



Sumário

1. INTRODUÇÃO AO GIT	3
1.1 Fluxo de trabalho	3
1.2 Criando um repositório	3
1.3 Configurações.....	4
1.3.1 Setar usuário e e-mail.....	4
1.3.2 Listar configurações	4
2. ADICIONANDO AO INDEX	5
2.1 Adicionar todos os arquivos e diretórios.....	5
2.2 Adicionar um diretório em específico	5
3. COMMITANDO AO REPOSITÓRIO LOCAL.....	6
4. ENVIANDO ALTERAÇÕES	6
4.1 Clonando repositório remoto	6
4.2 Enviando repositório ao GitHub	7
4.3 Adicionando repositório remoto	8

1. INTRODUÇÃO AO GIT

1.1 Fluxo de trabalho

O Git ajuda a gerenciar diferentes versões de projetos, permitindo que você trabalhe de forma colaborativa e organize suas alterações de forma eficaz. É como uma máquina do tempo que rastreia o histórico das suas edições e facilita a colaboração em equipe. Seu funcionamento pode ser resumido em um fluxo de trabalho simples:



Diretório de Trabalho: Você trabalha nos arquivos do projeto em uma área chamada " **Workspace** ". É onde você faz suas edições e cria novos arquivos.

Área de Preparação: Quando você deseja salvar suas alterações, move os arquivos que deseja salvar para a "Index". É como preparar uma caixa para envio, onde você seleciona o que deseja incluir.

Commit: Depois de colocar os arquivos na Área de Preparação, você faz um "commit". Isso é como tirar uma foto instantânea do estado atual dos arquivos, ou seja, enviá-los ao "**Local Repository**". Cada commit tem uma mensagem que descreve o que foi feito.

Repositório Remoto: Além do seu projeto local, você pode ter uma cópia no servidor chamada "**Remote Repository**". Isso permite que você compartilhe seu trabalho com outras pessoas e mantenha uma cópia segura online.

1.2 Criando um repositório

Crie uma pasta, abra-a e execute o comando "git init" para criar um novo repositório.

```
MINGW64:/c/Users/usuário/Desktop/pasta
usuário@DESKTOP-RVFIGLO MINGW64 ~/Desktop/pasta
$ git init
Initialized empty Git repository in C:/Users/usuário/Desktop/pasta/.git/
usuário@DESKTOP-RVFIGLO MINGW64 ~/Desktop/pasta (master)
$
```

1.3 Configurações

As configurações do GIT são armazenadas no arquivo .gitconfig localizado dentro do diretório do usuário do Sistema Operacional. (Ex.: Windows: C:\Users\NOME_PC\.gitconfig)

As configurações realizadas através dos comandos abaixo serão incluídas no arquivo citado acima

1.3.1 Setar usuário e e-mail

Realize os comandos abaixo dentro da pasta onde o git foi inicializado.

```
git config --global user.name "SEU NOME COMPLETO"  
git config --global user.email "seu@email.com"
```

Estes comandos são importantes porque configuram a identidade do usuário no Git. Isso é essencial para associar suas contribuições e commits aos devidos autores, tornando o histórico de versionamento preciso e permitindo uma colaboração eficaz em projetos compartilhados.

1.3.2 Listar configurações

O comando git config --list é usado para listar todas as configurações do Git no ambiente atual. Ele exibe as configurações, incluindo nome de usuário, endereço de e-mail, alias, entre outras, que foram definidas no Git. Isso é útil para verificar e confirmar as configurações globais e locais que afetam o comportamento do Git em um projeto específico. É uma maneira rápida de revisar as configurações para garantir que tudo esteja configurado conforme necessário antes de começar a trabalhar em um projeto.

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)  
$ 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/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager-core  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
user.email=  
user.name=  
core.repositoryformatversion=0  
core.filemode=false  
core.bare=false  
core.logallrefupdates=true  
core.symlinks=false  
core.ignorecase=true
```

2. ADICIONANDO AO INDEX

Você pode propor mudanças (adicioná-las ao Index) usando "git add", ou seja, adiciona os arquivos novos ou alterados em seu diretório de trabalho à área de preparação do Git. Este é o primeiro passo no fluxo de trabalho básico do git.

2.1 Adicionar todos os arquivos e diretórios

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arquivo.txt
        arquivo2.txt
        arquivo3.txt

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

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git add .

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   arquivo.txt
        new file:   arquivo2.txt
        new file:   arquivo3.txt

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ |
```

Observe que o comando "git status" verifica os estados dos arquivos que estão em nosso diretório.

2.2 Adicionar um diretório em específico

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git add arquivo.txt

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git status
On branch master

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        arquivo2.txt
        arquivo3.txt
```

3. COMMITANDO AO REPOSITÓRIO LOCAL

Para confirmar as mudanças enviadas ao index (isto é, fazer um commit), utilizamos o comando "git commit". Assim, os arquivos serão enviados para o HEAD (Repositório local), mas ainda não para o repositório remoto.

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$ git commit -m "COMENTÁRIO"
[master (root-commit) afc3b5c] COMENTÁRIO
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 arquivo.txt
 create mode 100644 arquivo2.txt
 create mode 100644 arquivo3.txt

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)
$
```

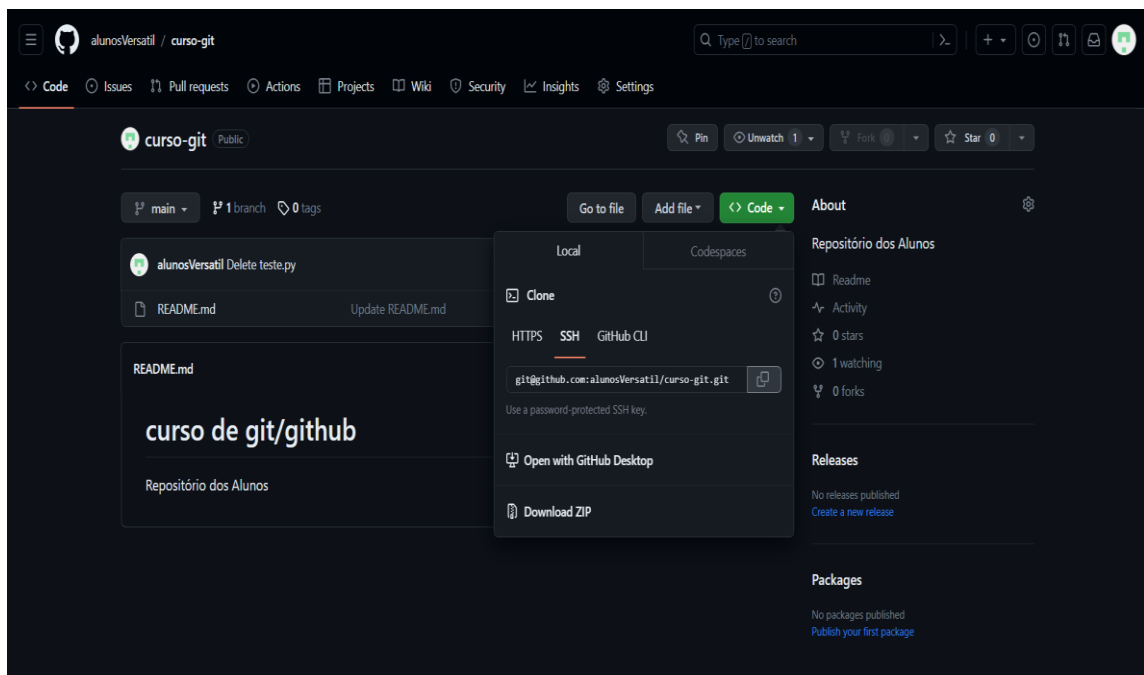
4. ENVIANDO ALTERAÇÕES

4.1 Clonando repositório remoto

Para darmos seguimento ao nosso processo de envio, enviaremos nossa pasta para um repositório remoto já existente, portanto, é necessário fazer um "git clone" do nosso repositório criado no GitHub.

Acesse a url: <https://github.com/alunosVersatil/curso-git>

Em seguida clique na opção "<>Code" de copie o caminho **HTTPS** do repositório.



Feito o último passo a passo. Abra o git bash na área de trabalho e execute o comando "git clone <caminho https copiado>"

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop
$ git clone git@github.com:alunosVersatil/curso-git.git
Cloning into 'curso-git'...
The authenticity of host 'github.com (20.201.28.151)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCoqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

IMPORTANTE: Após realizar o git clone, faça uma alteração no repositório clonado, adicione um arquivo ou crie uma pasta dentro dele. Use os comandos git add para estagiar as mudanças e git commit para prepará-las para envio.

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/curso-git (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/curso-git (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 422 bytes | 422.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:alunosVersatil/curso-git.git
   c8f8451..f05313b  main -> main
```

4.2 Enviando repositório ao GitHub

Para enviar as alterações ao repositório remoto, utilizamos o comando "git push".

O primeiro push de um repositório deve conter o nome do repositório remoto e em seguida a branch. Como no exemplo a seguir:

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/curso-git (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/curso-git (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 422 bytes | 422.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:alunosVersatil/curso-git.git
   c8f8451..f05313b  main -> main

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/curso-git (main)
$
```

4.3 Adicionando repositório remoto

OBS: Comando a ser utilizado caso queira enviar sua pasta para um servidor remoto sem utilizar o “git clone”.

As alterações agora estão no HEAD. Para conectar seu repositório a um servidor remoto, você deve adicioná-lo com o comando “git remote add origin <servidor>”.

Repositório remoto: <https://github.com/alunosVersatil/curso-git.git>

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/pasta (master)  
$ git remote add origin https://github.com/alunosVersatil/curso-git.git
```

Agora deve ser capaz de enviar suas alterações para o servidor remoto com.