

Comandos Essenciais do Git

Bem-vindos ao seu guia rápido dos comandos Git mais importantes. Esta apresentação foi criada para estudantes iniciantes que precisam de uma referência clara e direta para o dia a dia com Git. Vamos descomplicar o controle de versão juntos!



SEÇÃO 1

Configuração Inicial

```
git config --global  
user.name "Seu Nome"
```

Define o nome do autor para todos os seus commits.

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

Define o e-mail do autor para todos os seus commits.

```
git config --list
```

Exibe todas as configurações Git.



Estes comandos são seguros e fundamentais para começar. Configure-os uma vez e esqueça!

SEÇÃO 2

Repositório

git init

Cria um novo repositório Git vazio no diretório atual. Essencial para iniciar um projeto.

git clone [URL]

Clona um repositório Git existente de uma URL remota para o seu computador local.

Estes são os primeiros passos para começar a versionar seu código ou trabalhar em um projeto existente.



Dia a Dia

1

```
git status
```

Mostra o estado dos arquivos no diretório de trabalho e área de staging.

2

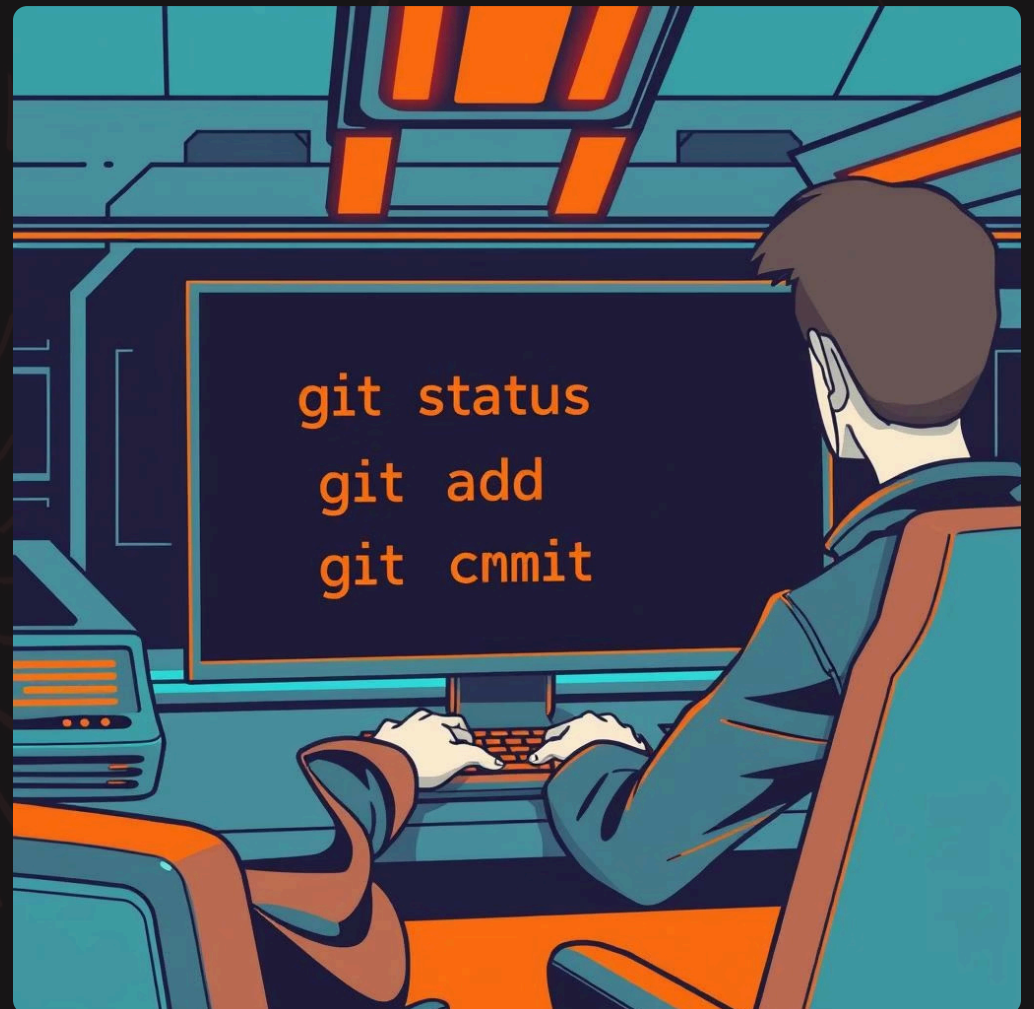
```
git add [arquivo]
```

Adiciona mudanças do diretório de trabalho para a área de staging.

3

```
git commit -m "Mensagem"
```

Grava as mudanças da área de staging no histórico do repositório.



 Estes comandos formam o ciclo básico de trabalho no Git. Pratique-os diariamente!

SEÇÃO 4

Visualização

1

```
git log
```

Exibe o histórico de commits do repositório.

2

```
git diff
```

Mostra as diferenças entre o diretório de trabalho e a área de staging.

3

```
git show [commit ID]
```

Exibe detalhes de um commit específico, incluindo as mudanças feitas.

Use esses comandos para entender o que aconteceu e o que está por vir no seu projeto.

Branches

`git branch`

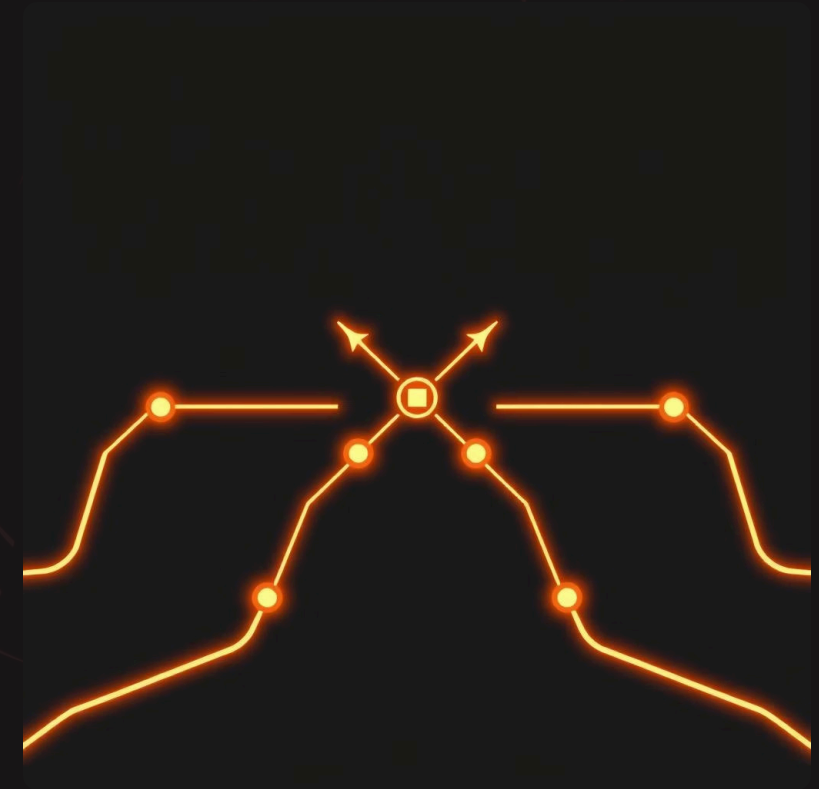
Lista todas as branches locais.

`git checkout [branch]`

Muda para a branch especificada.

`git merge [branch]`

Combina o histórico de uma branch com a branch atual. Pode gerar conflitos.



⊗ **Cuidado:** Operações de merge podem gerar conflitos. Saiba como resolvê-los!

Remoto



`git push`

Envia commits locais para o repositório remoto.



`git pull`


Baixa commits do repositório remoto e os mescla com o local.



`git remote -v`

Lista os repositórios remotos configurados.



 **Atenção:** Sempre puxe antes de empurrar para evitar conflitos.

Emergência



git reset

Desfaz commits ou desfaz a adição de arquivos na área de staging.



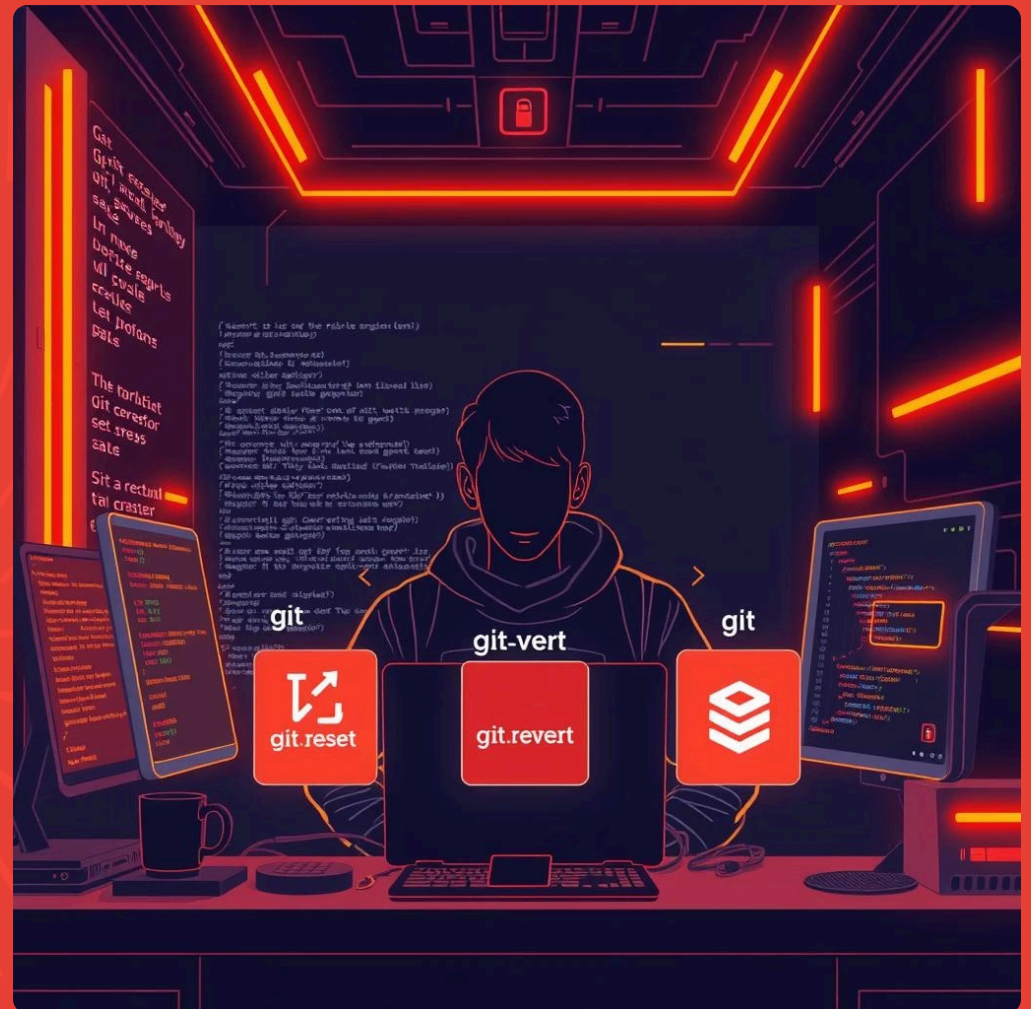
git revert

Cria um novo commit que desfaz as mudanças de um commit anterior.

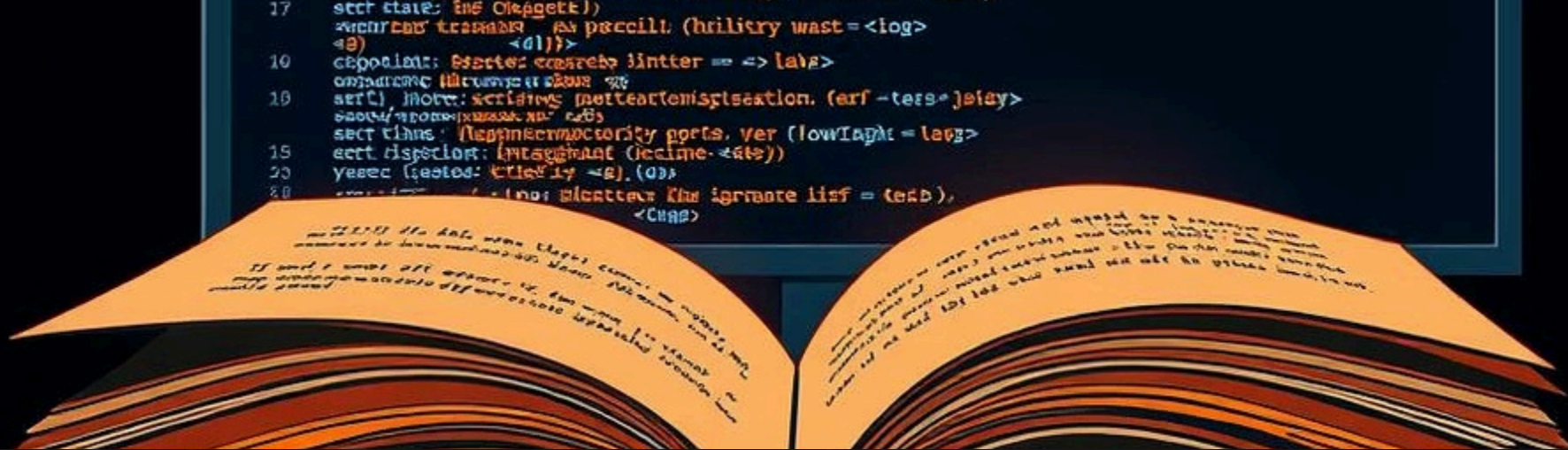


git stash

Salva temporariamente mudanças não comitadas para aplicá-las depois.



⊗ **Perigo:** Use `git reset` com cautela, especialmente com `--hard`, pois pode perder trabalho não salvo.



Recursos Adicionais

- Documentação Oficial do Git: [A fonte mais completa e atualizada para todos os comandos Git.](#)
- Pro Git Book: [Um livro online gratuito que cobre o Git em profundidade, desde o básico ao avançado.](#)
- **Git Cheatsheets:** Procure por "Git cheatsheet" online para referências rápidas e visuais dos comandos mais usados.
- **Exercícios Práticos:** Plataformas como GitHub Learning Lab e Codecademy oferecem cursos interativos para praticar Git.

Próximos Passos

Você agora tem uma base sólida nos comandos essenciais do Git. O próximo passo é colocar a mão na massa!



Pratique Diariamente

Crie seus próprios repositórios e experimente os comandos aprendidos.



Colabore em Projetos

Participe de projetos open source ou crie projetos com colegas para entender o fluxo de trabalho em equipe.



Não Tenha Medo de Errar

O Git foi feito para te ajudar a reverter erros. Explore e aprenda com eles!

Bom Código!