



GUIA PESSOAL GIT/GITHUB

Anderson Job Loeffler

git init

Para iniciarmos um repositório local no git, primeiro devemos entrar na pasta do projeto e depois rodar o comando

```
~$ git init
```



exemplo

```
~$ cd praticas-git  
~/praticas-git$
```

```
~/praticas-git$ git init  
Repositório vazio Git inicializado em /home/user/praticas-git/.git/
```

github repo

Para a criação do repositório remoto, devemos acessar o github.com, em 'repositories' clicar em 'new'

Logo após, devemos configurar:

- ★ O nome do repositório,
- ★ A descrição
- ★ Visibilidade
- ★ Arquivo 'leia-me'
- ★ Um arquivo que ignora evio de arquivos do repositório (gitignore)
- ★ Tipo de licença



exemplo



aloefflerj ▾

Repositories



New


Find a repository...




exemplo

Owner *

Repository name *

 aloefflerj


 /


hello-world 

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-memory](#)?

Description (optional)

Repositório de programa que imprime 'Hello world' em q

☒  **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.](#)

Create repository

github repo

Em seguida, o github vai pedir para que iniciemos um repositório para podermos subir para o remoto do github

Como já possuímos um repositório local, basta executarmos no terminal na pasta do projeto:

```
~$ git remote add origin <repo-name>
```



exemplo

```
~/praticas-git$ git remote add origin  
git@github.com:aloefflerj/hello-world.git
```


git add

Agora, devemos adicionar nossas alterações do projeto

Para isso usamos

```
~$ git add <filename>
```

Caso queiramos adicionar todos os arquivos alterados, podemos usar:

```
~$ git add .
```



exemplo

```
~/praticas-git$ git add .
```

git status

Para vermos o status das alterações do nosso git, podemos rodar o comando:

```
~$ git status
```

Como acabamos de adicionar arquivos com o comando 'add', o status nos mostrará os arquivos adicionados



exemplo

```
~/praticas-git$ git status
```

```
No ramo master
```

```
No commits yet
```

```
Mudanças a serem submetidas:
```

```
(utilize "git rm --cached <arquivo>..." para não apresentar)
```

```
new file: hello-world
```

```
new file: hello-world.c
```

git commit

Para 'comitarmos' nossos arquivos adicionados, rodamos o comando:

```
~$ git commit <filename>
```

O editor padrão do git abrirá para que uma mensagem de commit seja inserida

Podemos também rodar o comando direto com a mensagem da seguinte forma:

```
~$ git commit <filename> -m "mensagem"
```



exemplo

```
~/praticas-git$ git commit -m "meu primeiro commit"
```



exemplo

Agora que adicionamos o commit, podemos rodar novamente o 'status'

```
~/praticos-git$ git status  
[master (root-commit) 5ac08ab] primeiro commit  
2 files changed, 8 insertions(+)  
create mode 100755 hello-world  
create mode 100644 hello-world.c
```

git log

Com nosso primeiro commit já feito,
podemos ver o histórico com:

```
~$ git log
```




exemplo

```
~/praticas-git$ git log  
commit 5ac08abe8b226b898225f7df9bfbb2cc4826a179 (HEAD -> main)  
Author: Anderson Job Loeffler <aloefflerj@gmail.com>  
Date: Thu Nov 4 21:27:37 2021 -0300
```

primeiro commit

git branch

Para criarmos uma branch no git, usamos o comando:

```
~$ git branch <branch-name>
```

Já que o github trabalha com a principal branch chamada 'main', e precisamos subir o projeto para o remoto, rodamos o seguinte comando para criarmos uma branch principal

```
~$ git branch -M main
```



exemplo

```
~/praticas-git$ git branch -M main
```

Se rodarmos o comando sem parâmetro nenhum, o git nos lista as branches e em qual estamos trabalhando

```
~/praticas-git$ git branch
bug
* main
```

git push

Agora que temos tudo configurado para subirmos o repositório para o remoto no github, rodamos o seguinte comando:

```
~$ git push -u origin <branch-name>
```

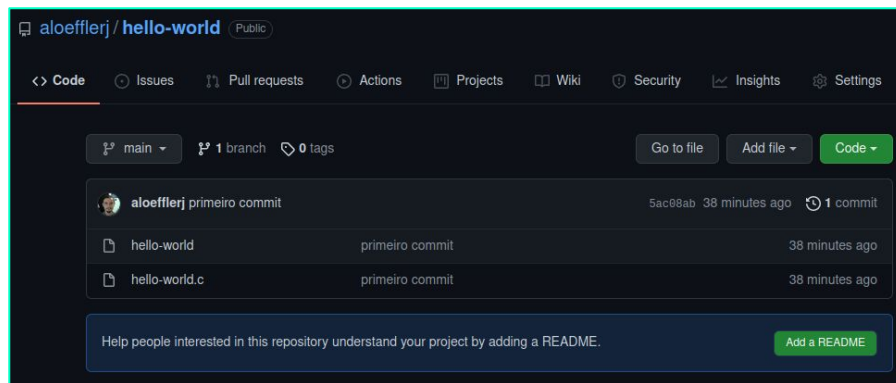


exemplo

```
~/proticos-git$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.78 KiB | 1.39 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To github.com:aloefflerj/hello-world.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

git push

Se conferirmos agora o repositório do github, teremos nosso projeto no repositório remoto:



git README



Vamos adicionar agora, um arquivo de README para descrevermos nosso projeto



Para isso, podemos acessar o repositório no github e clicar em 'add README'

Devemos então adicionar a informação e fazer o commit



exemplo

 **aloefflerj** primeiro commit 5ac08ab 41 minutes ago  **1** commit

 hello-world	primeiro commit	41 minutes ago
 hello-world.c	primeiro commit	41 minutes ago


Help people interested in this repository understand your project by adding a README.

[Add a README](#)

hello-world / README.md

in main

<> Edit new file

 Preview

```
1 # Hello World|
2 Repositório de programa que imprime 'Hello world!' em c
3
```




exemplo

Commit new file

Adição do arquivo README

Add an optional extended description...

☒ Commit directly to the `main` branch.

☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file

Cancel

git pull

Agora que temos uma alteração no repositório remoto, devemos puxá-la para nosso repositório local

Para isso rodamos o seguinte comando

```
~$ git pull
```



exemplo

```
~/praticas-git$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 772 bytes | 772.00 KiB/s, done.
From github.com:aloefflerj/hello-world
   5ac08ab..0d80503  main      -> origin/main
Updating 5ac08ab..0d80503
Fast-forward
 README.md | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 README.md
```

git merge

Para fundir duas branches, devemos entrar na branch que queremos receber e rodamos o comando:

```
~$ git merge <branch-name>
```



exemplo

Primeiro criamos uma nova branch

```
~/praticas-git$ git branch alteracao
```

Entramos então na branch criada com o comando

```
~/praticas-git$ git checkout alteracao
```



exemplo

Fazemos então uma alteração no projeto, adicionamos a alteração e a comitamos. Após, voltamos para branch 'main'

```
~/praticas-git$ git checkout main
```

Agora podemos então rodar o comando merge

```
~/praticas-git$ git merge alteracao
Updating 0d80503..6cee28d
Fast-forward
 hello-world.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

github pull request

Podemos ainda fazer o merge direto pelo github através do 'pull request'

Primeiro devemos fazer uma alteração em uma branch, adicionar, comitar e dar um push

Vamos até o github e podemos ver que ele já identifica 'pushs' recentes

Clicamos então em 'Compare & pull request'



exemplo

Podemos entrar na branch 'alteracao' e rodar o seguinte comando para subirmos para o remoto

```
~/praticas-git$ git push -u origin alteracao
```

No github, podemos clicar em 'Compare & pull request'

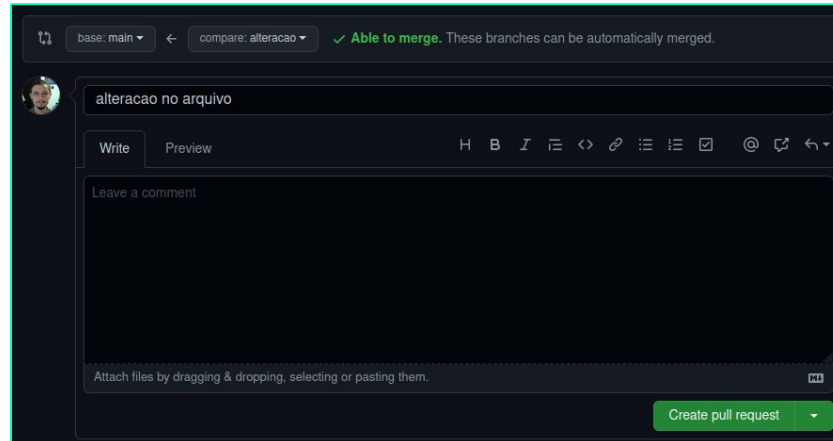


alteracao had recent pushes less than a minute ago

Compare & pull request

github pull request

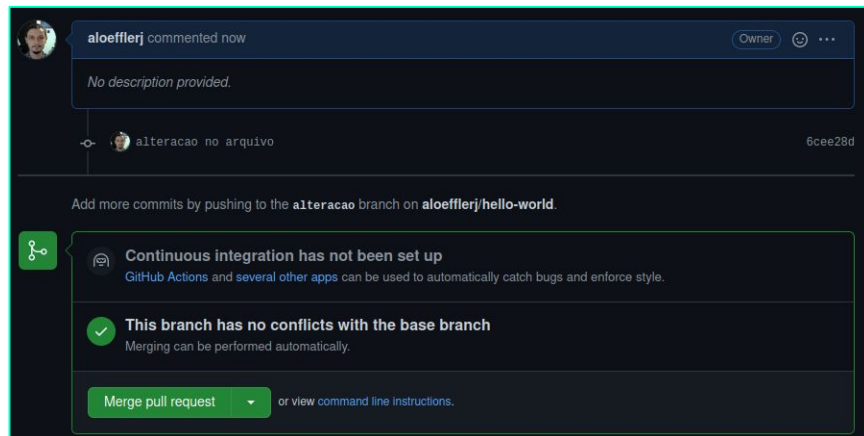
Caso não haja nenhum conflito,
podemos fazer o pull request
adicionando o commit e clicando em
'Create pull request'





exemplo

O github vai verificar então se existe algum conflito entre as branches, caso não haja, podemos clicar em 'Merge pull request' e 'Confirm merge'



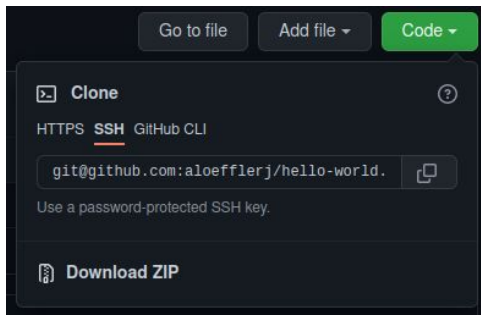
github clone

Podemos clonar um repositório do github com o comando:

```
~$ git clone <repo-address>
```

O endereço de clonagem está na aba 'code' no repositório (podemos clonar pelo protocolo http ou ssh)

Podemos também, simplesmente, baixar um .zip





exemplo

```
~/ $ git clone git clone git@github.com:aloefflerj/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 3), reused 7 (delta 2), pack-reused 0
Receiving objects: 100% (11/11), 4.24 KiB | 4.24 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

OBRIGADO

Práticas de Engenharia de Software

Anderson Job Loeffler
Matrícula 202022605

Porto Alegre, 4 de novembro de 2021

