

Fluxo Organizado de Comandos Git

1 Inicialização e Configuração Básica [🔗](#)

```
1 cd <diretório>          # Navega até o diretório do projeto
2 git init                # Inicializa o repositório Git
3 git config --global user.name "Seu Nome"
4 git config --global user.email "seu@email.com"
```

2 Status e Controle de Arquivos [🔗](#)

```
1 git status              # Verifica o status dos arquivos
2 git add <arquivo>       # Adiciona arquivo ao staging
3 git add .               # Adiciona todos os arquivos da pasta atual
4 git rm --cached <arquivo> # Remove do staging (mantém no disco)
5 git restore <arquivo>    # Restaura alterações locais
6 git restore --staged <arquivo> # Remove do staging
```

3 Commits [🔗](#)

```
1 git commit -m "mensagem"      # Commit normal
2 git commit -a -m "mensagem"   # Pula o staging para arquivos modificados
3 git commit --amend -m "nova msg" # Edita o último commit
```

4 Verificando Alterações e Histórico [🔗](#)

```
1 git diff                 # Mostra diferenças não staged
2 git log                  # Log completo dos commits
3 git log --oneline        # Log simplificado
4 git log -p               # Log com diferenças dos arquivos
```

5 Branching e Trabalho em Equipe [🔗](#)

```
1 git branch              # Lista branches
2 git branch <nova-branch> # Cria branch
3 git switch <branch>     # Alterna branches
4 git checkout <branch>   # Alternativa ao switch
5 git merge <branch>      # Mescla a branch atual com outra
6 git branch -d <branch>  # Deleta branch mesclada
7 git branch -D <branch>  # Força a exclusão
```

6 Renomeando, Removendo e Movendo Arquivos [🔗](#)

```
1 git rm <arquivo>        # Remove do Git e do disco
2 git mv antigo.ext novo.ext # Renomeia/move arquivo
```

7 Reset (Recuo no Histórico) [🔗](#)

```
1 git reset --soft <commit>      # Volta e mantém no staging
2 git reset --mixed <commit>     # Volta e deixa como unstaged
3 git reset --hard <commit>      # Volta e descarta tudo
```

⚠ Cuidado com `--hard` : pode causar perda de dados permanente.

8 Alias (Atalhos de Comandos) [🔗](#)

```
1 git config --global alias.st status    # Cria atalho: git st
2 git config alias.co checkout          # Atalho local
```

9 Interagindo com o GitHub (Remoto) [🔗](#)

```
1 # Conecta sua pasta local ao repositório remoto do GitHub
2 git remote add origin https://github.com/seu-usuario/nome-do-repositorio.git
3
4 # (Opcional) Renomeia a branch principal para "main" se estiver como "master"
5 git branch -M main
6
7 # Clona um repositório remoto do GitHub para seu computador
8 git clone https://github.com/usuario/repositorio.git
9
10 # Envia a branch local para o GitHub
11 git push origin nome-da-branch
12
13 # Atualiza sua branch local com as mudanças da branch main do GitHub
14 git pull origin main
15
16 # Envia as alterações para o GitHub
17 git push
```