

# Tutorial para iniciar com o Git e o GitHub

Aprenda a trabalhar com Git e github, artigo introdutório bastante prático e que promete reduzir a curva de aprendizado da ferramenta.

---

Esta é uma introdução prática ao Git e GitHub, indicado para quem está dando os primeiros passos com o Git. Neste artigo nós veremos como:

- Criar um repositório localmente
- Adicionar (add) e comitar (commit)
- Criar um repositório no GitHub
- Conectar o repositório local com o repositório público
- Sincronizar ambos os repositórios

Antes de tudo, é preciso saber que Git não é GitHub.

[Git]() é um sistema distribuído de controle de versão de código.

[GitHub]() é o local onde você hospeda o seu Git na web.



Você não está preso ao GitHub, existe uma infinidade de outros "hospedeiros", alguns pagos outros gratuitos. O GitHub é gratuito para projetos abertos e pago para projetos privados.

Antes de iniciar com o Git, seria bom se você entendesse o conceito de **versionamento de código**, mas talvez você esteja justamente experimentando o Git para poder entender o que é versionamento, então vamos lá...

## Criando um repositório local

Antes de criar o repositório você deve criar uma pasta para ele, obviamente.

Para criar uma nova pasta no Linux executamos `mkdir nome-da-pasta`.

Você entra nela, ou melhor, vai até ela `cd nome-da-pasta`.

Agora é só criar o repositório

```
git init
```

Para testar, pergunte ao Git sobre o status do repositório `git status`.

O resultado deve ser parecido com isto...

```
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
```

"Nada para comitar"... bom, então vamos inserir um arquivo e "comitar".

## Adicionado e comitando

Considerando que você já criou o seu repositório Git e está na pasta do repositório.

Você deve criar um arquivo qualquer, utilize seu editor de texto ou então utilize o comando seguinte.

```
touch readme.md
```

O comando `touch` é um utilitário do Linux.

Antes de comitar, devemos adicionar, a sequência é sempre esta:

- Adicionar ( `add` )
- Comitar ( `commit -m "mensagem"` )

Veja os comandos:

```
git add readme.md  
git commit -m "my first commit"
```

## Criando um repositório no GitHub

Primeiro, crie uma conta no GitHub.

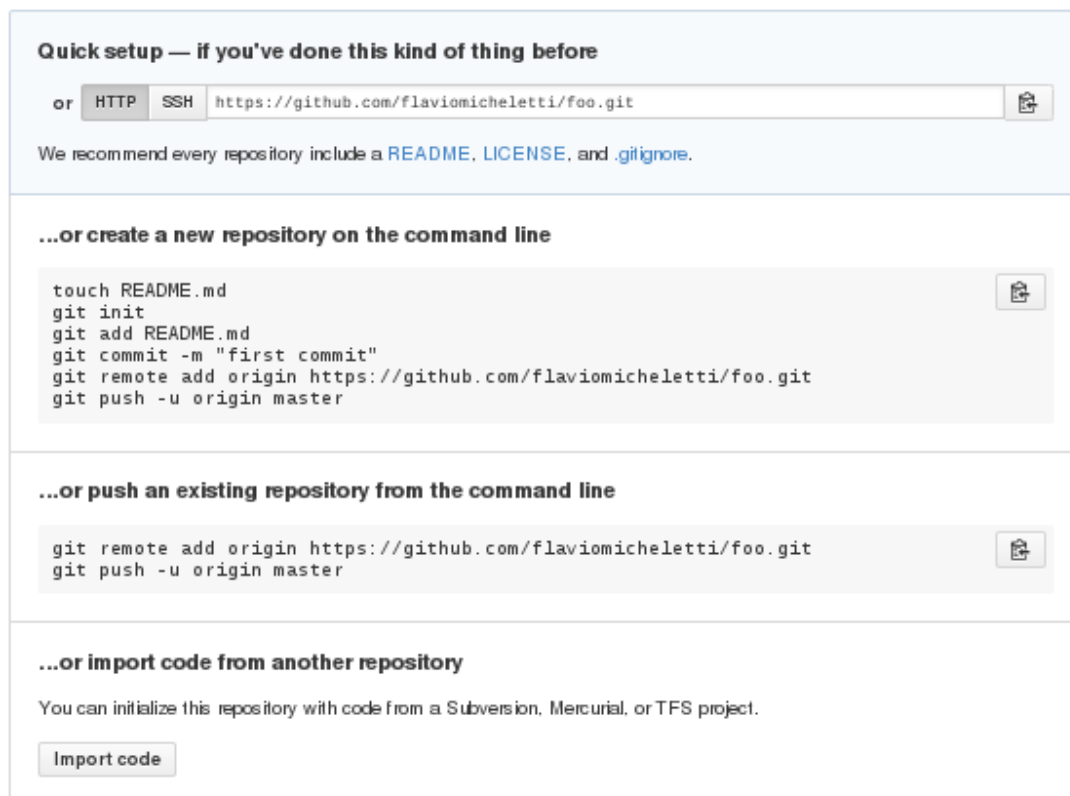
Utilize um nome de guerra que transmita seriedade, ele será visto por todos, inclusive seu futuro empregador.

Agora procure o ícone com um sinal de mais (+) chamado "Create new...".

Defina um nome para seu novo repositório.

Por enquanto, ignore as outras opções. Depois, com calma, você mesmo aprenderá sobre elas.

Como seu repositório está vazio, o GitHub lhe dará algumas dicas, sua tela deve parecer-se com esta:



**Quick setup — if you've done this kind of thing before**

or ☐ HTTP ☒ SSH

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

---

**...or create a new repository on the command line**

```
touch README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin https://github.com/flaviomicheletti/foo.git  
git push -u origin master
```

---

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/flaviomicheletti/foo.git  
git push -u origin master
```

---

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Apenas anote a URL exibida no campo **HTTP**, vamos precisar dela daqui a pouco.

Agora, temos um repositório no GitHub e um localmente, então vamos conectá-los.

## Conectando o repositório local com o da web

O seu QG (quartel general) é o seu repositório local. A partir dele, você enviará ou receberá informações do repositório na web (no GitHub).

Para "conectar" os repositórios executamos o comando abaixo.

```
git remote add origin https://github.com/flaviomicheletti/github-foo.git (http://github.com/flaviomicheletti/github-foo.git)
```

Onde **origin** signifca um apelido para seu repositório, poderia ser qualquer outro nome.

E no lugar da minha URL você deve utilizar a sua URL, eu avisei que precisaríamos dela!

Para testar, execute `git remote -v`, sua tela deve ser parecida com a seguinte:

```
origin https://github.com/flaviomicheletti/github-foo.git (http://github.com/flaviomicheletti/github-foo.git) (fetch)
origin https://github.com/flaviomicheletti/github-foo.git (http://github.com/flaviomicheletti/github-foo.git) (push)
```

## Sincronizando os repositórios

Você já criou um repositório local e outro no GitHub.

Já adicionou alguns arquivos e "comitou" algumas modificações (localmente).

O seu repositório local já está conectado com o da web (possui uma referência para ele).

Agora só falta enviar as informações do repositório local para o repositório na web (no GitHub):

```
git push origin master
```

Lembrando que **origin** é o apelido para seu repositório na web e **master** é o seu branch principal.

Se você acabou de instalar o Git, talvez tenha se esquecido das configurações globais.

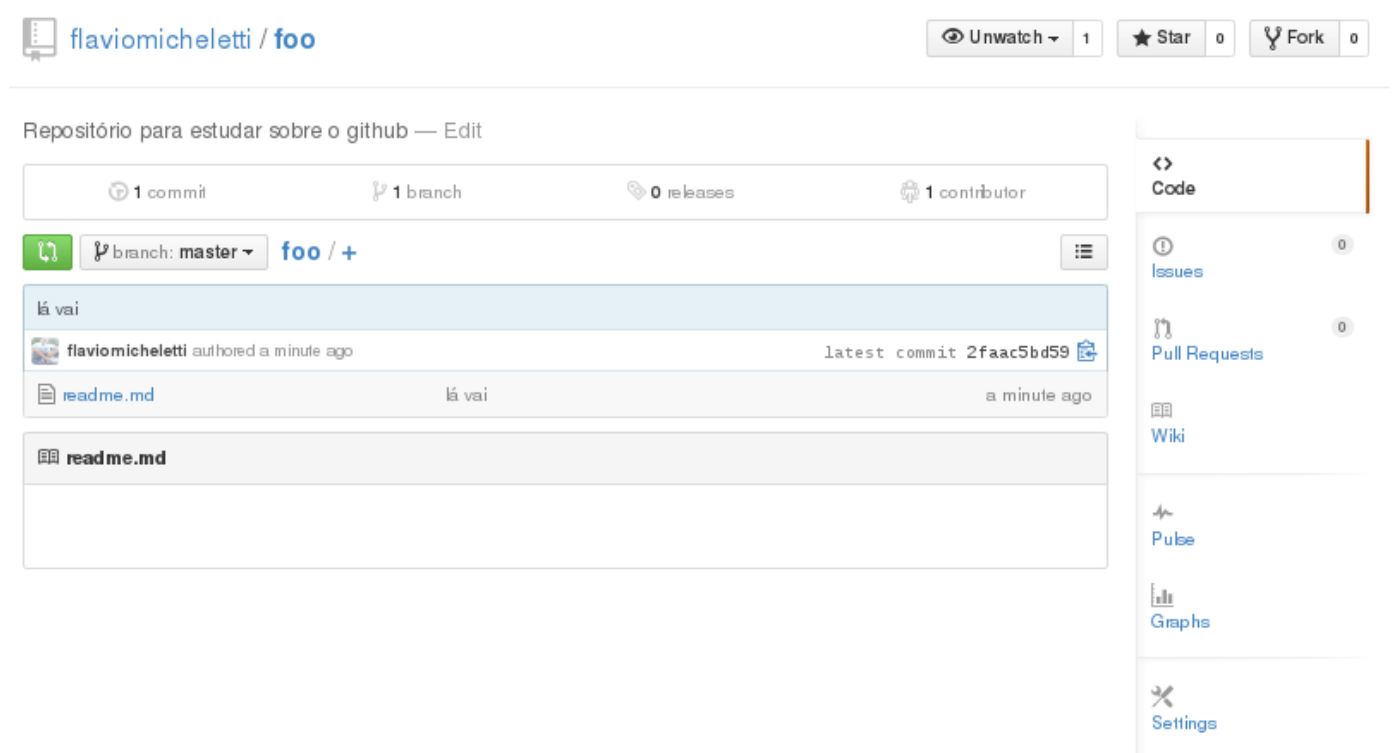
Configure seu email e nome que serão exibidos em cada commit.

```
git config --global user.name "Seu nome para exibição"
git config --global user.email "seu-email@email.com"
```

Você deve configurar antes de realizar o **push**.

## Fechando a conta

Se tudo deu certo, acesse novamente (ou atualize) a página de seu repositório, você deve estar vendo algo parecido com a figura abaixo.



Este repositório (<https://github.com/devfuria/foo>) existe de verdade!

Eu criei ele para você poder treinar, bagunçar, testar, brincar, fazer o que você quiser com ele, sem a preocupação de acertar.

Só não pode não fazer nada, então vamos praticar....

Faça um **fork** do projeto, crie um arquivo de texto qualquer com alguma mensagem tipo "oi, me chamo... e passei por aqui" e envie através de um **pull request**.

Navegue nesta série!



Como é trabalhar com Git - par



(/git/como-e-trabalhar-com-git-1/)

## Comentários

8 Comentários devfuria

 Entrar ▾

 Recomendar 2

 Tweet

 Compartilhar

Ordenar por Mais votados ▾

Participe da discussão...

**Lucas Duarte** • 3 anos atrás

Obrigado, me ajudou muito!

1 ^ | v • Responder • Compartilhar ›

**Flávio** Mod ➔ Lucas Duarte • 3 anos atrás

Oi Lucas, convido vc a deixar uma marca (escreve algo) no rodapé do repositório!

Valeu!

1 ^ | v • Responder • Compartilhar ›

**Eder Hazard** • 4 meses atrás

Simples e objetivo.

Muito obrigado!

^ | v • Responder • Compartilhar ›

**Romildo Júnior** • 5 meses atrás

Simples e objetivo... Parabéns pelo post!

^ | v • Responder • Compartilhar ›

**Romildo Júnior** ➔ Romildo Júnior • 5 meses atrás

Tentei enviar o comentário no arquivo porém está sem acesso para push... Permissão negada!

^ | v • Responder • Compartilhar ›

**Flávio** Mod ➔ Romildo Júnior • 5 meses atrás

Oi Romildo, é porque vc deve ter clonado o repositório antes de fazer o fork.

Então, faça um fork primeiro e clone o "seu" repositório, na sequencia, faça um pull request através da interface do github.

Abs.

^ | v • Responder • Compartilhar ›

**Romildo Júnior** ➔ Flávio • 5 meses atrás

Muito Obrigado.

Agora sim, tudo certo...

Aceita minha pull request ai... :)

^ | v • Responder • Compartilhar ›

**Flávio** Mod ➔ Romildo Júnior

• 5 meses atrás

Ôpa, pull request realizado!

^ | v • Responder • Compartilhar ›

## Instalando o nodejs

2 comentários • 3 anos atrás



**Flávio** — Obrigado a vc pela leitura Itamar! Em qual distro vc instalou o node? Vc vai estudar

## Sintaxe básica de PHP

1 comentário • 2 anos atrás



**wevertom melo** — Na parte de laços está errado. Pois \$i está com > e 0 sempre é menor que

## SQL - Structured Query

## Instalando o Python 3 (junto

(<https://github.com/flaviomicheletti/devfuria.com.br>)

Esta obra está licenciada sob a Creative Commons Atribuição 4.0 Internacional  
(<http://creativecommons.org/licenses/by/4.0/>).

[www.devfuria.com.br](http://www.devfuria.com.br) (/) (desde 2012)