

# Guia de Git

Neste manual irei utilizar o GitBash nas imagens para exemplificar os comandos.

GIT - é um sistema de versionamento: você controla as modificações de um projeto por meio de versões chamadas "commits".

Um projeto controlado pelo Git é chamado de repositório de versionamento. Tipicamente uma cópia "oficial" do repositório fica salvo em um servidor (repositório remoto). Cada pessoa que trabalha no projeto pode fazer uma cópia do repositório para seu computador (repositório local). A pessoa então faz suas alterações no projeto (novos commits) e depois salva as alterações no servidor.

Principais comandos:

- CLONE Copia o repositório remoto para seu computador
- PULL Atualiza seu repositório local em relação ao repositório remoto
- COMMIT Salva uma nova versão (tipicamente no seu repositório local)
- PUSH Envia o repositório local para o repositório remoto

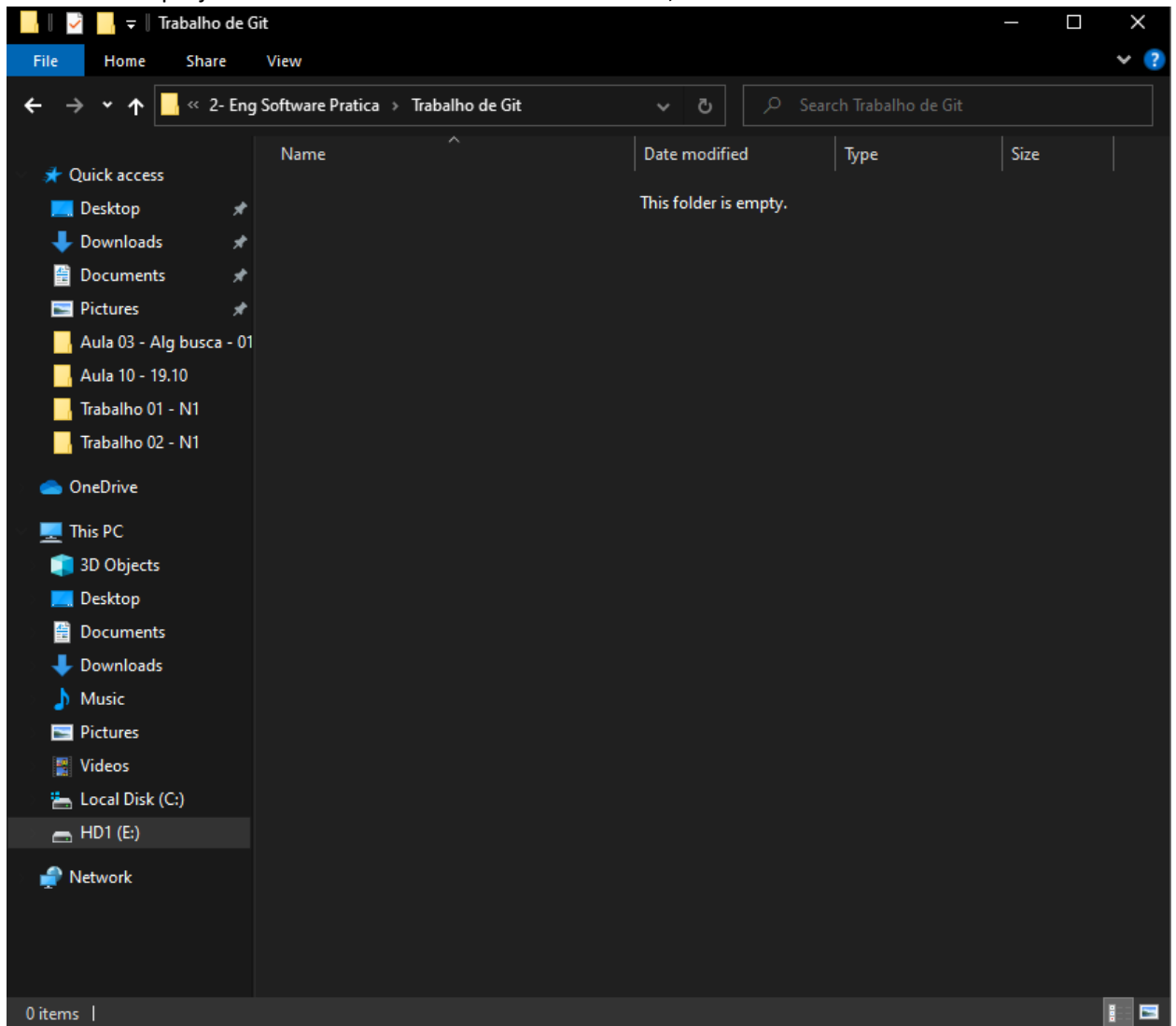
**GITHUB** - é um serviço de hospedagem de repositórios Git remotos.

- Possui uma interface gráfica web: [github.com](https://github.com)
- É uma plataforma social (usuários, página de perfil, seguidores, colaboração, etc.).
- Maior serviço do mundo de hospedagem de projetos de código aberto
- Modelo de cobrança: gratuito para projetos de código aberto, pago para projetos privados
- Alternativas: BitBucket, GitLab, etc

## Exemplos:

Criando repositório:

1. Crie um novo projeto no seu ambiente de desenvolvimento;



2. Crie um novo repositório no Github. Caso o seu ambiente de trabalho não gere o arquivo `.gitignore` automaticamente, criaremos neste passo.


## 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 \*

 LeonardoOliveriadeFarias ▾


 / 

TrabalhoGitEngSoftware ✓


Great repository names are short and memorable. Need inspiration? How about [silver-giggle?](#)

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.

---

**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.](#)

.gitignore template: Python ▾

☐ **Choose a license**

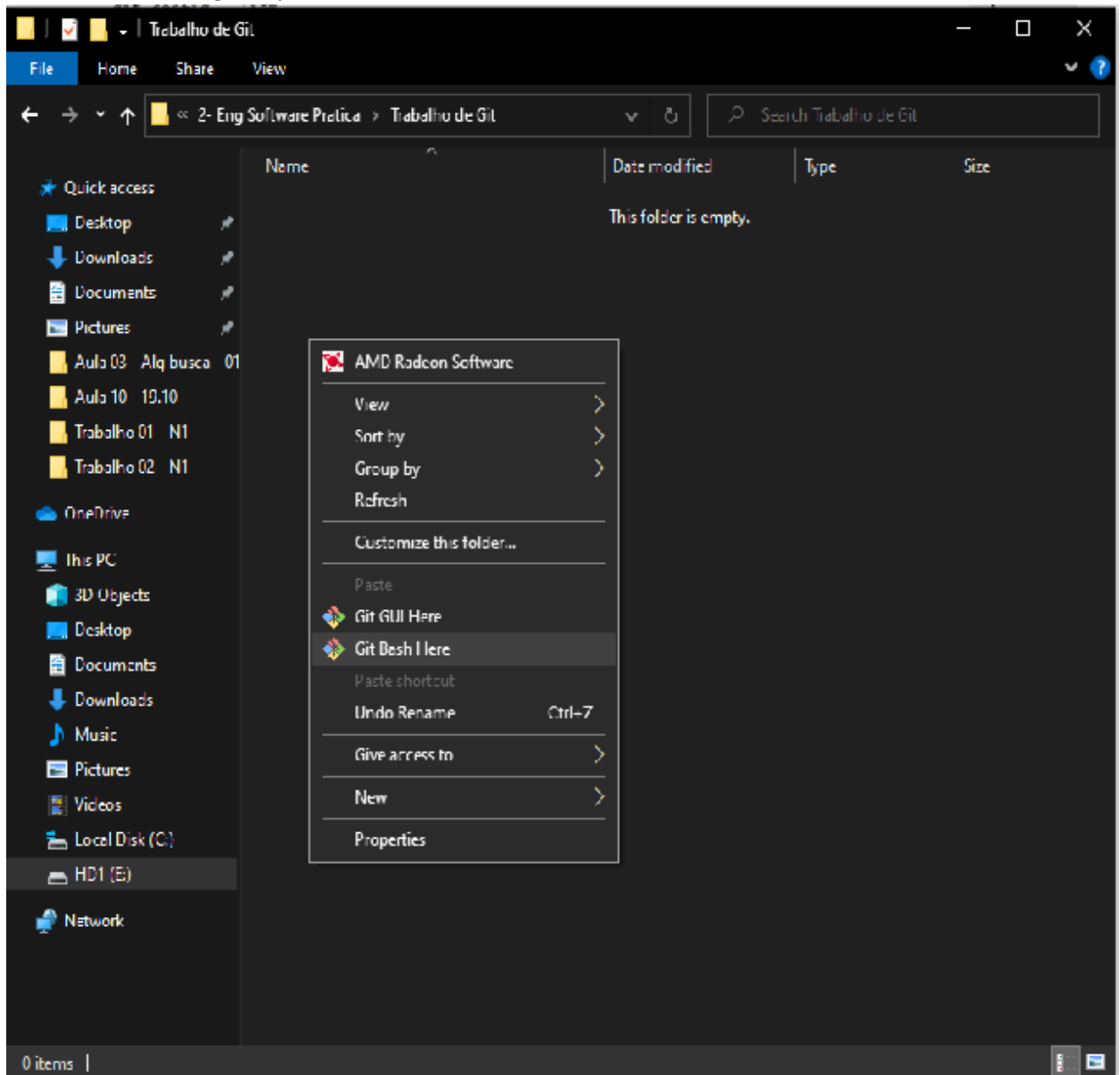
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

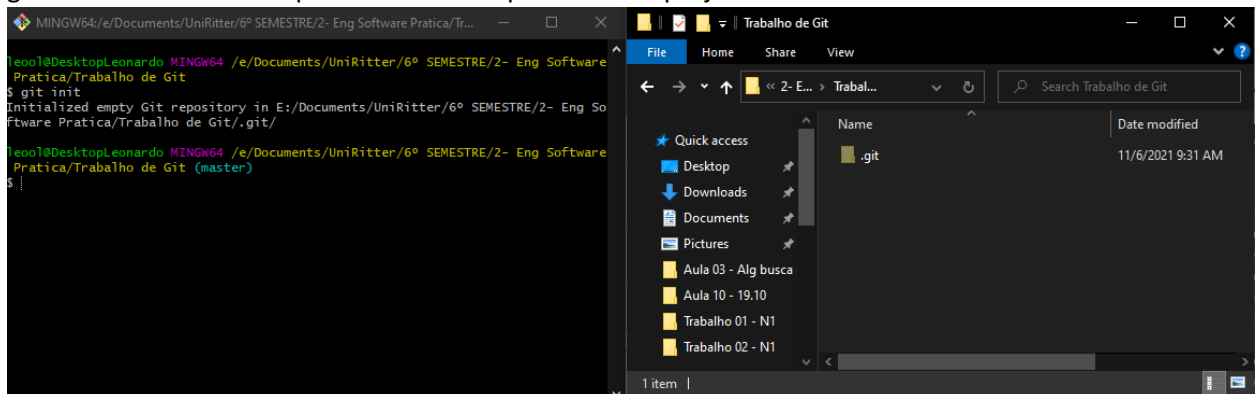
---

Create repository

3. Abra o terminal do git na pasta



4. Agora começam os códigos para criarmos o repositório no nosso ambiente de trabalho  
git init Inicia um novo repositório local na pasta do seu projeto

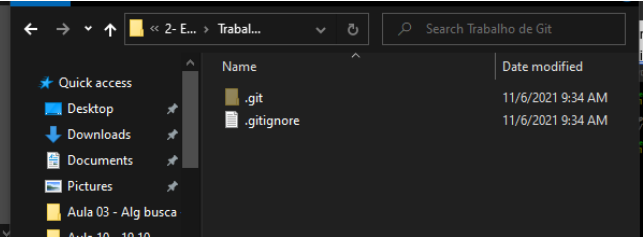


5. git remote add origin <https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware.git>  
Associa seu repositório local ao repositório remoto, com o apelido de "origin"

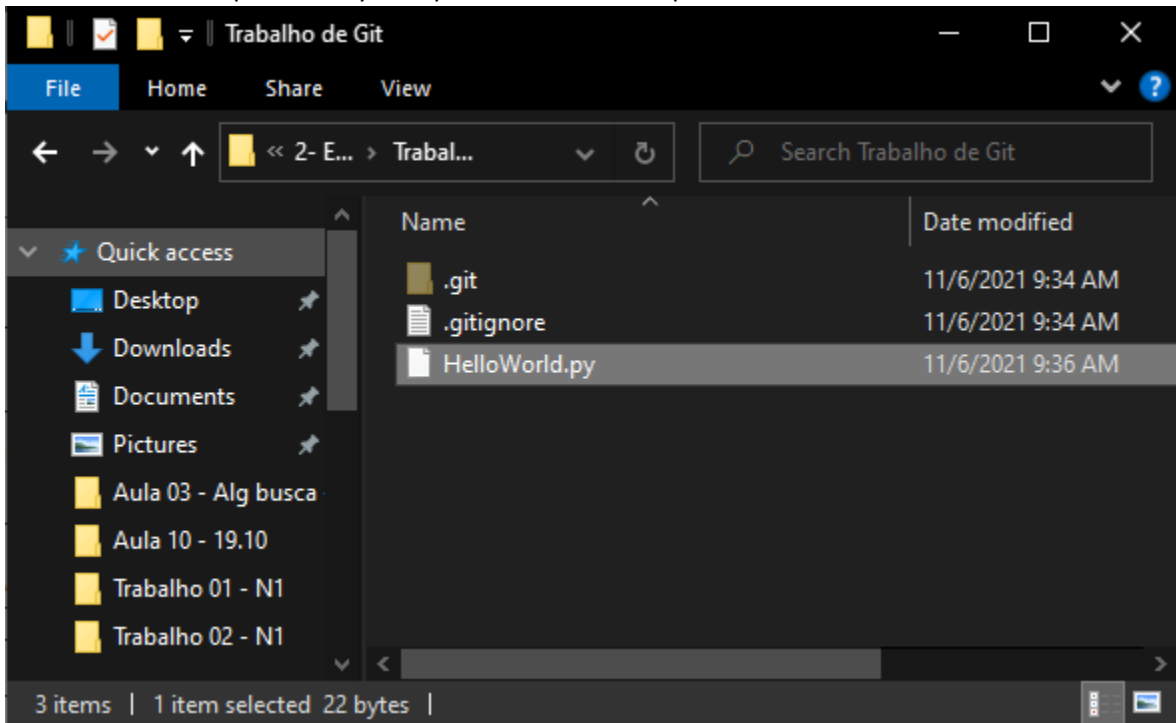
```
leo01@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git remote add origin https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware.git
```

6. git pull origin main: Atualiza seu repositório local em relação ao repositório remoto  
OBS: isso é para adicionarmos o .gitignore ao repositório local.

```
leo01@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git pull origin main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.53 KiB | 9.00 KiB/s, done.
From https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware
* branch      main      -> FETCH_HEAD
* [new branch] main      -> origin/main
```



7. Iremos criar um arquivo em Python para adicionar ao repositório:



8. git status: Verifica arquivos

```
leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        HelloWorld.py

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

leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ |
```

Ele apresenta que tem um arquivo diferente que não está pronto para commit

9. git add . : Adiciona todos arquivos ao stage.

```
leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git add .

leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   HelloWorld.py
```

OBS: agora o arquivo .py está pronto para ser commitado

10. git commit -m "Mensagem" Salva uma nova versão do projeto

```
leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git commit -m "projeto de python para aula de Engenharia de software"
[master 00d26f6] projeto de python para aula de Engenharia de software
1 file changed, 1 insertion(+)
create mode 100644 HelloWorld.py

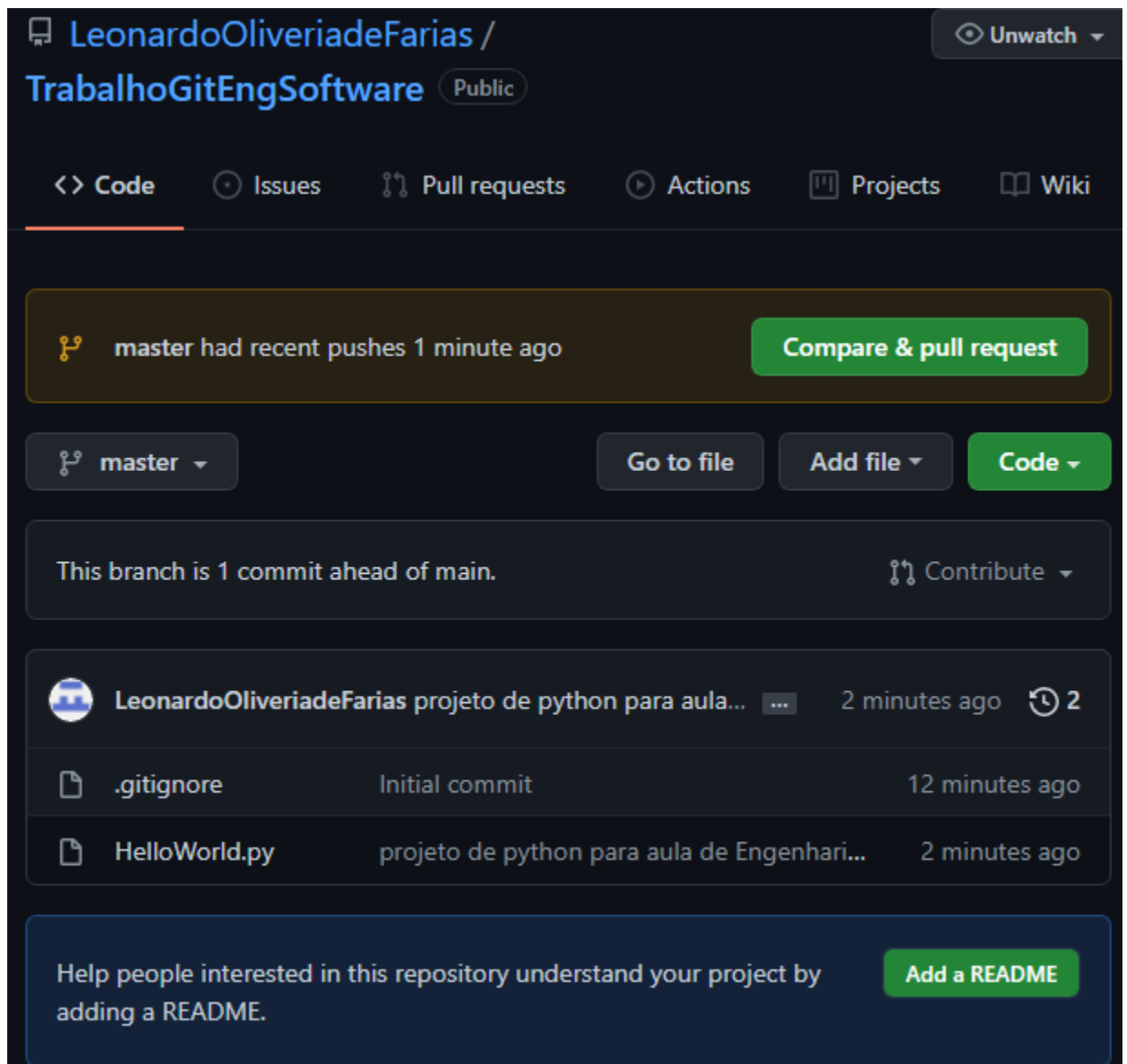
leool@DesktopLeonardo MINGW64 /e/Documents/UniRitter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ |
```

11. git push -u origin master: Envia o repositório local para o repositório remoto

```
leool@DesktopLeonardo MINGW64 /e/Documents/Uniritter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ git push -u origin master
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), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware/
pull/new/master
remote:
To https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

leool@DesktopLeonardo MINGW64 /e/Documents/Uniritter/6º SEMESTRE/2- Eng Software
Pratica/Trabalho de Git (master)
$ |
```

OBS: nas próximas vezes basta fazer: git push



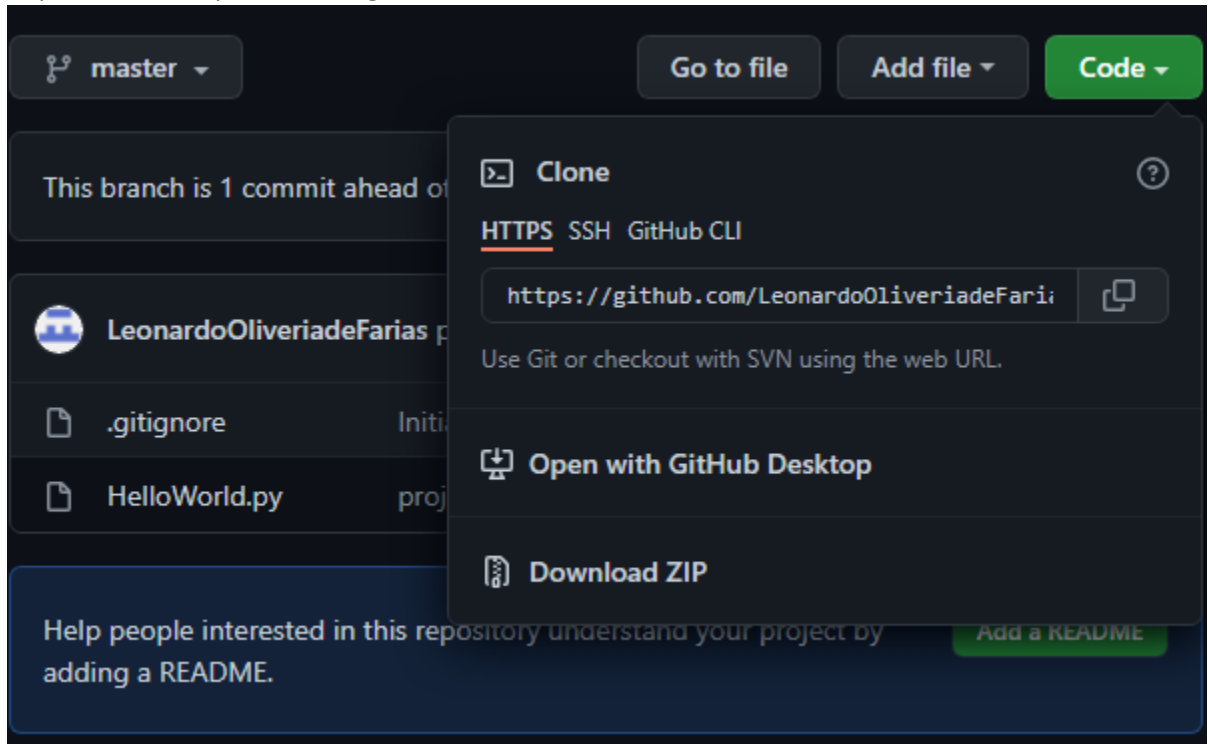
Clonando o repositório.

Usei o mesmo repositório, somente apaguei os arquivos.

1. Abra o terminal do git na pasta;

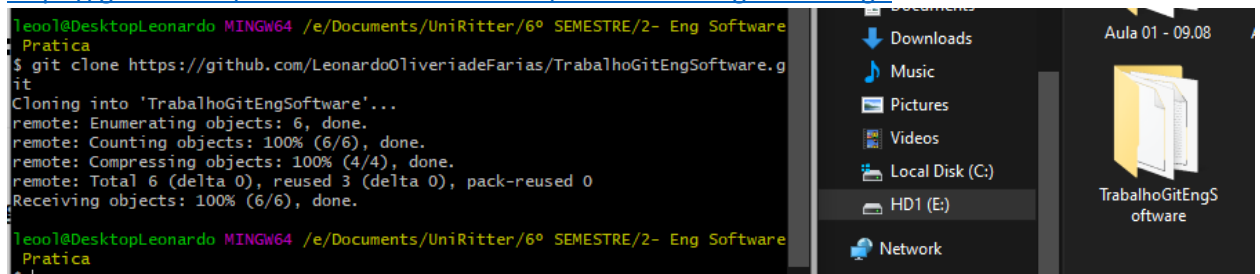


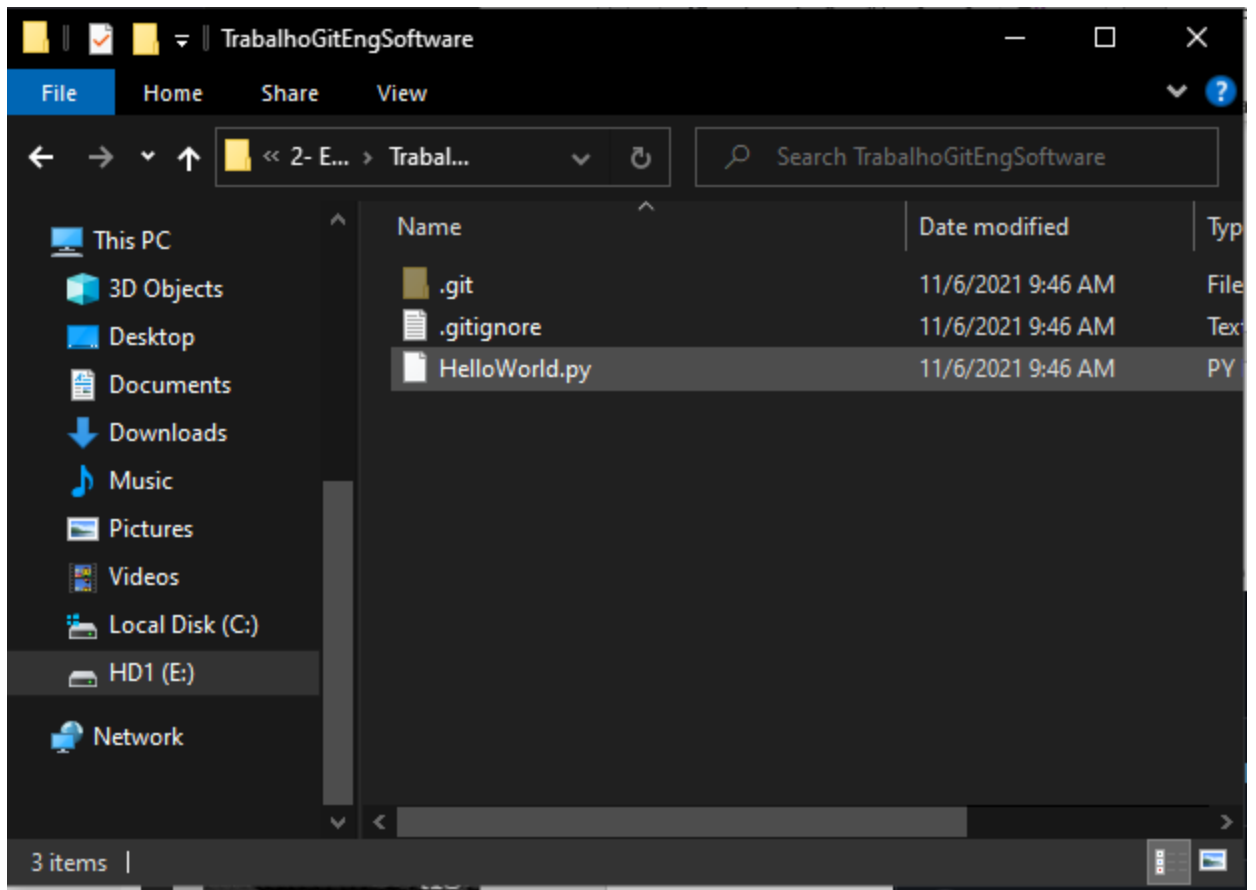
2. Copie o link do repositório no github



3. Utilize no gitbash: git clone LINK -> git clone

<https://github.com/LeonardoOliveriadeFarias/TrabalhoGitEngSoftware.git>





Alguns comandos para “salvar” o repositório de erros:

PROBLEMA	COMANDO
Quero desfazer tudo que eu fiz desde o último commit	<code>git clean -df</code> <code>git checkout -- .</code>
Preciso remover o último commit, porém mantendo os arquivos do jeito que estão.	<code>git reset --soft HEAD~1</code>
Preciso remover o último commit, inclusive as alterações nos arquivos.	<code>git reset --hard HEAD~1</code>
Quero alterar temporariamente os arquivos do projeto de modo a ficarem no estado do commit informado. OBS1: não podem haver modificações não commitadas no projeto. OBS2: para voltar ao último	<code>git checkout &lt;código do commit&gt;</code>

commit faça: git checkout master	
Preciso apagar o último commit no Github	git push -f origin HEAD^:master
Quero mudar o meu repositório remoto "origin"	git remote set-url origin <a href="https://github.com/LeonardoOliveiradeFarias/novoprojeto.git">https://github.com/LeonardoOliveiradeFarias/novoprojeto.git</a>
Entrei no VIM por engano. Como sair?	Tecle ESC, depois digite :q! e tecle ENTER