

Primeiros passos com o GitHub – Guia Simples

Este tutorial foi desenvolvido por mim, aluno do Instituto Federal do Paraná, Campus Umuarama, João Paulo de Souza, aluno do curso Técnico em Informática Subsequente no período Noturno. Com auxílio do professor e coordenador do GEP, Grupo de Estudos em Programação, Marcelo Pereira, Professor de Programação do Campus.

Git Hub é uma ferramenta bem prática de controle de versões baseada em git , praticamente um terminal do Linux e Mac ou cmd do Windows. Nela é possível você controlar bem mais facilmente todos os seus códigos. É uma ferramenta padrão entre os desenvolvedores de software, existem outros com o git que funcionam semelhantemente.

Aqui está o link do site do github: <https://github.com/>

Lá você poderá fazer sua conta no github

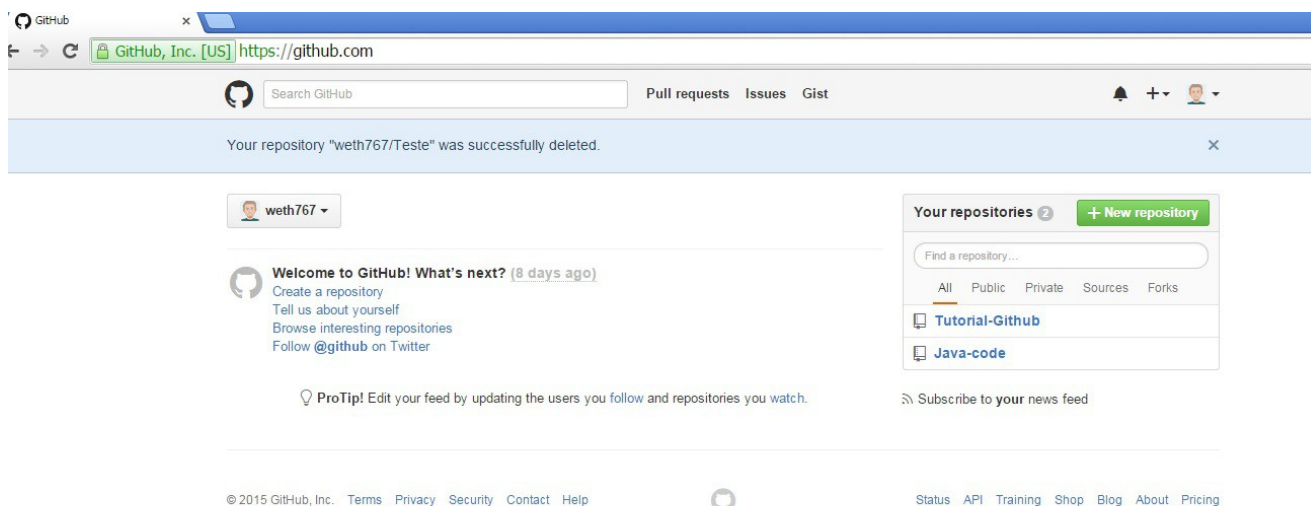
Aqui está o link para download do github e do gitshell, os dois vem junto no mesmo programa <https://desktop.github.com/>

Depois de instalado vão aparecer assim na sua área de trabalho.

Depois você entrará no github e lá aparecerá para você entrar com sua conta e e-mail, depois de feito isso você pode fecha-lo.

Agora vamos no site do git hub criar um repositório para fazer um teste. Entramos com nossa conta no git hub.





Aparecerá desta forma lá, vamos clicar no botão verde, novo repositório.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

weth767 / Teste

Great repository names are short and memorable. Need inspiration? How about **itchy-moo**.

Description (optional)

☒

Public

Anyone can see this repository. You choose who can commit.

☐



Private

You choose who can see and commit to this repository.

☒

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None



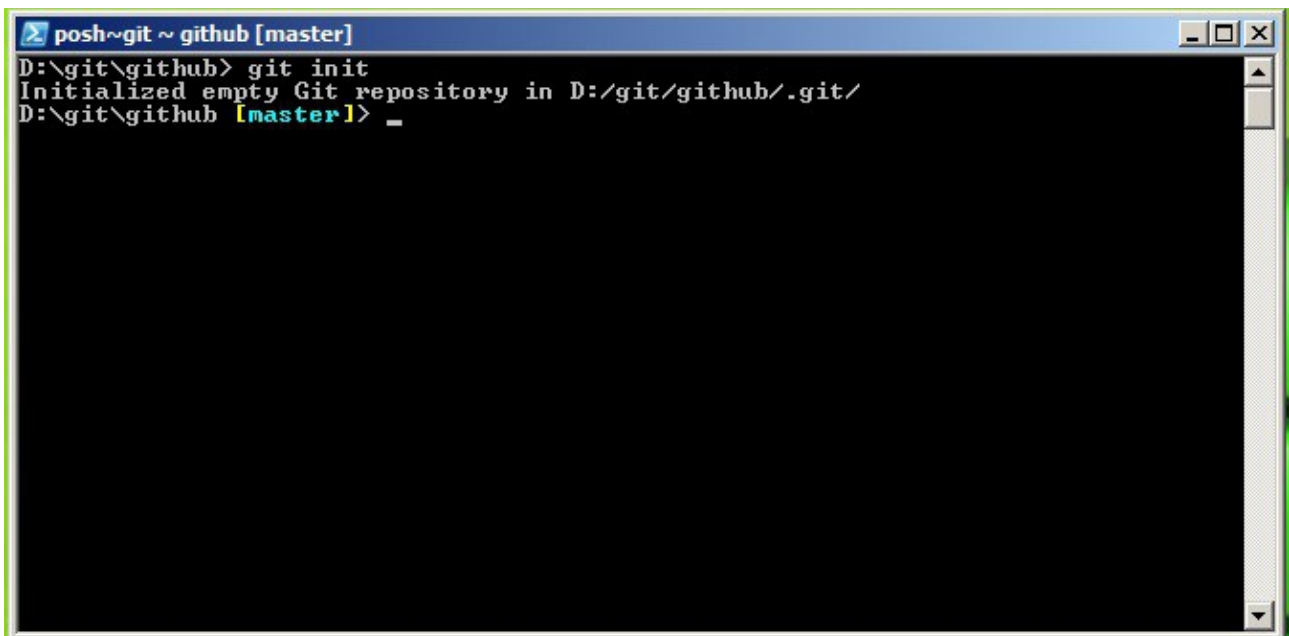
Create repository



Agora pode fechar seu navegador se quiser e o programa git hub, vamos usar agora o git shell.

Entre no git shell e digite um e siga os passos abaixo.

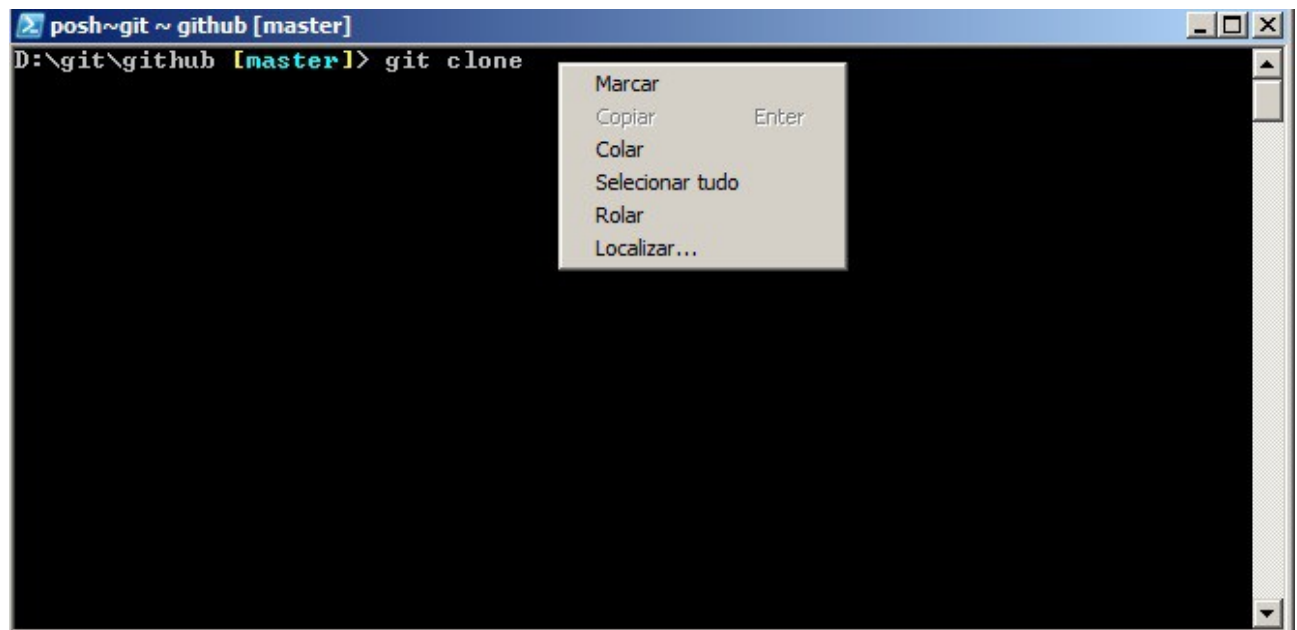
Passo 1: Inicialmente para criarmos um novo repositório local onde serão depositados seus arquivos, usamos o comando: git init

A screenshot of a Windows terminal window titled "posh~git ~ github [master]". The terminal shows the command "git init" being executed at the path "D:\git\github". The output is "Initialized empty Git repository in D:/git/github/.git/". The prompt then changes to "D:\git\github [master]>".

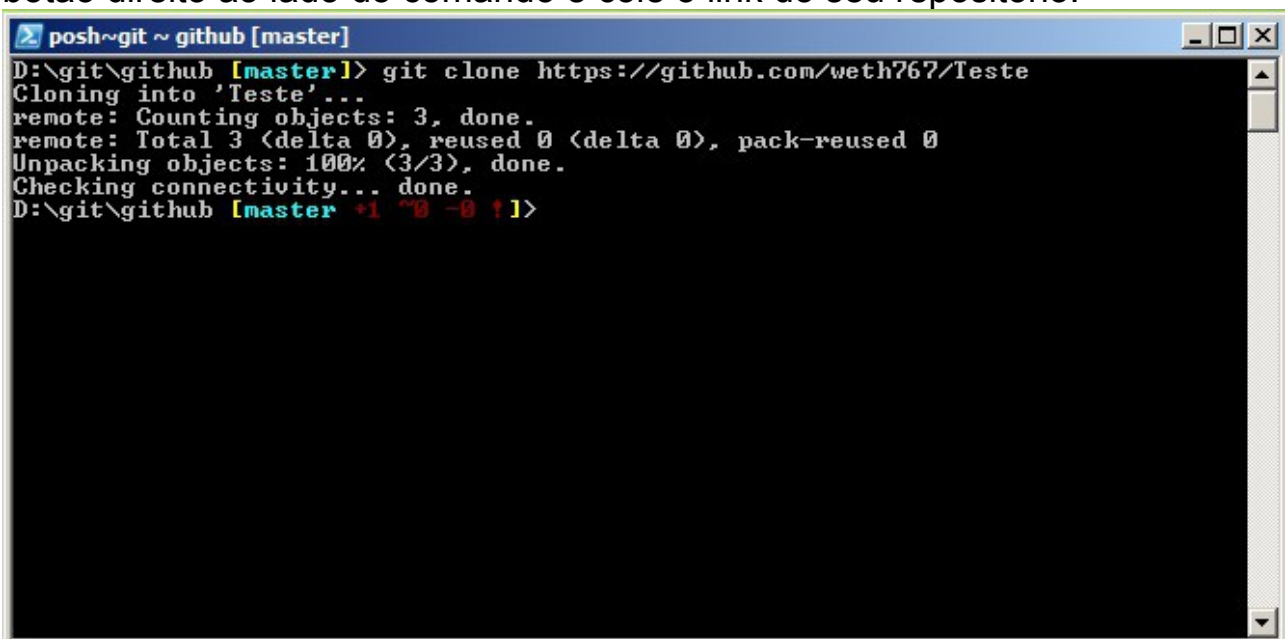
```
posh~git ~ github [master]
D:\git\github> git init
Initialized empty Git repository in D:/git/github/.git/
D:\git\github [master]> _
```

Como mostra na imagem acima, um novo repositório **local** vazio foi inicializado. Eu escolhi coloca-lo em um disco separado, para trabalhar melhor com ele, mas ele tem como padrão colocar em uma pasta .git nos documentos do disco local C, você pode colocar onde achar melhor.

Passo 2: Agora vamos baixar o nosso repositório criado nos passos anteriores, para trabalhar com ele, o nome do comando é git clone e o link do seu repositório.

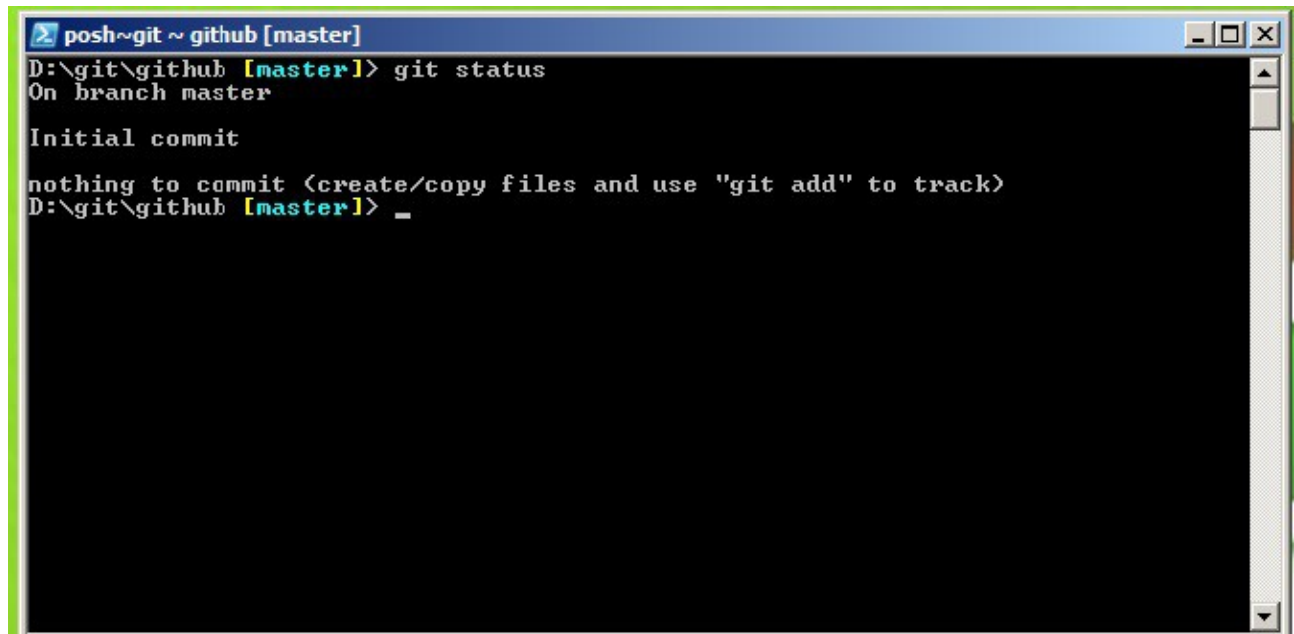


Digite o comando, lembra que pedir para copiar o link, então clique com o botão direito ao lado do comando e cole o link do seu repositório.



Como na imagem acima, copie o link e de ENTER para executar o comando e verá que aparecerá como na imagem. Isso indica que seu repositório foi clonado com sucesso.

Passo 3: Agora usamos o comando: git status para listar todos os arquivos que foram depositados no repositório mostrado na primeira imagem, no meu caso é D:\git\github\.git que é a pasta criada pelo git hub como padrão.

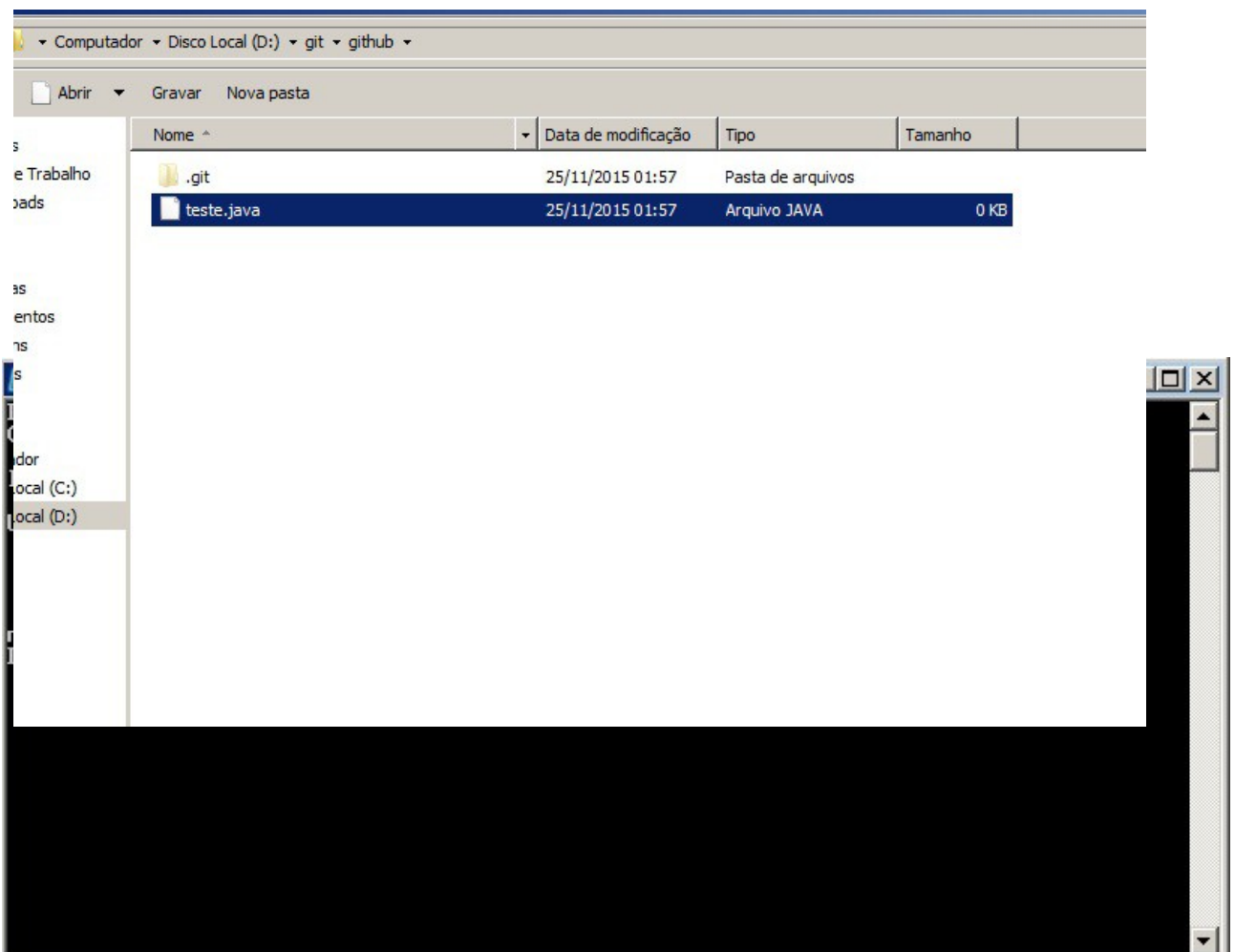
A terminal window titled 'posh~git ~ github [master]' showing the output of the 'git status' command. The output indicates an initial commit on the master branch with no files to track.

```
posh~git ~ github [master]
D:\git\github [master]> git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
D:\git\github [master]> _
```

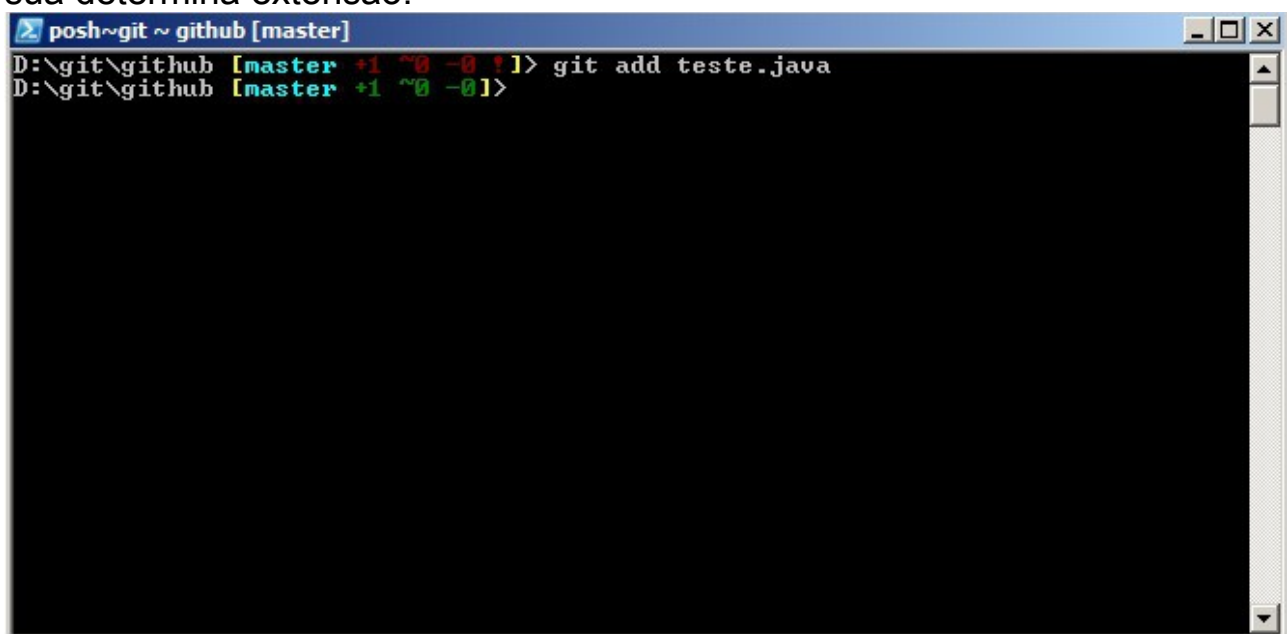
Como mostra na imagem acima não temos arquivos e nem arquivos para serem comitados, commit é um comando do git que será explicado nos passos abaixo.



Como podem ver eu criei um arquivo .java na minha pasta do github , logo após criar o arquivo novamente usaremos o git status para listar os arquivos.

Então como podemos ver, tem um novo arquivo que não foi adicionados no controle do github e nem comitado.

Passo 4: Agora vamos fazer um teste de como adicionamos um arquivo pelo git hub, vamos usar o comando: git add “nomearquivo”.“ext” usamos o git add para adicionar o arquivo logo após o nome do arquivo com sua determina extensão.

A screenshot of a Windows command prompt window. The title bar reads "posh~git ~ github [master]". The command prompt shows the current directory as "D:\git\github" and the branch as "master". The status bar at the bottom of the window indicates a new file has been added: "[master +1 ~0 -0 !]". The user has entered the command "git add teste.java" and the prompt has moved to the next line.

```
posh~git ~ github [master]
D:\git\github [master +1 ~0 -0 !]> git add teste.java
D:\git\github [master +1 ~0 -0 !]>
```

Após adicionarmos o arquivo , podemos ver que a cor mudou para verde , indicando que foi regulamentado um arquivo, agora vamos novamente dar um git status para ver como esta.

```
posh~git ~ github [master]
D:\git\github [master +1 ~0 -0 !]> git add teste.java
D:\git\github [master +1 ~0 -0]> git status
On branch master

Initial commit

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

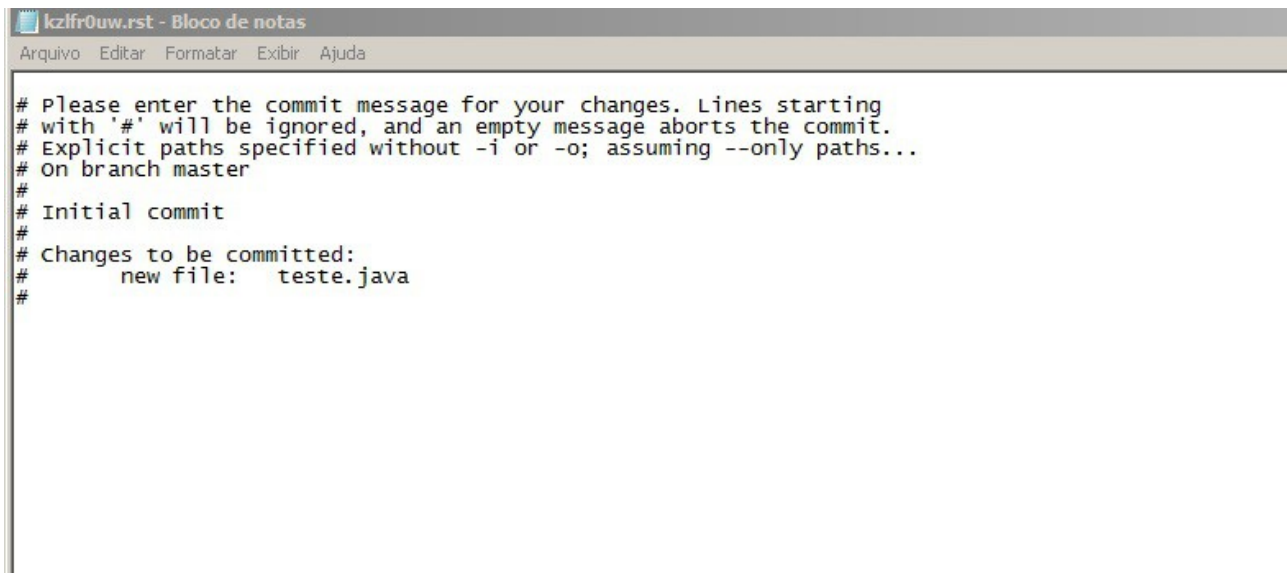
        new file:   teste.java
D:\git\github [master +1 ~0 -0]> _
```

Ali mostrou que o arquivo foi adicionado com sucesso mas ainda sem commit, o que nos leva a explicação do que seria um commit. O commit é o controle do github sobre seus arquivos adicionados, ele seria o passo que deve ser realizado antes de enviar para os servidores do github, então o git usa o commit com uma forma de validação de arquivos.

Passo 5: Então agora usaremos o comando: git commit colocando após o comando o nome do arquivo mais sua extensão para comita-lo

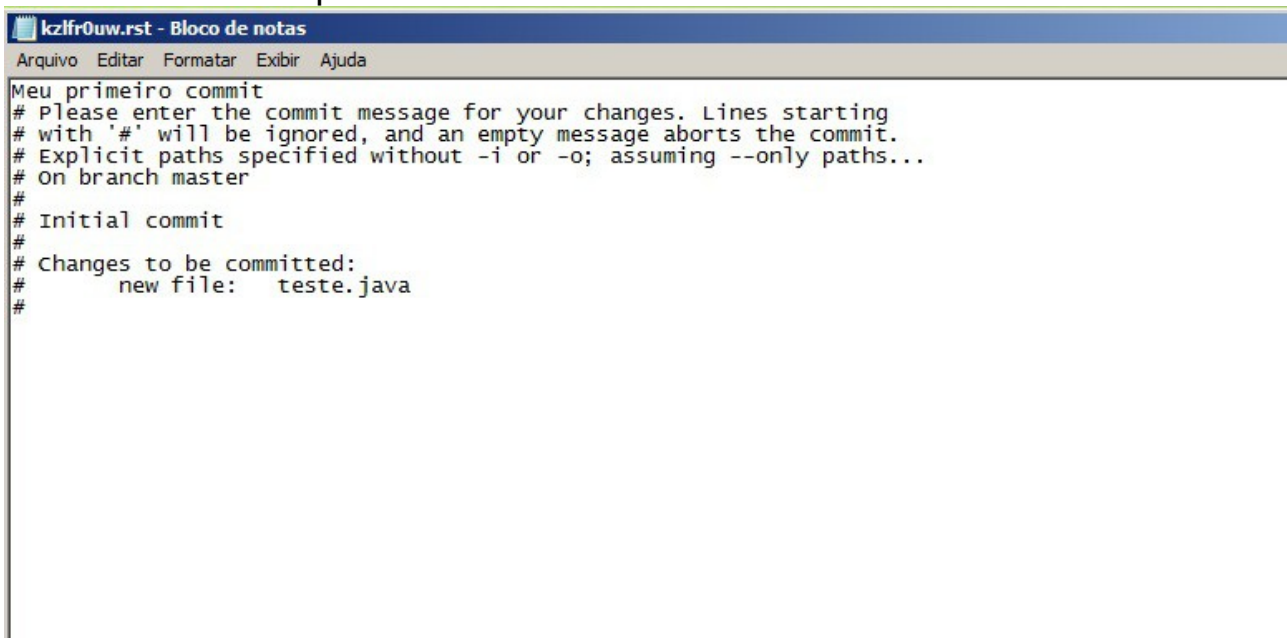
```
posh~git ~ github [master]
D:\git\github [master +1 ~0 -0]> git commit teste.java
```

Então vai aparecer uma telinha, como um bloco de notas, onde será listado todos os itens que você está adicionando e mais algumas informações. Acima disso estará um local para você colocar um comentário sobre esse commit.



```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# Explicit paths specified without -i or -o; assuming --only paths...
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   teste.java
#
```

Vou colocar “ Meu primeiro Commit “ .



```
Meu primeiro commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# Explicit paths specified without -i or -o; assuming --only paths...
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   teste.java
#
```

Vou no campo arquivo , logo depois em salvar e pode fechar o arquivo.
Depois de commitado, o master do git mudará para cor verde, ou seja ele
está a espera do seu push, que será o próximo passo.



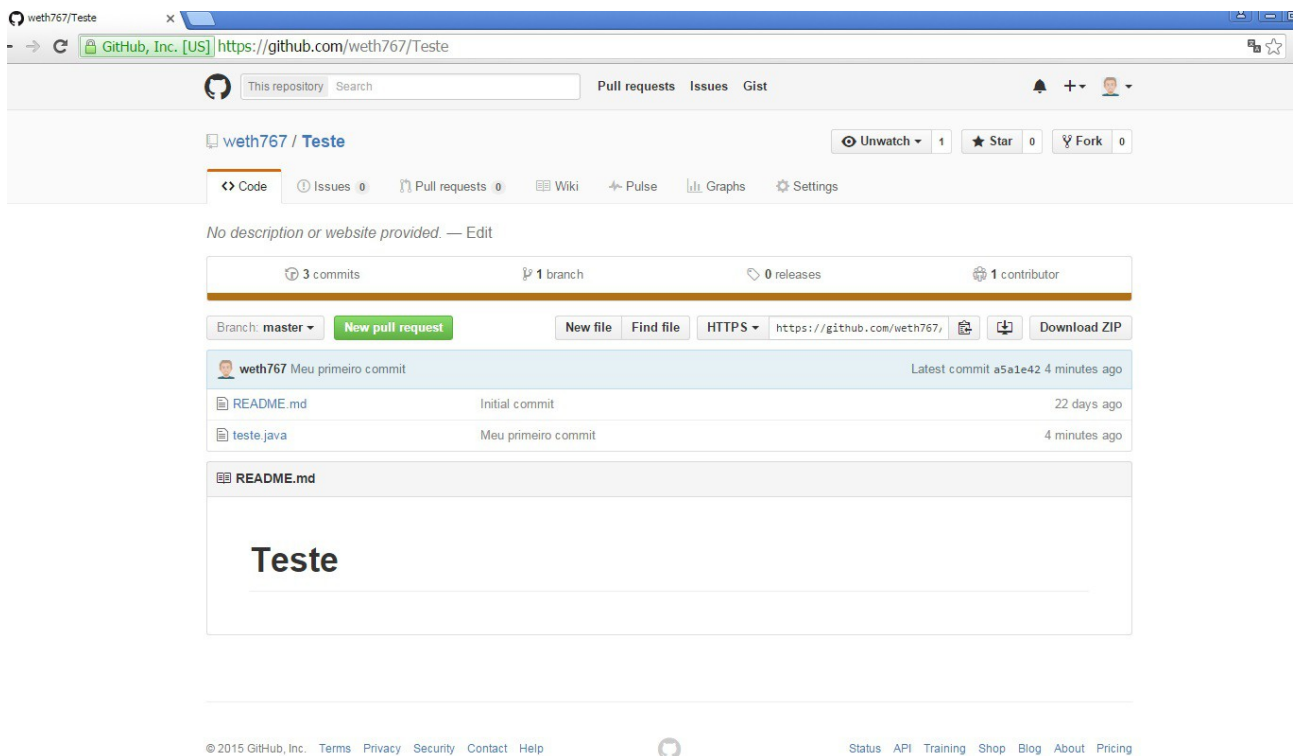
```
posh~git ~ teste [master]
C:\Users\IFPR\desktop\teste [master]>
```

Passo 6: E finalmente, agora usaremos o comando : git push, este comando enviará todas as nossas mudanças em nosso código para o servidor git, como é nosso primeiro commit só terá ele no repositório.



```
posh~git ~ teste [master]
C:\Users\IFPR\desktop\teste [master]> git push
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 552 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/weth767/Teste
   ea0f926..a5a1e42  master -> master
C:\Users\IFPR\desktop\teste [master]> _
```

Agora vamos conferir nosso repositório, se o arquivo realmente foi enviado.



Muito bem, obrigado por ler meu tutorial sobre o github, nele contém as informações básicas para você usar o github, se houver alguma dúvida, entre em contato pelo meu e-mail: joao_paulo767@hotmail.com , e continue com os commits.