

## COMANDOS GIT

Após ter instalado o GIT e ter criado a sua conta no GITHUB, vamos precisar configurar o nosso computador para que a gente possa subir ou pegar mudanças do github para nossa máquina.

### 1. Configurando GIT localmente

Nesse caso, vamos inserir nossos dados de acordo com o que colocamos no github, nome de usuário e e-ma. No Git bash digite os comandos abaixo:

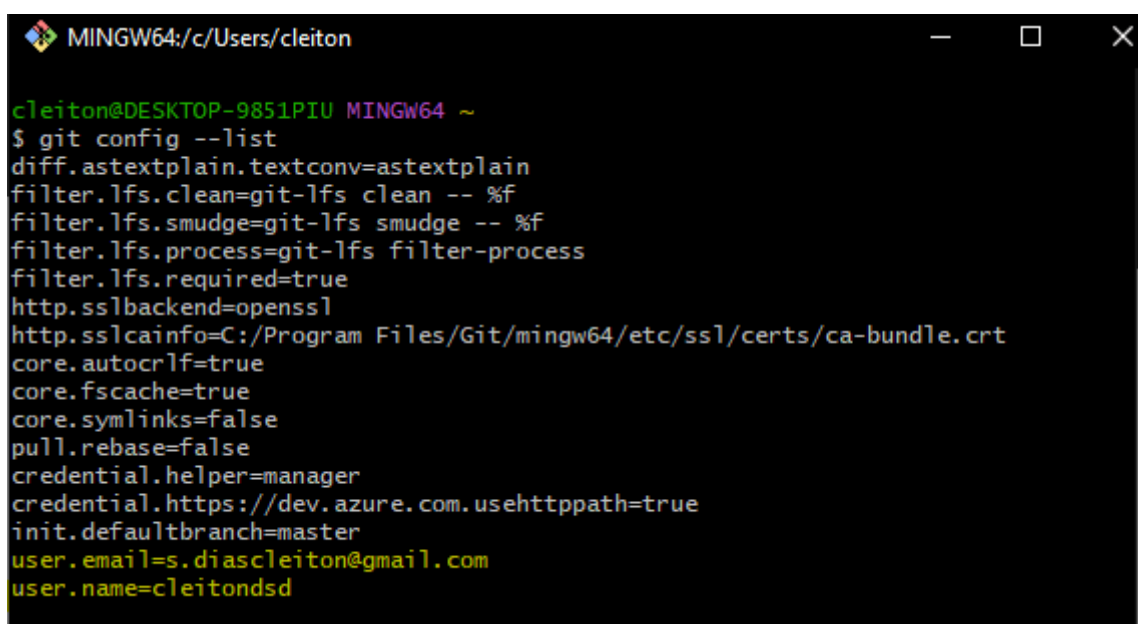
1. `git config --global user.name "Seu Nome"`
2. `git config --global user.email "seuemail@etecuirapuru.com"`



```
MINGW64:/c/Users/cleiton
cleiton@DESKTOP-9851PIU MINGW64 ~
$ git config --global user.name "cleitondsd"
cleiton@DESKTOP-9851PIU MINGW64 ~
$ git config --global user.email "s.diascleiton@gmail.com"
cleiton@DESKTOP-9851PIU MINGW64 ~
$ |
```

Para verificar se os dados estão corretos, vamos digitar o comando:

`git config --list`



```
MINGW64:/c/Users/cleiton
cleiton@DESKTOP-9851PIU MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=s.diascleiton@gmail.com
user.name=cleitondsd
```

Verifique se os seus dados estão corretos, como na imagem acima.

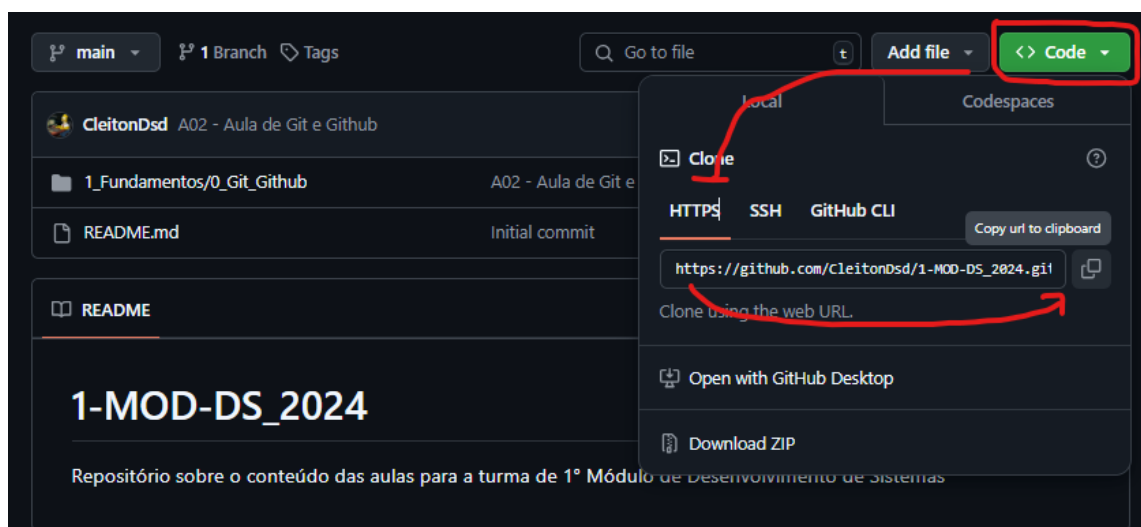
<https://github.com/CleitonDsd>

## 2. Iniciando um projeto do zero ou baixando um já existente

Para iniciar um projeto em uma pasta, ou seja, para sinalizar ao git que ele acompanhe tudo que acontecer dentro daquela pasta, utilize:

**git init**

Podemos iniciar um projeto de outra forma também, pelo github, ao criar um repositório no github podemos clonar ele na nossa máquina:



Copie a URL do seu projeto, no terminal do git bash utilize o comando

**git clone url-do-projeto.git**

exemplo:

**git clone [https://github.com/CleitonDsd/1-MOD-DS\\_2024.git](https://github.com/CleitonDsd/1-MOD-DS_2024.git)**

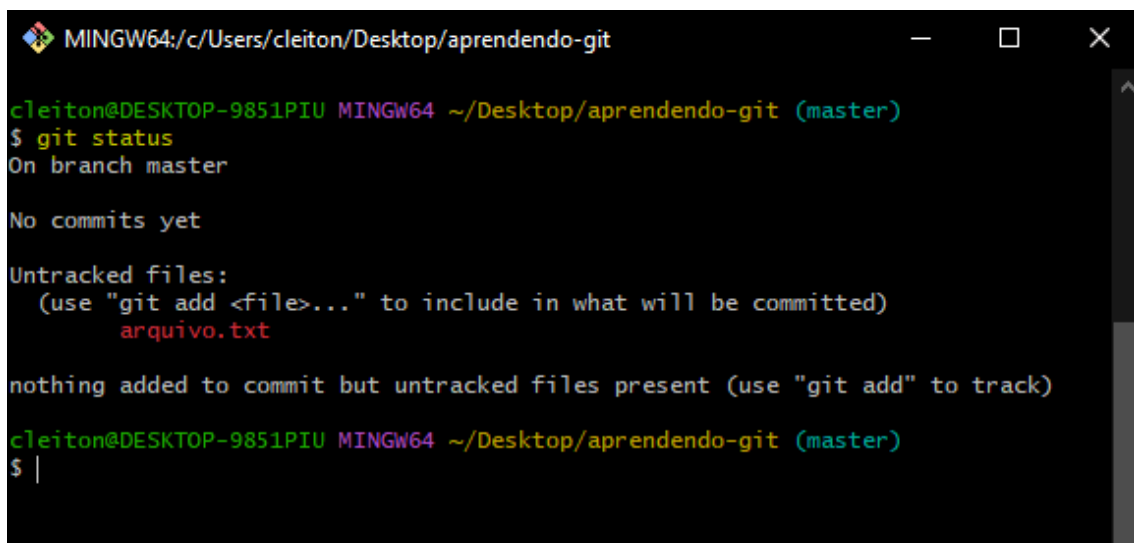
```
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ git clone https://github.com/CleitonDsd/1-MOD-DS_2024.git
Cloning into '1-MOD-DS_2024'...
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 19 (delta 2), reused 14 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (19/19), 8.16 MiB | 3.92 MiB/s, done.
Resolving deltas: 100% (2/2), done.

cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ |
```

### 3. Controlando as mudanças

Esse comando será um dos mais utilizados, quando você digitar ele o git vai ter retornar uma mensagem dizendo se “determinado arquivo foi modificado e que você ainda não salvou (na cor vermelha). Digite o comando:

**git status**



```
MINGW64:/c/Users/cleiton/Desktop/aprendendo-git
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arquivo.txt

nothing added to commit but untracked files present (use "git add" to track)
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ |
```

Git retornou que existe um arquivo chamado “arquivo.txt” que é novo e/ou foi modificado e que ainda não salvamos essa alteração para git acompanhar.

Para dizer ao git: “Quero salvar essa prévia de alterações nesse arquivo”, vamos utilizar o comando abaixo:

**git add nome\_arquivo\_ou\_nome\_da\_pasta**

no caso:

**git add arquivo.txt**



```
MINGW64:/c/Users/cleiton/Desktop/aprendendo-git
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ git add arquivo.txt
warning: in the working copy of 'arquivo.txt', LF will be replaced by CRLF the n
ext time Git touches it
```

Para verificar o status do projeto, digite: **git status**

```
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   arquivo.txt
```

*Indica que tem um arquivo novo que o git passará a acompanhar suas alterações.*

Mas para que essa mudança, **realmente seja salva de forma fixa**, precisamos adicionar uma descrição a ela. Para salvar, digite o comando:

**git commit -m “este é um arquivo de texto”**

```
MINGW64:/c/Users/cleiton/Desktop/aprendendo-git
cleiton@DESKTOP-9851PIU MINGW64 ~/Desktop/aprendendo-git (master)
$ git commit -m "Este é um arquivo de texto"
[master (root-commit) f424c49] Este é um arquivo de texto
1 file changed, 3 insertions(+)
create mode 100644 arquivo.txt
```

Isso significa que salvamos nosso arquivo, com isso, ele criou o famoso “**commit**”.

#### 4. Subindo alterações para o Github

Para subir nossas alterações para o repositório que clonamos do github, vamos utilizar o comando:

**git push**

esse comando diz ao git: “suba as alterações para o github para que todos possam acessar”

Para que possamos atualizar nosso repositório com o que tem no github, vamos utilizar o comando:

**git pull**

**Utilize os materiais disponibilizados em aula.**

**Comandos mais utilizados:**

- git pull (atualize minha máquina com o que tem no github)
- git status (o que tem de alterações novas?)
- git add (adicione as alterações para o git)
- git commit -m “descrição” (salve e adicione uma descrição as alterações no git)
- git push (envie as alterações para o github)