

O que é o Git?

O **Git** é um sistema de controle de versão de arquivos. É um software livre e muito utilizado no desenvolvimento de software onde diversas pessoas estão contribuindo simultaneamente, podendo criar e editar arquivos. Sempre quando alguém disponibiliza sua parte do projeto no Git, ele gerencia as alterações feitas e guarda um histórico. Isso é importante pois se houver algum problema você pode desfazer as alterações e voltar para a versão que estava estável.

O que é o GitHub?

O **GitHub** é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem.



Instalando o Git Bash no Windows

- <https://git-scm.com/downloads>



Downloads



macOS



Windows



Linux/Unix

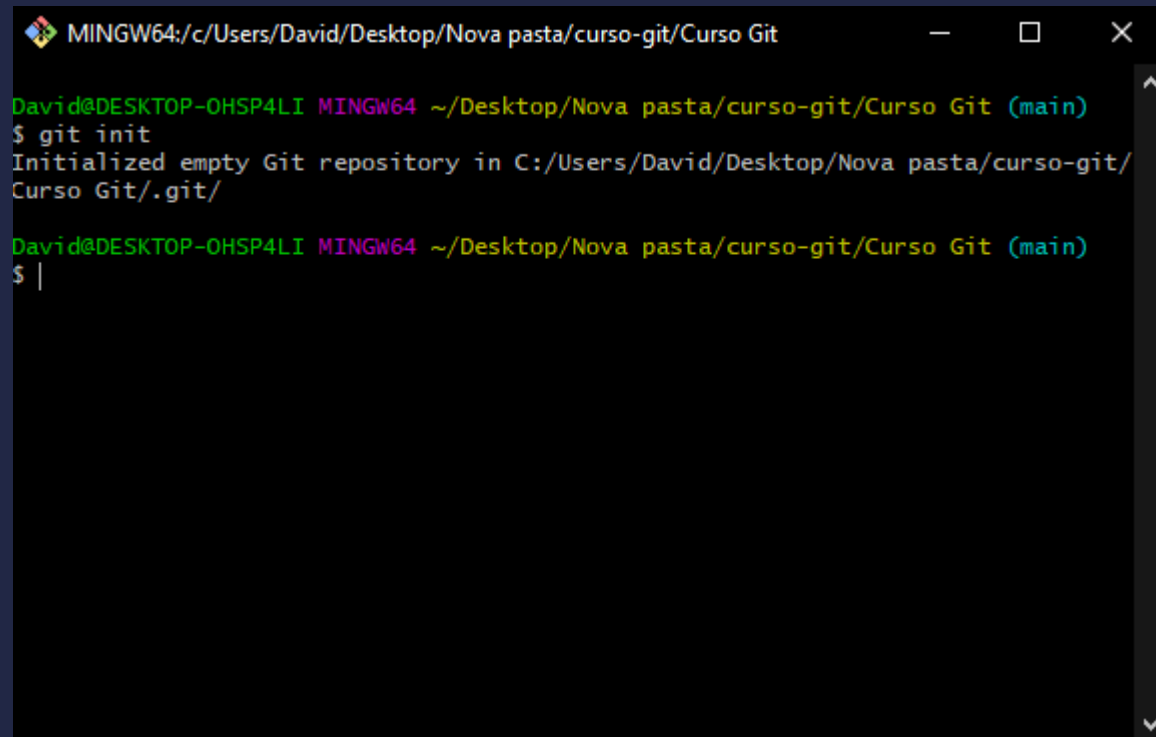
Older releases are available and the Git source repository is on GitHub.



Nosso primeiro comando:

Digite: **git init**

git init é um comando único que você usa durante a configuração inicial de um novo repositório. A execução desse comando cria um novo subdiretório `.git` no diretório de trabalho atual. Essa ação também vai criar uma ramificação principal

A screenshot of a Windows terminal window titled "MINGW64; c:/Users/David/Desktop/Nova pasta/curso-git/Curso Git". The terminal shows the command "git init" being executed, which initializes an empty Git repository in the current directory. The output message is "Initialized empty Git repository in C:/Users/David/Desktop/Nova pasta/curso-git/Curso Git/.git/". The prompt then returns to the user, showing the current directory and branch as "(main)".

```
MINGW64; c:/Users/David/Desktop/Nova pasta/curso-git/Curso Git
David@DESKTOP-OHSP4LI MINGW64 ~/Desktop/Nova pasta/curso-git/Curso Git (main)
$ git init
Initialized empty Git repository in C:/Users/David/Desktop/Nova pasta/curso-git/
Curso Git/.git/

David@DESKTOP-OHSP4LI MINGW64 ~/Desktop/Nova pasta/curso-git/Curso Git (main)
$ |
```

Configurando a identificação do usuário no Git

- `git config --global user.name "Seu Nome"`
- `git config --global user.email "Seu Email"`

Mostra status dos arquivos que precisa ser adicionar ou não

- `git status`

adiciona todos os arquivos da pasta

- `git add .`

Committa seu projeto

- `git commit -m "first projeto"`

Mostra log do projeto

- `git log`
- `git log --oneline`

Descartar todas modificações, voltando ao estado do commit atual:

- `git clean -df`
- `git checkout -- .`

Remover o último commit mantendo as alterações nos arquivos

- `git reset --soft HEAD~1`

Remover o último commit inclusive as alterações nos arquivos

- `git reset --hard HEAD~1`

Navegar entre commits, alterando os arquivos temporariamente

- `git checkout <código do commit>`

Voltar para o último commit

- `git checkout <nome do branch> => master`



- Em suma, o **GitHub** é um serviço baseado em nuvem que hospeda um sistema de controle de versão chamado Git. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.
- Para melhor entender o que é o GitHub e como ele funciona, precisamos dar uma olhada mais a fundo.

Criando um repositório remoto no Github

- Faça login no Github

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.


[Read the guide](#)

[Start a project](#)

Create a new repository

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

Owner *

 Unordestino ▾

Repository name *

CursoGitHub ✓

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

Description (optional)

Aprendendo GitAndGitHubD



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](#)

Associar nosso repositório local ao repositório remoto, dando o apelido de "origin" a ele:

- `git remote add origin <URI do repositório remoto>`

O comando `git branch` permite criar, listar, renomear e excluir ramificações

- `git branch -M main`

Enviar o repositório local para o Github

- `git push -u origin main`

Nota: se você usou a opção `-u`, nas próximas vezes que for subir o branch main basta fazer : `git push`

Copiar o repositório, inclusive o histórico de commits:

- `git clone <URI do repositório remoto>`

Comando para atualizar seu repositório local

- `git pull origin main`



- **Github:**
<https://github.com/Unordestino>
- **Instagram :**
https://www.instagram.com/dav1_souza/
- **LinKedin:**
<https://www.linkedin.com/in/idavisilveira/>
- **Discord:**
<https://discord.gg/zXuKuu9>