

## GUIA DE UTILIZAÇÃO GITHUB / GIT



Marcelo Abbi

Práticas de Engenharia de Software

2021/2

## Requisitos

Alguns requisitos simples são necessários para a utilização do GitHub e GIT

- ❖ Computador com acesso à Internet;
- ❖ Sistemas operacionais compatíveis: Windows, Linux e MacOS;
- ❖ Download e Instalação GIT SCM;
- ❖ Conta GitHub.

## Criação de Repositórios

Um repositório do GIT é onde os projetos ficarão armazenados virtualmente. Ele serve para salvar versões do código e acessá-los quando necessário.

A criação do repositório no GitHub é feita no botão **“New”**, na página **“Your Repositories”**. Após isso, a página de criação irá se abrir, e é preciso escolher um nome para o repositório.

Repositories

New

Find a repository...

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner \*


Repository name \*


MarceloAbbi

 /

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

Description (optional)

☒  **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.

No GIT, a criação é feita acessando o GIT no seu computador, e ele ficará armazenado localmente no seu computador.

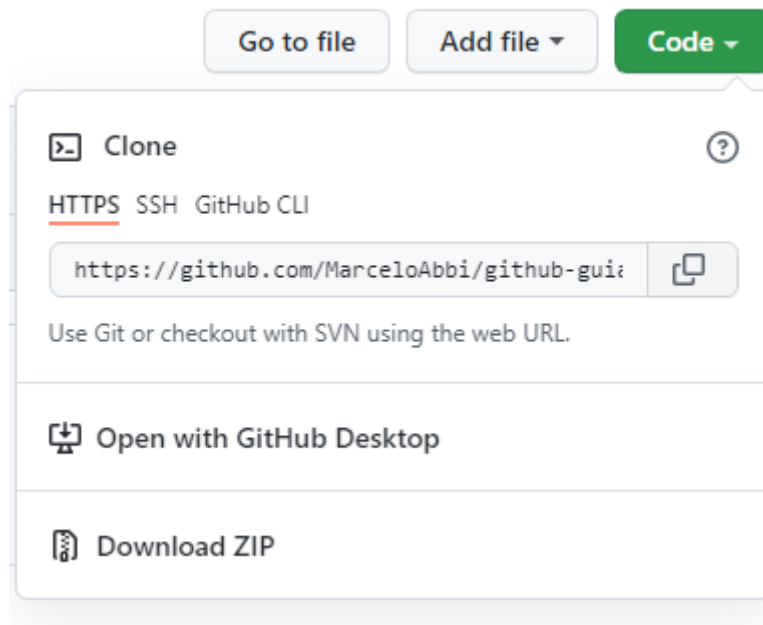
Após aberto, digitar ***git init*** “***nome do projeto***” irá criar um novo repositório no seu computador:

```
Marcelo@DESKTOP-1C2AIP8 MINGW64 ~ (master)
$ git init novo_repositorio
Initialized empty Git repository in C:/Users/Marcelo/novo_repositorio/.git/
Marcelo@DESKTOP-1C2AIP8 MINGW64 ~ (master)
$
```

Para acessar o repositório, é só abrir o diretório em que o repositório foi criado.

## Clonagem

Como o nome já diz, a clonagem serve para que você possa clonar um repositório seu ou de terceiros, localmente para a sua máquina, a fim de fazer modificações em qualquer parte dele, servindo para testar, arrumar, melhorar e acrescentar modificações no projeto e nos códigos. A clonagem é feita com o comando “***git clone***” seguido do link criado no GitHub, e o repositório também pode ser baixado zipado.



(Janela de clonagem GitHub)

```

Marcelo@DESKTOP-1C2AIP8 MINGW64 ~ (master)
$ git clone https://github.com/MarceloAbbi/github-guia.git
Cloning into 'github-guia'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

```

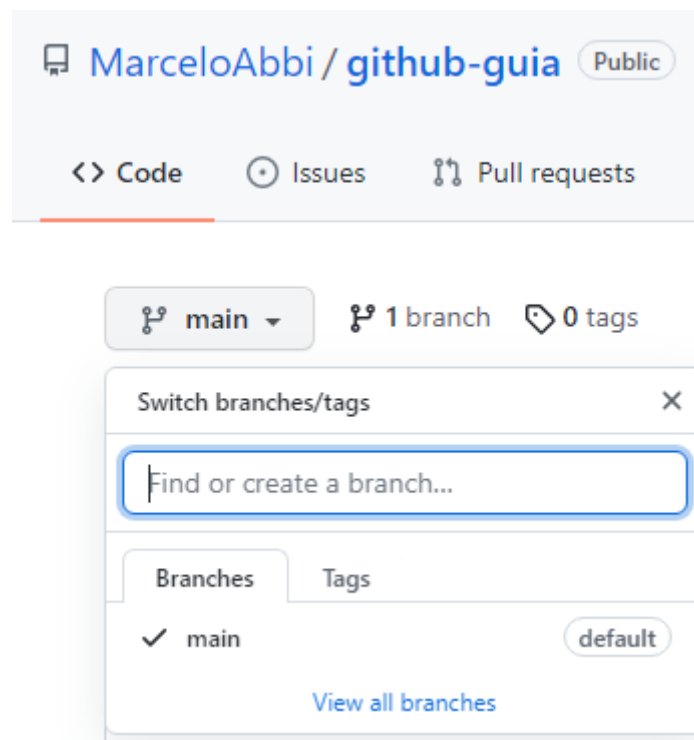
(Clonagem de um repositório criado)

## Criação de Branches

As Branches são as ramificações que o repositório possuirá. Normalmente os projetos só possuem uma Branch Master, que é onde costuma ficar tudo aquilo já testado e construído de forma correta.

No decorrer do desenvolvimento de um repositório, inúmeras branches podem ser criadas. Sejam elas para separar assuntos ou códigos, ou dividir por categoria. Também podem ser utilizadas para dividir tarefas entre membros de um grupo que necessita criar algo junto, porém, cada um pode criar uma branch e deixar o projeto dividido com a tarefa de cada indivíduo.

Para trocar ou criar uma nova Branch no seu repositório no GitHub, é necessário ir até a aba <> Code e selecionar uma nova Branch ou digitar um nome para criar uma nova.



(Local de criação de Branch GitHub)

## Commits

Commits são as atualizações e incorporações do seu git. Basicamente, tudo que for adicionado no repositório será feito através de um commit.

Após adicionar os arquivos que você necessita colocar no seu repositório, é necessário digitar estes comandos para fazer um novo commit:

```
MINGW64:/c/Users/Marcelo/github-guia  
  
Marcelo@DESKTOP-1C2AIP8 MINGW64 ~/github-guia (main)  
$ git add .
```

**git add** . Irá adicionar todos os arquivos do diretório

```
Marcelo@DESKTOP-1C2AIP8 MINGW64 ~/github-guia (main)  
$ git commit -m "exemplo"  
[main 361a458] exemplo  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 exemplo.png
```

**git commit -m "nome"** irá fazer o commit e definirá o nome que o seu commit vai ter.

## Push

O comando **git push** irá “empurrar” ou, melhor dizendo, enviar o seu commit para o GitHub.

```
Marcelo@DESKTOP-1C2AIP8 MINGW64 ~/github-guia (main)  
$ git push  
Enumerating objects: 4, done.  
Counting objects: 100% (4/4), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 3.25 KiB | 3.25 MiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/MarceloAbbi/github-guia.git  
03b2d36..361a458 main -> main
```

## Pull Request

Pull Request é o termo utilizado para designar os commits recebidos no seu repositório. Também para quando você quer integrar os seus commits em outro repositório que não seja o seu. Muito prático e útil para quando se está trabalhando em equipe. Para fazer um Pull request via console, é necessário ter a url do repositório e usar o código a seguir:

***git request-pull (nome da sua versão) (link do repositório) (branch escolhida)***

## Merge

Merge Request é o comando utilizado para que o seu Pull Request seja inserido corretamente no Branch escolhido.

Para fazê-lo, é muito simples. Basta digitar o código abaixo:

***Git merge (nome da branch escolhida)***

## Referências

<https://git-scm.com/book/pt-br/v2/Branches-no-Git-O-b%C3%A1sico-de-Ramifica%C3%A7%C3%A3o-Branch-e-Mesclagem-Merge>

<https://git-scm.com/docs/git-request-pull>