

# Manual de Comandos Git - Guia Rápido

## **git status**

Exibe o estado atual do repositório, mostrando arquivos modificados, novos arquivos não adicionados, e arquivos prontos para commit.

Comando:

```
git status
```

Exemplo de uso:

```
git status
```

Use sempre antes de dar git add ou git commit, para ver o que mudou.

## **git branch**

Lista as branches existentes e mostra qual está ativa.

Comando:

```
git branch
```

Criar uma nova branch:

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

Exemplo:

```
git branch feature-login
```

Dica: não muda de branch automaticamente! Para isso, use git checkout ou git switch.

## **git add**

Adiciona arquivos à área de stage.

Comando:

# Manual de Comandos Git - Guia Rápido

```
git add nome-do-arquivo
```

Adicionar todos os arquivos:

```
git add .
```

Exemplos:

```
git add index.html
```

```
git add .
```

Dica: o ponto (.) adiciona todos os arquivos modificados.

## **git commit**

Salva as mudanças adicionadas com uma mensagem.

Comando:

```
git commit -m "mensagem"
```

Exemplo:

```
git commit -m "Adiciona funcionalidade de login"
```

Dica: mensagens claras facilitam o entendimento do histórico.

## **git stash**

Guarda temporariamente alterações não commitadas.

Comando:

```
git stash
```

Recuperar alterações:

```
git stash pop
```

# Manual de Comandos Git - Guia Rápido

Dica: ideal quando precisa trocar de branch sem perder progresso.

## **git push**

Envia os commits locais para o repositório remoto.

Comando:

```
git push origin nome-da-branch
```

Exemplo:

```
git push origin main
```

Dica: use sempre após commits importantes.

## **git pull**

Puxa alterações do repositório remoto.

Comando:

```
git pull origin nome-da-branch
```

Exemplo:

```
git pull origin main
```

Dica: use antes de começar a trabalhar para evitar conflitos.