

## CRIAR

Clona um repositório existente

```
$ git clone ssh://user@domain.com/rep.git
```

Cria um novo repositório local

```
$ git init
```

## MODIFICAÇÕES LOCAIS

Arquivos modificados no diretório ativo

```
$ git status
```

Modificações em arquivos versionados

```
$ git diff
```

Adicione todas as alterações atuais ao próximo commit

```
$ git add .
```

Adicione as mudanças do <file> no próximo commit

```
$ git add -p <file>
```

Commita todas as modificações de arquivos versionados

```
$ git commit -a
```

Commita modificações preparadas

```
$ git commit
```

Modifica o último commit

*Não modifique commits publicados!*

```
$ git commit --amend
```

## LINHA DO TEMPO

Mostra todos os commits, começando pelo mais novo

```
$ git log
```

Mostra as modificações para um arquivo específico

```
$ git log -p <file>
```

Quem mudou o quê e quando em um arquivo

```
$ git blame <file>
```

## BRANCHES & TAGS

Lista todas as branches existentes

```
$ git branch -av
```

Muda a branch atual

```
$ git checkout <branch>
```

Cria uma branch a partir do HEAD atual

```
$ git branch <new-branch>
```

Cria uma nova branch de rastreamento com base em uma branch remoto

```
$ git checkout --track <remote/branch>
```

Deleta uma branch local

```
$ git branch -d <branch>
```

Marca o commit atual com uma tag

```
$ git tag <tag-name>
```

## ATUALIZAR E PUBLICAR

Lista todos os remotes configurados atualmente

```
$ git remote -v
```

Mostra informações sobre um remote

```
$ git remote show <remote>
```

Adiciona um novo repositório remoto, nomeado <remote>

```
$ git remote add <shortname> <url>
```

Baixa todas as modificações do <remote>, mas não integra ao HEAD

```
$ git fetch <remote>
```

Baixa as modificações e automaticamente faz o merge

```
$ git pull <remote> <branch>
```

Publica as modificações locais em um remote

```
$ git push <remote> <branch>
```

Deleta uma branch no remote

```
$ git branch -dr <remote/branch>
```

Publica suas tags

```
$ git push --tags
```

## MERGE & REBASE

Fazer merge da <branch> no HEAD atual

```
$ git merge <branch>
```

Fazer rebase de seu HEAD na <branch>

*Não faça rebase com commits publicados!*

```
$ git rebase <branch>
```

Abortar um rebase

```
$ git rebase --abort
```

Continuar um rebase depois de resolver conflitos

```
$ git rebase --continue
```

Usar a sua ferramenta de merge configurada para resolver conflitos

```
$ git mergetool
```

Use seu editor para resolver conflitos manualmente e marcar o arquivo como resolvido

```
$ git add <resolved-file>
```

```
$ git rm <resolved-file>
```

## DESCAZER

Descarta todas as mudanças locais no diretório atual

```
$ git reset --hard HEAD
```

Descarta mudanças locais em um arquivo específico

```
$ git checkout HEAD <file>
```

Reverte um commit (criando um novo com as modificações ao contrário)

```
$ git revert <commit>
```

Reseta o ponteiro do HEAD para um commit anterior

...e descarta as modificações desde então

```
$ git reset --hard <commit>
```

...e preserva todas as modificações como modificações não preparadas

```
$ git reset <commit>
```

...e preserva modificações locais não commitadas

```
$ git reset --keep <commit>
```

4.1 Git init

4.2 Git clone

4.3 Git branch

4.4 Git checkout

4.5 Git status

4.6 Git diff

4.7 Git add

4.8 Git commit

4.9 Git push

4.10 Git pull

4.11 Git revert

4.12 Git merge

4.13 Git stash



# LINUX

## 10 comandos essenciais

pwd

Exiba o nome do diretório atual

cd

Navegue pelas pastas

ls

Visualize arquivos no diretório atual

mkdir

Crie um novo diretório

rmdir

Remova um diretório

sudo

Tarefas com permissões root ou adm

man

Manual de comandos com descrições

cp

Copie arquivos e pastas

mv

Mova arquivos e pastas

clear

Limpe o terminal

