

1369 dias atrás em [GIT](#) [GITHUB](#) [OPEN SOURCE](#)

Criando e enviando arquivos para seu repositório no Github.

Antes de mais nada: se tem dúvida sobre as diferenças entre Git e Github, dá uma conferida [nesse vídeo](#).

No [primeiro](#) post sobre Git e Github, ensinei a fazer a parte de configuração e instalação.

No [segundo](#), mostrei como fazer o *fork* de um repositório existente e baixar ele pra sua máquina.

Agora, chegou a hora de aprender a **criar** seu próprio repositório e interagir com ele.

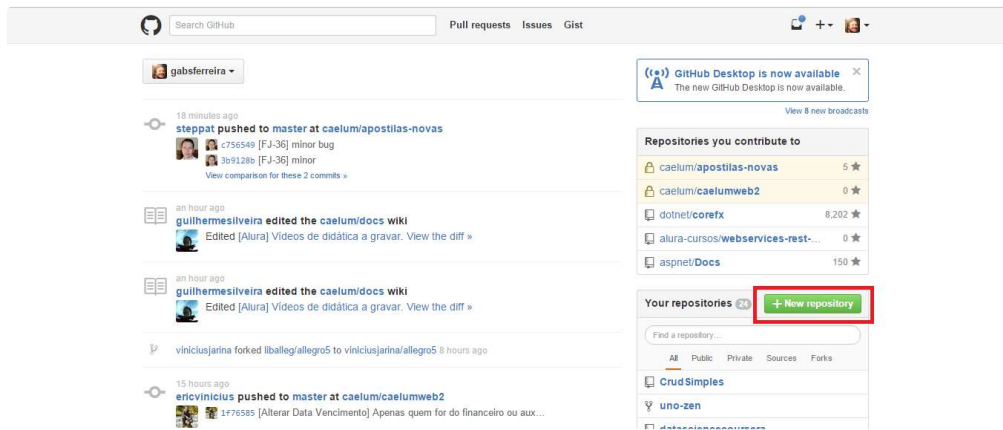
Se gostar desse post, não deixe de [me seguir no Twitter](#) :)

Geralmente as pessoas criam repositórios no Github para hospedar seus projetos pessoais. E esses projetos não precisam ser necessariamente código.

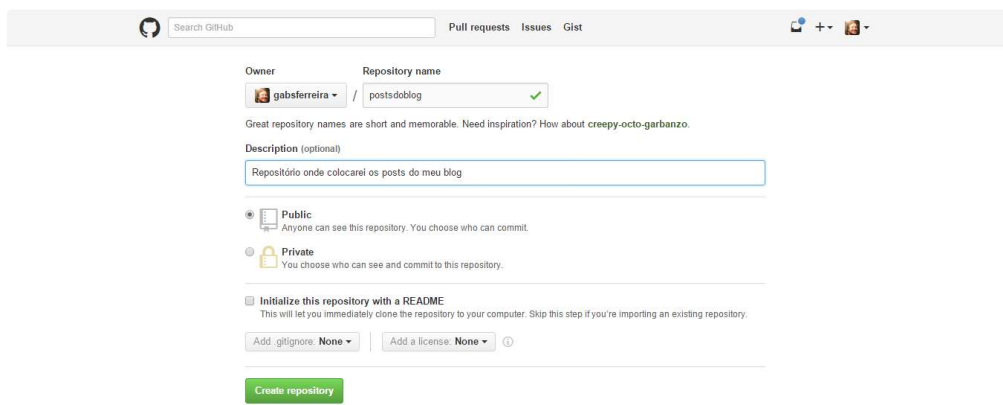
Como exemplo nesse post, eu vou criar um repositório para guardar os posts do meu blog. Os comandos que vou usar são **os mesmos** pra qualquer tipo de projeto/arquivo.

Criando o repositório

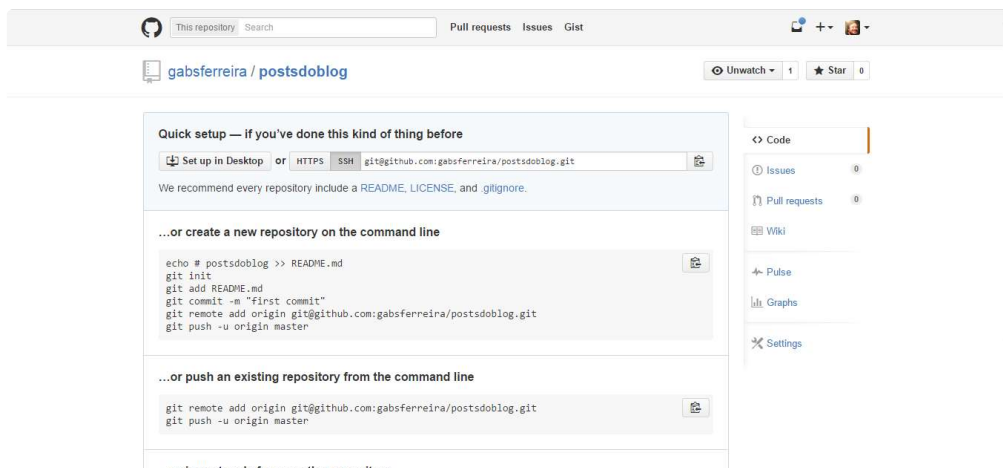
Abra o [Github](#), faça login com a sua conta e clique em *New repository*.



Em seguida, basta colocar o nome e descrição do repositório que você está criando e clicar em *Create repository*.



Depois, aparecerá essa tela:



O repositório foi criado, só que ainda está vazio.

Vamos enviar nossos arquivos para lá!

Inicializando um repositório na sua máquina



Abra o Git Bash e vá até a pasta onde está o seu projeto.

Eu no caso vou até a minha pasta *Posts*, onde tenho um post meu salvo no formato *txt*.

```
MINGW32/c/Users/Gabriel/Posts
Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Gabriel@GABRIEL ~
$ cd Posts/

Gabriel@GABRIEL ~/Posts
$ ls
Post.txt

Gabriel@GABRIEL ~/Posts
$
```

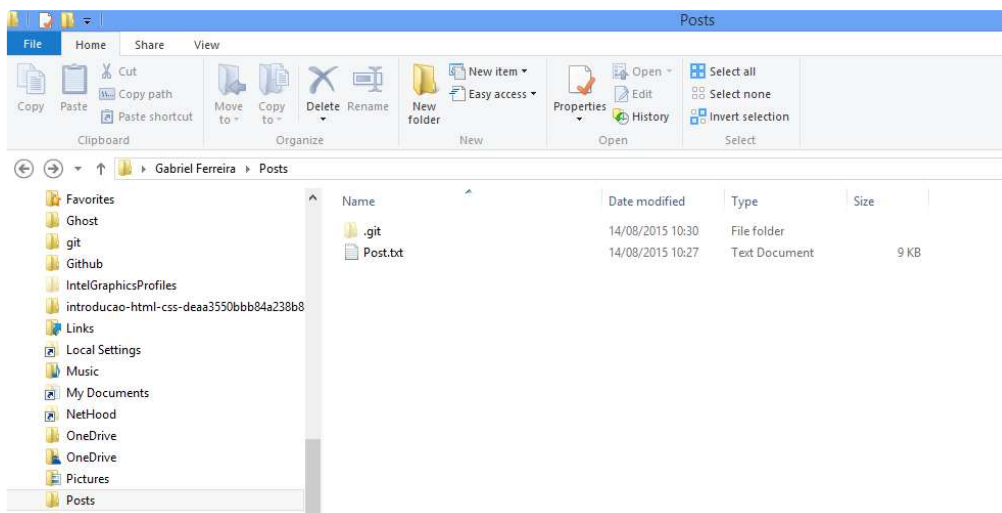
Agora, quero transformar essa minha pasta em um **repositório Git**.

Pra isso, basta digitar **git init** e dar enter:

```
$ git init
Initialized empty Git repository in c:/Users/Gabriel/Posts/.git/
```

Ele está falando que iniciou um repositório vazio nessa pasta. Como eu sei que agora tenho um repositório nessa pasta?

Se você abrir a pasta no *File Explorer* e configurar ele para mostrar os arquivos ocultos, verá que tem uma pasta a mais chamada **.git**:





O Git usa essa pasta pra controlar as alterações feitas no seu repositório.

Ok, agora temos um repositório na sua máquina e um no Github...como **sincronizar** os dois?

Adicionando uma origin

Vamos agora executar esse comando:

```
git remote add origin git@github.com:gabsferreira/postsdoblog.git
```

Basicamente o que estamos dizendo nessa linha de comando é: "Git, esse meu repositório **local** se conectará com um **remoto**, o *caminho dele(origin)* é `git@github.com:gabsferreira/postsdoblog.git`. Estabeleça essa comunicação pra mim".

Se você abrir aquela pasta **.git** e abrir o arquivo chamado **config** lá dentro, verá a seguinte linha:

```
[remote "origin"]
  url = git@github.com:gabsferreira/postsdoblog.git
  fetch = +refs/heads/*:refs/remotes/origin/*
```

É onde está configurada nossa **origin**.

Enviando alterações para o Github

Quando estou dentro de uma pasta que é um repositório Git e quero saber o que eu fiz de alterações, eu digito o seguinte comando:

```
git status
```



Esse comando é um dos que você **mais** vai usar no Git.

Ele irá mostrar:

```
On branch master

Initial commit

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

        Post.txt

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

Ele está dizendo aí que o arquivo *Post.txt* está *untracked*, ou seja, não está sendo **monitorado/gerenciado** pelo Git. Quero que ele seja!

Então:

```
git add Post.txt
```

Com esse comando, eu **adicionei** o meu arquivo ao Git. Se eu der *git status* de novo:

```
$ git status
On branch master

Initial commit

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

        new file:   Post.txt
```

Agora ele me mostra que o arquivo está para **ser comitado**. Ou seja, ele ainda não guardou o estado do arquivo **permanentemente**, é só temporário.

Se eu quiser que seja permanente, uso o *git commit*.

```
git commit -m "aqui coloco uma mensagem, tipo: comitando primeiro post"
```

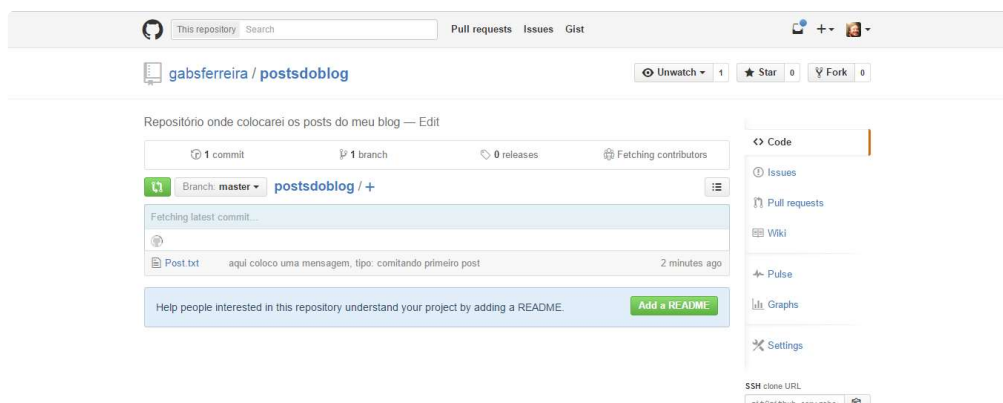


Beleza, comitado.

Agora quero enviar isso para o meu repositório remoto:

```
git push -u origin master
```

Se você for até o seu repositório no Github agora, voilá:



Legal né? (:

No próximo post sobre o assunto eu pretendo falar um pouco mais sobre o fluxo do Git.

Gostou desse post?

Confira outros que acho que também vai gostar:

[Calma: fazer as coisas bem leva tempo](#)

[Quer ficar bom em algo? Faça uma coisa de cada vez](#)

[Se acha muito velho pra aprender a programar? Pense de novo](#)

[Cuidado com ofertas de trabalho que focam no que não é importante](#)

[4 dicas para ser chamado para entrevistas](#)

[7 podcasts de programação e tecnologia que você tem que ouvir](#)

[Já se sentiu um impostor? Eu também](#)

[O guia do programador iniciante: como ter um portfólio nunca tendo trabalhado com programação](#)

Quer ficar em dia com os meus posts e novidades? Se inscreva na minha newsletter, me [segue no Twitter](#) e na [minha página](#)!