

Resumo Completo de Git e GitHub

GIT E GITHUB - RESUMO COMPLETO

O que é Git?

Git é um sistema de controle de versão distribuído. Ele permite que você acompanhe mudanças no código, volte a versões anteriores, trabalhe em diferentes funcionalidades (branches) e colabore com outras pessoas.

O que é GitHub?

GitHub é uma plataforma baseada em nuvem que hospeda repositórios Git. Ele oferece funcionalidades para colaboração, revisão de código, controle de issues, pull requests, CI/CD, entre outros.

PRINCIPAIS COMANDOS DO GIT

Configuração inicial

```
git config --global user.name "Seu Nome"
git config --global user.email "seu@email.com"
```

Criando e clonando repositórios

```
git init                # Inicializa um repositório Git
git clone <url-do-repo> # Clona um repositório existente
```

Adicionando e commitando arquivos

```
git status              # Mostra o status do repositório
git add <arquivo>       # Adiciona um arquivo específico
git add .               # Adiciona todas as mudanças
git commit -m "mensagem" # Salva as mudanças com uma mensagem
git commit --amend      # Edita o último commit (mensagem ou conteúdo)
```

Enviando e recebendo dados do GitHub

```
git remote add origin <url>      # Conecta com o repositório remoto
git push -u origin main           # Envia os commits para o GitHub (1ª vez)
git push                           # Envia alterações subsequentes
git pull                           # Baixa mudanças do repositório remoto
```

Branches

```
git branch                        # Lista as branches
git branch <nome>                 # Cria nova branch
git checkout <nome>               # Troca para a branch
git checkout -b <nome>            # Cria e troca para a branch
git merge <branch>                # Mescla outra branch na atual
git branch -d <nome>              # Deleta branch local
```

Stash (Guardar mudanças temporariamente)

```
git stash                         # Salva mudanças sem commitar
git stash list                    # Lista stashes salvos
git stash apply                   # Reaplica o último stash
git stash apply stash@{n}         # Reaplica stash específico
git stash drop stash@{n}          # Deleta stash específico
git stash clear                   # Limpa todos os stashes
```

Visualização e histórico

```
git log                           # Mostra o histórico de commits
git log --oneline                  # Histórico resumido
git diff                           # Mostra as mudanças feitas
git diff <branch1> <branch2>      # Compara duas branches
```

Corrigindo e limpando

```
git reset --soft HEAD~1           # Volta commit mas mantém mudanças
git reset --hard HEAD~1           # Apaga commit e mudanças
git clean -f                       # Remove arquivos não rastreados
git rm <arquivo>                  # Remove arquivo e registra no Git
```

TRABALHANDO COM GITHUB

Criando repositório remoto no GitHub

1. Crie o repositório via interface web.

2. No terminal:

```
git remote add origin https://github.com/usuario/repositorio.git
```

```
git push -u origin main
```

Pull Requests (PR)

- Usados para propor mudanças e colaborar.

- A revisão de código acontece nos PRs.

- Merged via web ou com:

```
git checkout main
```

```
git pull
```

```
git merge nome-da-branch
```

```
git push
```

BOAS PRÁTICAS

- Use commits pequenos e frequentes.

- Sempre escreva mensagens de commit claras.

- Use branches para funcionalidades, correções e testes.

- Faça pull antes de push para evitar conflitos.

- Use .gitignore para evitar que arquivos desnecessários sejam versionados.

ERROS COMUNS E COMO CORRIGIR

Commit errado?

```
git commit --amend
```

Esqueci de adicionar arquivo no último commit?

```
git add arquivo
```

```
git commit --amend --no-edit
```

Conflito de merge?

- Resolva os conflitos nos arquivos.

- Depois:

```
git add .
```

```
git commit
```

Preciso descartar mudanças locais?

```
git checkout -- arquivo
```

```
git reset --hard
```

CHAVES SSH PARA GITHUB

```
ssh-keygen -t ed25519 -C "seu@email.com"
```

```
cat ~/.ssh/id_ed25519.pub
```

(Cole a chave pública nas configurações do GitHub.)