

# Comandos do Git

Guia de referência rápida

## Configuração Inicial

```
git config --global user.name "Seu Nome" - Define o nome do usuário globalmente
```

```
git config --global user.email "email@exemplo.com" - Define o e-mail globalmente
```

```
git config --list - Lista todas as configurações
```

```
git config --global core.editor vim - Define o editor padrão
```

## Criar / Clonar Repositório

```
git init - Inicia um novo repositório no diretório atual
```

```
git init nome-do-projeto - Cria um novo repositório em uma pasta
```

```
git clone https://github.com/user/repo.git - Clona um repositório remoto
```

```
git clone https://github.com/user/repo.git pasta - Clona em uma pasta específica
```

## Status e Histórico

```
git status - Exibe o estado dos arquivos no repositório
```

```
git log - Mostra o histórico de commits
```

```
git log --oneline - Histórico resumido (uma linha por commit)
```

```
git log --oneline --graph --all - Histórico com grafo de branches
```

```
git diff - Mostra mudanças ainda não adicionadas ao stage
```

```
git diff --staged - Mostra mudanças no stage (prontas para commit)
```

## Adicionar e Commitar

```
git add arquivo.txt - Adiciona um arquivo ao stage
```

```
git add . - Adiciona todos os arquivos modificados ao stage
```

```
git add -p - Adiciona partes específicas de arquivos (interativo)
```

```
git commit -m "mensagem" - Cria um commit com mensagem
```

```
git commit -am "mensagem" - Adiciona e commita arquivos rastreados
```

```
git commit --amend - Modifica o último commit
```

## Branches

```
git branch - Lista todas as branches locais
```

```
git branch -a - Lista branches locais e remotas
```

```
git branch nome-da-branch - Cria uma nova branch
```

```
git checkout nome-da-branch - Troca para a branch especificada
```

```
git checkout -b nome-da-branch - Cria e troca para nova branch
```

```
git switch nome-da-branch - Troca de branch (comando moderno)
```

```
git switch -c nome-da-branch - Cria e troca para nova branch (moderno)
```

```
git branch -d nome-da-branch - Deleta uma branch (seguro)
```

```
git branch -D nome-da-branch - Força a deleção de uma branch
```

## Merge e Rebase

```
git merge nome-da-branch - Mescla uma branch na branch atual
```

```
git merge --no-ff nome-da-branch - Merge com commit de merge forçado
```

```
git merge --abort - Cancela um merge em andamento
```

```
git rebase nome-da-branch - Reaplica commits sobre outra branch
```

```
git rebase -i HEAD~3 - Rebase interativo dos últimos 3 commits
```

```
git rebase --abort - Cancela um rebase em andamento
```

## Repositório Remoto

```
git remote -v - Lista os remotos configurados
```

```
git remote add origin https://github.com/user/repo.git - Adiciona um remoto
```

```
git remote remove origin - Remove um remoto
```

```
git fetch origin - Busca atualizações do remoto sem aplicar
```

```
git pull - Busca e aplica atualizações do remoto
```

```
git pull --rebase - Puxa com rebase em vez de merge
```

```
git push origin main - Envia commits para o remoto
```

```
git push -u origin nome-da-branch - Envia e vincula branch ao remoto
```

```
git push --force-with-lease - Força push com segurança
```

```
git push origin --delete nome-da-branch - Deleta branch no remoto
```

## Desfazer Alterações

```
git restore arquivo.txt - Descarta mudanças no arquivo (não staged)
```

```
git restore --staged arquivo.txt - Remove arquivo do stage
```

```
git reset HEAD~1 - Desfaz o último commit mantendo as mudanças
```

```
git reset --hard HEAD~1 - Desfaz o último commit e descarta mudanças
```

```
git reset --hard origin/main - Reseta para o estado do remoto
```

```
git revert abc1234 - Cria um commit que desfaz outro commit
```

```
git clean -fd - Remove arquivos e diretórios não rastreados
```

## Stash

```
git stash - Salva mudanças temporariamente
```

```
git stash push -m "descrição" - Salva com uma descrição
```

```
git stash list - Lista todos os stashes
```

```
git stash pop - Aplica e remove o stash mais recente
```

```
git stash apply stash@{0} - Aplica um stash específico sem remover
```

```
git stash drop stash@{0} - Remove um stash específico
```

```
git stash clear - Remove todos os stashes
```

## Tags

```
git tag - Lista todas as tags
```

```
git tag v1.0.0 - Cria uma tag leve
```

```
git tag -a v1.0.0 -m "versão 1.0.0" - Cria uma tag anotada
```

```
git push origin v1.0.0 - Envia uma tag para o remoto
```

```
git push origin --tags - Envia todas as tags para o remoto
```

```
git tag -d v1.0.0 - Deleta uma tag local
```

## Outros Comandos Úteis

```
git cherry-pick abc1234 - Aplica um commit específico na branch atual
```

```
git bisect start - Inicia busca binária por um bug
```

```
git blame arquivo.txt - Mostra quem alterou cada linha do arquivo
```

```
git shortlog -sn - Resumo de commits por autor
```

```
git show abc1234 - Exibe detalhes de um commit específico
```

```
git reflog - Histórico de todas as ações (incluindo resets)
```

```
git archive --format=zip HEAD -o projeto.zip - Exporta o repositório como zip
```

**Dica:** Use `git help <comando>` para ver a documentação completa de qualquer comando. Ex: `git help log`