

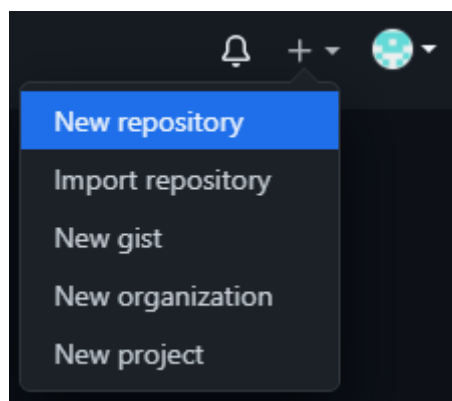
[CRIANDO REPOSITÓRIO REMOTO – GITHUB](#)

[CLONAR REPOSITÓRIO REMOTO PARA O COMPUTADOR](#)

[TRABALHANDO COM VÁRIOS REPOSITÓRIOS LOCAIS](#)

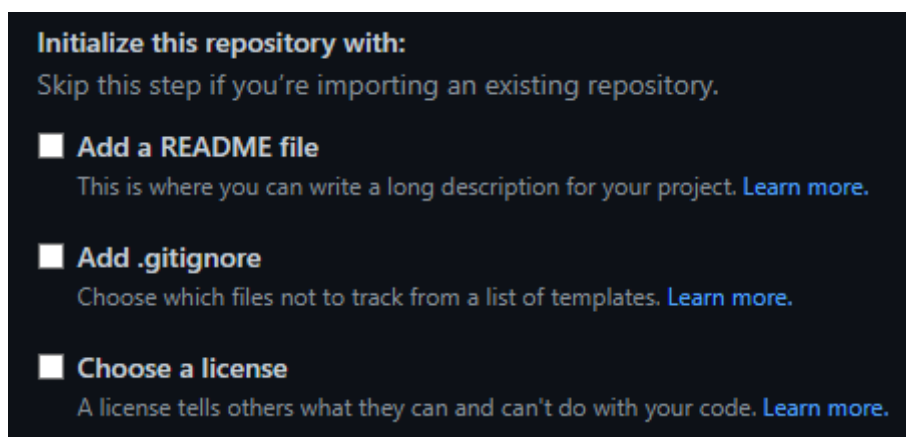
CRIANDO REPOSITÓRIO REMOTO – GITHUB

1. Após criar uma conta no GitHub ([clique para acessar o site](#)), é necessário criar um novo repositório através do menu superior

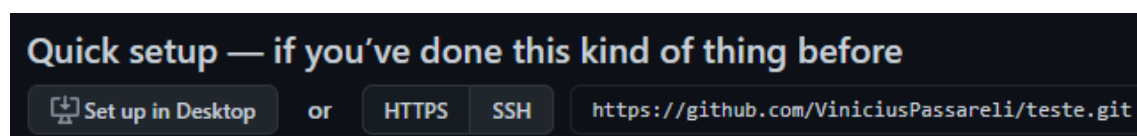


1.1) Preencha os campos de nome, descrição, privacidade;

1.2) O último campo só deve ser preenchido caso o diretório não exista no computador. Nesse caso, marque “Add a README file”



2. Para **criar um novo projeto** no repositório remoto ou **portar um projeto do repositório local** para o repositório remoto, utiliza-se o HTTPS ou o SSH. As instruções também são dadas pelo próprio GitHub



2.1) Para **criar um novo projeto**, utiliza-se a seguinte sequência de linhas de comando no Git BASH

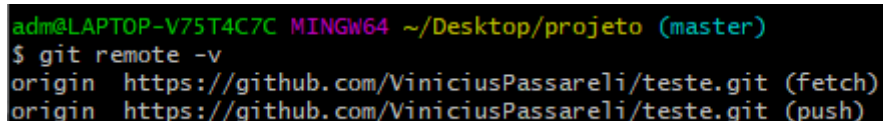
```
echo "# <nome_repositorio_remoto>" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin <HTTPS>
git push -u origin main
```

2.2.1) Para **portar um projeto existente em repositório local** para o repositório remoto, utiliza-se as seguintes linhas de comando. Pode ser solicitada a permissão de acesso do GitHub durante o processo

```
git remote add origin <HTTPS>
git branch -M main
git push -u origin main
```

2.2.2) Para verificar através do Git BASH qual o repositório remoto vinculado ao projeto local, utiliza-se o comando

```
git remote -v
```



```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/projeto (master)
$ git remote -v
origin https://github.com/ViniciusPassareli/teste.git (fetch)
origin https://github.com/ViniciusPassareli/teste.git (push)
```

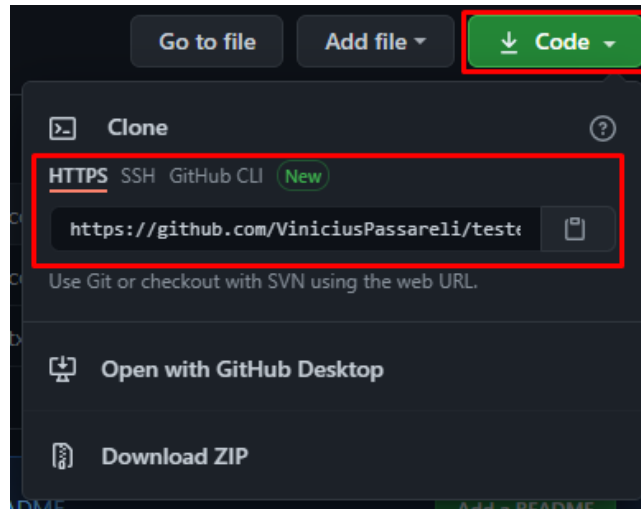
2.2.3) Após o processo, é possível verificar no site do GitHub que o repositório local foi portado com sucesso

3. Para **salvar as novas versões do Projeto no repositório remoto**, faça todo o processo padrão de *commit* e finalize utilizando o comando

```
git push
```

CLONAR REPOSITÓRIO REMOTO PARA O COMPUTADOR

1. Selecione “Code” no repositório do GitHub e copie o endereço HTTPS



2. No diretório local onde se deseja clonar o repositório, inicialize o Git BASH e utilize o comando

`git clone <HTTPS>`

```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto
$ git clone https://github.com/ViniciusPassareli/teste.git
Cloning into 'teste'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 18 (delta 2), reused 18 (delta 2), pack-reused 0
Receiving objects: 100% (18/18), done.
Resolving deltas: 100% (2/2), done.
```

3. A partir desse ponto, é possível alterar o projeto e realizar *commits* por esse diretório local

TRABALHANDO COM VÁRIOS REPOSITÓRIOS LOCAIS: SINCRONIZAÇÃO E SOLUÇÃO DE CONFLITOS

Quando se tem mais de um repositório local que pode realizar alterações no projeto, é importante sempre sincronizar as versões das alterações, utilizando o comando a seguir, antes de realizar mudanças no projeto

```
git pull
```

Esse comando atualiza a versão do projeto no repositório local utilizando a versão presente no GitHub.

Entretanto, existe a possibilidade de duas ou mais pessoas estarem atualizando o repositório remoto constantemente, ao mesmo tempo. Nesse caso, ao tentar realizar um *commit*, mesmo tendo utilizado `git pull` anteriormente, pode ocorrer um conflito de versões (caso tenha ocorrido alguma nova atualização no repositório nesse meio tempo).

Portanto, **para realizar um *commit* a partir de uma versão dessincronizada**, siga o Exemplo 1, a seguir.

EXEMPLO 1

1. Com dois repositórios locais distintos, sincronizados com o mesmo repositório remoto, criou-se um novo arquivo “f.txt” no primeiro repositório local, escrito

“Editando o arquivo f.txt no diretório local 1”

e realizou-se o *commit* para o repositório remoto, utilizando o procedimento normal e finalizando com o comando

```
git push
```

```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/projeto (main)
$ git add *

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/projeto (main)
$ git commit -m "adicionando f.txt (diretorio local 1)"
[main 09a51e2] adicionando f.txt (diretorio local 1)
1 file changed, 1 insertion(+)
create mode 100644 f.txt

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/projeto (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ViniciusPassareli/teste.git
d7c042b..09a51e2 main -> main
```

2. Sem utilizar o `git pull` para sincronizar, no segundo repositório local criou-se um novo arquivo “f.txt” com o texto

“Editando o arquivo f.txt no diretório local 2”

e, ao realizar o `commit` para o diretório remoto, o resultado foi o seguinte

```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main)
$ git add *

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main)
$ git commit -m "adicionando f.txt (diretorio local 2)"
[main 36976f8] adicionando f.txt (diretorio local 2)
1 file changed, 1 insertion(+)
create mode 100644 f.txt

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main)
$ git push
To https://github.com/ViniciusPassareli/teste.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ViniciusPassareli/teste.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

3. Para **solucionar o conflito**, é necessário sincronizar essa alteração com a alteração realizada pelo outro repositório local. Para isso, utiliza-se a sequência de comandos

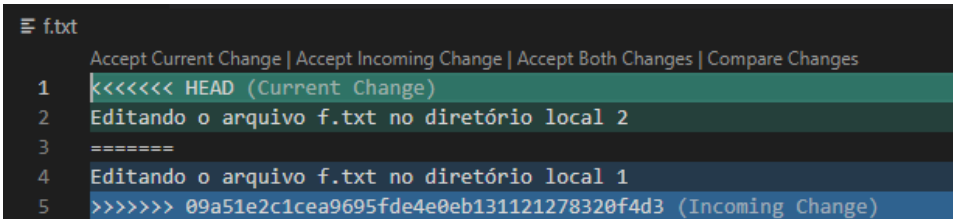
`git pull`

`git code .`

e soluciona-se os conflitos pelo Visual Studio Code (assim como na sincronização de diferentes ramos do Projeto, visto no PDF “3_Ramificações”).

```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 319 bytes | 1024 bytes/s, done.
From https://github.com/ViniciusPassareli/teste
d7c042b..09a51e2 main -> origin/main
CONFLICT (add/add): Merge conflict in f.txt
Auto-merging f.txt
Automatic merge failed; fix conflicts and then commit the result.

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main|MERGING)
$ code .
```



```

f.txt
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1 <<<<<< HEAD (Current Change)
2 Editando o arquivo f.txt no diretório local 2
3 =====
4 Editando o arquivo f.txt no diretório local 1
5 >>>>>> 09a51e2c1cea9695fde4e0eb131121278320f4d3 (Incoming Change)
```

4. Realiza-se, então, o *commit* desde o início (`git add`, `git commit ...`). A versão final presente no repositório remoto será a versão editada no Visual Studio Code

```
adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main|MERGING)
$ git add *

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main|MERGING)
$ git commit -m "alterando f.txt pelo diretorio local 2 sem conflitos"
[main d5ba9e6] alterando f.txt pelo diretorio local 2 sem conflitos

adm@LAPTOP-V75T4C7C MINGW64 ~/Desktop/copia_projeto/teste (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 664 bytes | 664.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/ViniciusPassareli/teste.git
09a51e2..d5ba9e6 main -> main
```