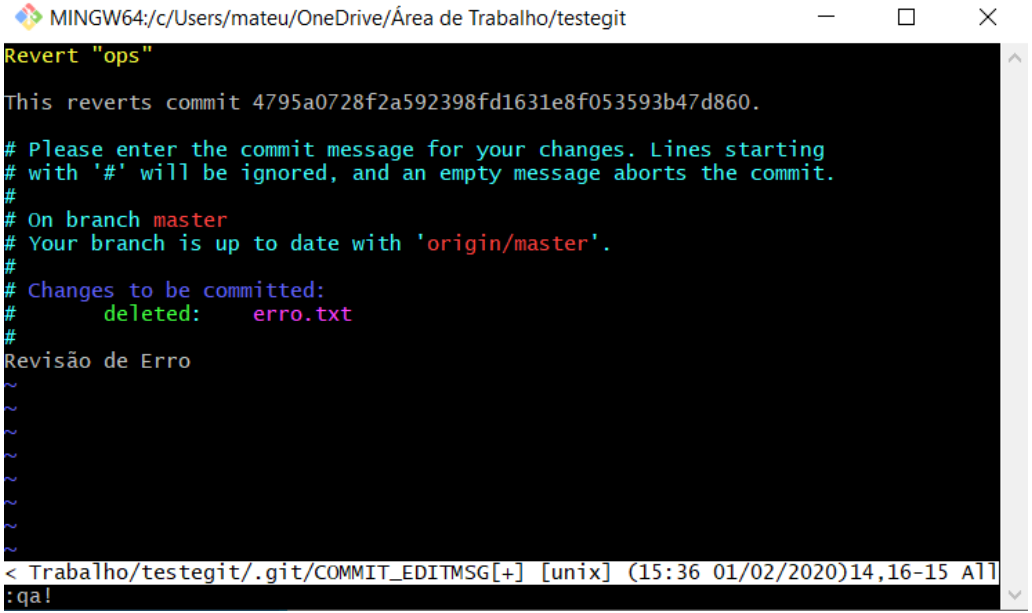
Comando: git revert <código>




```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git revert 4795a07
```

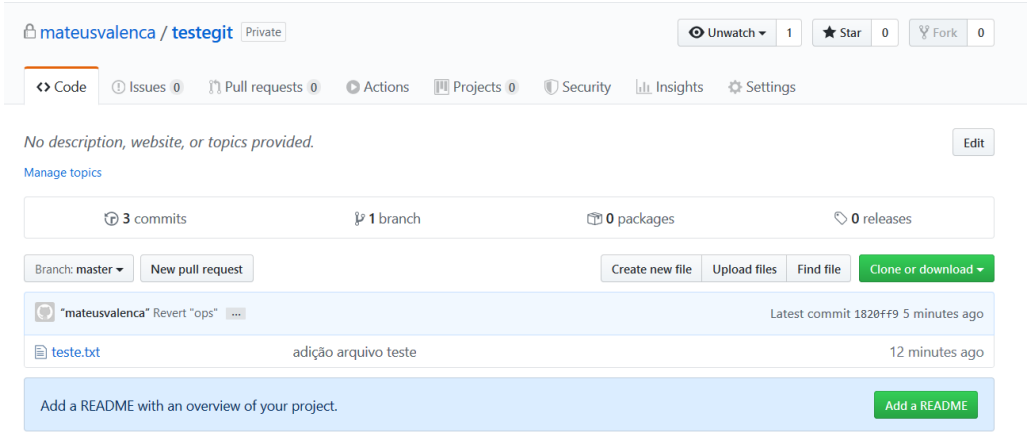
Revertendo o que foi feito!

Nesta tela você tem opção de inserir comentários, aperte qualquer tecla e em seguida vá para a linha onde deseja comentar. Quando terminar aperte esc para sair do modo de edição e digite ":qa!" sem aspas e enter para finalizar.



Revertendo o que foi feito!

A reversão foi local, para mudar no GitHub basta usar git push.



Branch

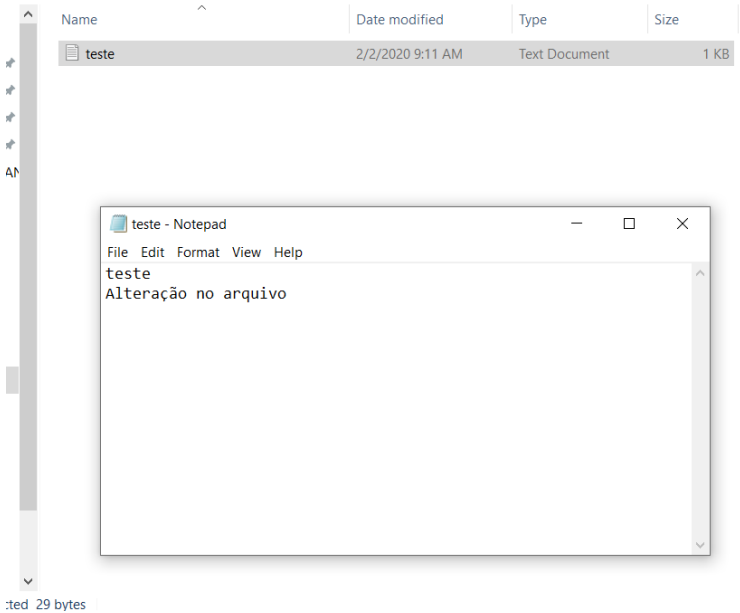
Para começar precisamos criar o branch na pasta local.
O branch criado vai ser uma cópia do master, mas o que for alterado aqui não é alterado no master.

```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git branch novo

mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git checkout novo
Switched to branch 'novo'
```

Branch

Vamos praticar!
Altere o arquivo Testeadd.txt e em seguida tente fazer o de sempre:
git add .
git commit –m “mensagem”
git push



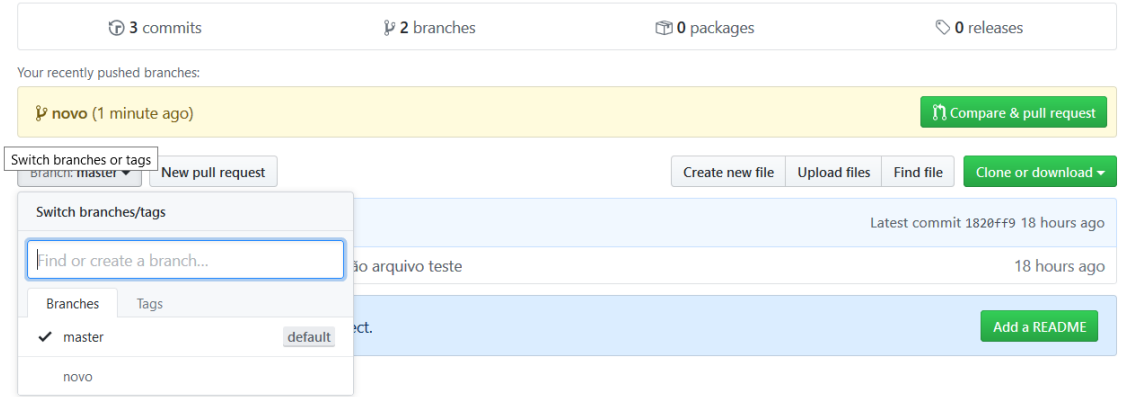
Indicando quem é o master do seu branch

O comando
git --set-upstream origin branch
Só é necessário usar na primeira vez que utilizar o push nesse branch, depois só:
git push

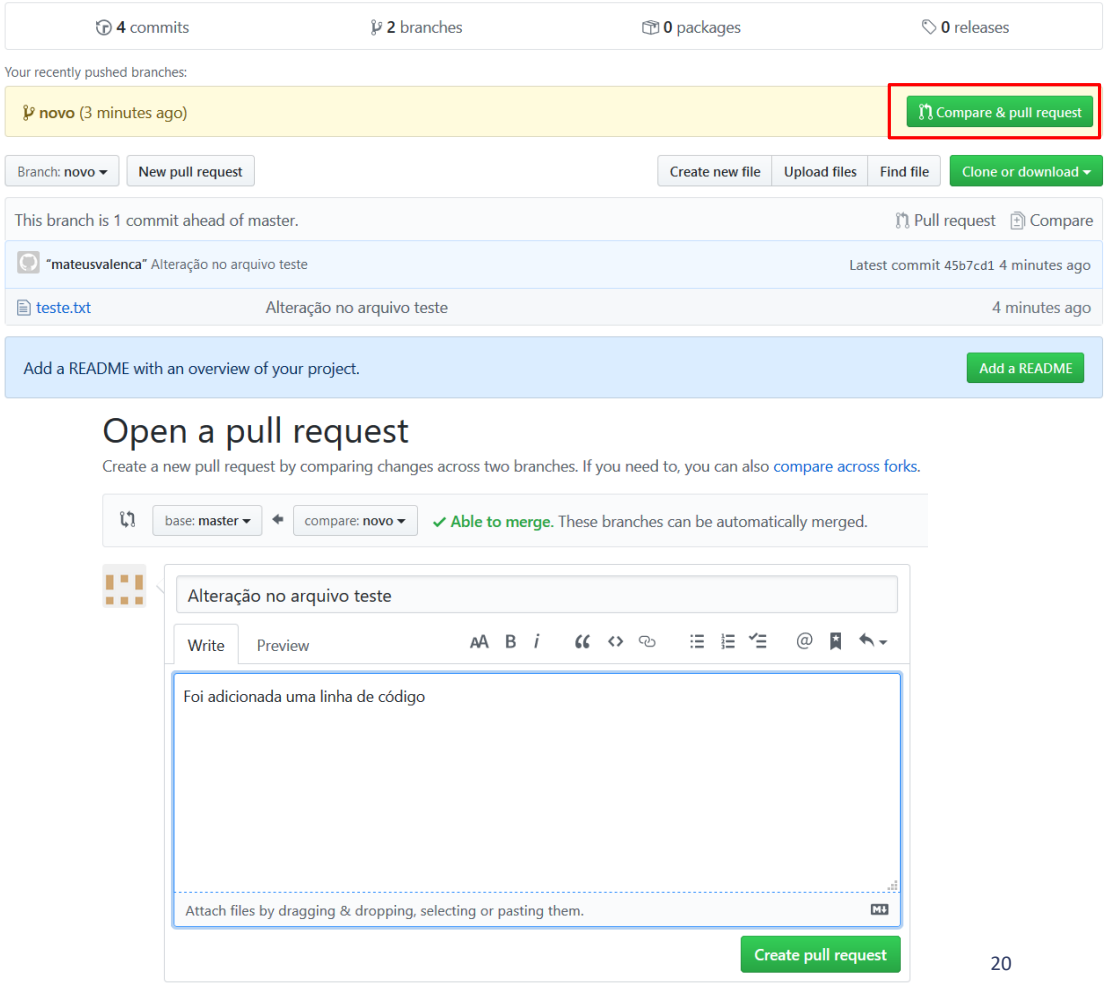
```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (novo)
$ git push
fatal: The current branch novo has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin novo


mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (novo)
$ git push --set-upstream origin novo
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 305 bytes | 101.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'novo' on GitHub by visiting:
remote:      https://github.com/mateusvalenca/testegit/pull/new/novo
remote:
To https://github.com/mateusvalenca/testegit
 * [new branch]      novo -> novo
Branch 'novo' set up to track remote branch 'novo' from 'origin'.
```



Requisitando a mudança no master



Master verificando mudanças

Primeiro volte ao ramo master:
git checkout master

Cheque se há diferenças entre os ramos:
git diff branch

Verificado as diferenças, estou de acordo e quero mudar o master:
git merge branch

Isso vai mesclar os branches localmente, depois é necessário usar git push para realizar as alterações no GitHub

```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (novo)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git diff novo
diff --git a/teste.txt b/teste.txt
index 0d16cee..bb8abfc 100644
--- a/teste.txt
+++ b/teste.txt
@@ -1,2 +1 @@
-teste
-Alteração no arquivo
\ No newline at end of file
+teste
\ No newline at end of file
```

```
mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git merge novo
Updating 1820ff9..45b7cd1
Fast-forward
 teste.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

mateu@LAPTOP-14FT9S2U MINGW64 ~/OneDrive/Área de Trabalho/testegit (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/mateusvalenca/testegit
   1820ff9..45b7cd1  master -> master
```

Vamos Praticar!

Primeiro acessem:
https://github.com/mateusvalenca/matriculas
E clonem esse projeto.