



Sumário

1. O QUE É GIT?	3
2. COMANDOS BÁSICOS NO TERMINAL	4
2.1 Navegação no Sistema de Arquivos	4
2.2 Manipulação de Diretórios e Arquivos	5
2.3 Listagem de Arquivos	7
2.4 Comandos de Ajuda	8

1. O QUE É GIT?

Propósito do Git

O Git é um sistema de controle de versão distribuído que tem como principal propósito rastrear as alterações feitas em arquivos ao longo do tempo, permitindo que várias pessoas colaborem em projetos de software de forma eficiente.

Suas principais características incluem:

- **Controle de Versão:** O Git permite que você acompanhe todas as mudanças em seus arquivos, fornecendo um histórico completo das edições.
- **Colaboração:** Facilita a colaboração entre desenvolvedores, permitindo que várias pessoas trabalhem no mesmo projeto simultaneamente.
- **Rastreamento de Mudanças:** Registra quem fez quais alterações e quando, o que é fundamental para a auditoria e resolução de conflitos.
- **Desenvolvimento Ramificado:** Suporta o desenvolvimento em ramificações separadas, permitindo que as alterações sejam desenvolvidas de forma isolada antes de serem mescladas no projeto principal.
- **Distribuído:** Cada cópia de um repositório Git é um repositório completo, tornando-o altamente resistente a falhas e adequado para desenvolvimento descentralizado.
- **Velocidade e Eficiência:** O Git é conhecido por sua velocidade ao lidar com grandes repositórios e suas operações são geralmente rápidas.
- **Flexibilidade:** É altamente personalizável e pode ser integrado com uma variedade de ferramentas de desenvolvimento.

2. COMANDOS BÁSICOS NO TERMINAL

Aprender comandos básicos de terminal para usar o Git é essencial. Essa habilidade economiza tempo, aumenta a eficácia e é fundamental para desenvolvedores que desejam aproveitar ao máximo o Git em seus projetos. Então, neste tópico será abordado alguns comandos básicos fundamentais para trabalhar com versionamento de código.

2.1 Navegação no Sistema de Arquivos

O sistema de arquivos é a estrutura que organiza e armazena seus arquivos e diretórios. Para navegar por ele, usamos comandos como "cd" para entrar em diretórios e "cd .." para voltar. Também é importante mencionar o comando "dir", usado para exibição de detalhes como tamanho, data de modificação e atributos dos arquivos.

```
Microsoft Windows [versão 10.0.22621.2134]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\usuário> cd ..

C:\Users>cd usuário

C:\Users\usuário>|
```

```
C:\Users>dir
O volume na unidade C é Windows
O Número de Série do Volume é F4D0-6838

Pasta de C:\Users

29/09/2022  14:09    <DIR>          .
29/09/2022  14:06    <DIR>          Public
19/08/2023  10:27    <DIR>          usuário
                0 arquivo(s)                0 bytes
                3 pasta(s)   122.090.545.152 bytes disponíveis

C:\Users>|
```

2.2 Manipulação de Diretórios e Arquivos

Para criar diretórios, usamos o comando "mkdir". Por exemplo, "mkdir novo_diretorio" criará um novo diretório chamado "novo_diretorio". (comando utilizado no powershell)

```
PS C:\Users\usuário> mkdir diretorio_novo

Diretório: C:\Users\usuário

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2023   15:39                diretorio_novo

PS C:\Users\usuário> |
```

Para mudar o nome de um arquivo ou diretório, usamos o comando mv. Exemplo: mv arquivo_antigo novo_nome.

```
PS C:\Users\usuário> mv diretorio_novo dir_teste
PS C:\Users\usuário> cd .\dir_teste\
PS C:\Users\usuário\dir_teste> |
```

Para copiar arquivos ou diretórios, usamos cp. Exemplo: cp arquivo.txt diretorio_destino/.

```
PS C:\Users\usuário\dir_teste> ls

Diretório: C:\Users\usuário\dir_teste

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2023   16:12             pasta

PS C:\Users\usuário\dir_teste> mkdir teste.txt

Diretório: C:\Users\usuário\dir_teste

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2023   16:13             teste.txt

PS C:\Users\usuário\dir_teste> cp .\teste.txt\ .\pasta\
PS C:\Users\usuário\dir_teste> cd .\pasta\
PS C:\Users\usuário\dir_teste\pasta> ls

Diretório: C:\Users\usuário\dir_teste\pasta

Mode                LastWriteTime         Length Name
----                -
d-----          04/09/2023   16:13             teste.txt

PS C:\Users\usuário\dir_teste\pasta> |
```

Para remover arquivos ou diretórios, usamos rm ou rmdir. Cuidado: rm apaga permanentemente os arquivos.

```
PS C:\Users\usuário\dir_teste\pasta> ls

Diretório: C:\Users\usuário\dir_teste\pasta

Mode                LastWriteTime         Length Name
----                -
d-----            04/09/2023    16:13             teste.txt

PS C:\Users\usuário\dir_teste\pasta> rm .\teste.txt\
PS C:\Users\usuário\dir_teste\pasta> ls
PS C:\Users\usuário\dir_teste\pasta> |
```

2.3 Listagem de Arquivos

O comando “ls” é usado para listar os arquivos e diretórios em um diretório atual. Pode ser combinado com várias opções para obter informações adicionais, como “ls -l” para exibir detalhes em formato de lista.

(comando utilizado com gitbash)

```
usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/Versátil
$ ls
Angular/  JAVA/  'PHP Básico e Avançado'/  Python/  'Python(Beta)'/  'SQL - MYSQL'/  projeto_sql/

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/Versátil
$ ls -l
total 24
drwxr-xr-x 1 usuário 197611 0 Apr 12 10:46 Angular/
drwxr-xr-x 1 usuário 197611 0 Jan 13 2023 JAVA/
drwxr-xr-x 1 usuário 197611 0 Jan 30 2023 'PHP Básico e Avançado'/
drwxr-xr-x 1 usuário 197611 0 Jan 30 2023 Python/
drwxr-xr-x 1 usuário 197611 0 Sep 2 10:46 'Python(Beta)'/
drwxr-xr-x 1 usuário 197611 0 Jan 26 2023 'SQL - MYSQL'/
drwxr-xr-x 1 usuário 197611 0 Jan 26 2023 projeto_sql/

usuário@DESKTOP-RVF1GLO MINGW64 ~/Desktop/Versátil
$ |
```

2.4 Comandos de Ajuda

Quando precisamos de informações sobre um comando específico, a opção “ - help” após o comando, pode ser anexada a muitos comandos para exibir informações de ajuda rápida.

```
usuário@DESKTOP-RVFLGLO MINGW64 ~/Desktop/Versátil
$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
-m, --mode=MODE    set file mode (as in chmod), not a=rwx - umask
-p, --parents       no error if existing, make parent directories as needed
-v, --verbose       print a message for each created directory
-Z                set SELinux security context of each created directory
                  to the default type
--context[=CTX]    like -Z, or if CTX is specified then set the SELinux
                  or SMACK security context to CTX
--help             display this help and exit
--version          output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/mkdir>
or available locally via: info '(coreutils) mkdir invocation'

usuário@DESKTOP-RVFLGLO MINGW64 ~/Desktop/Versátil
$
```