

CONFIGURAÇÃO

<code>git config --global user.name "nome_do_usuario"</code>	Faz a configuração do nome do usuário que será usado nos commits.
<code>git config --global user.email "email_do_usuario"</code>	Faz a configuração do email do usuário que será usado nos commits.
<code>git config --list</code>	Listagem de configurações e valores Git.

COMANDOS DIÁRIOS

<code>git init</code>	Inicializa um repositório Git.
<code>git status</code>	Exibe o estado atual do repositório, mostrando arquivos modificados, preparados para commit e não rastreados.
<code>git add <arquivo></code>	Adiciona um arquivo na área preparação, preparando-o para commit.
<code>git add .</code>	Adiciona todos os arquivos na área preparação, preparando-o para commit.
<code>git commit -m "mensagem"</code>	Registra as mudanças adicionadas ao repositório com uma mensagem.

INSPEÇÃO DE ALTERAÇÕES

<code>git diff</code>	Exibe as diferenças entre o diretório de trabalho e o último commit.
<code>git diff --cached</code>	Mostra diferenças entre arquivos na área de preparação e o último commit.
<code>git diff --staged</code>	Igual ao de cima. Mostra diferenças entre arquivos na área de preparação e o último commit.

REMOÇÃO E RESTAURAÇÃO

<code>git rm --cached -r .</code>	Remove arquivos e diretórios da área de preparação, mantendo-os no diretório de trabalho.
<code>git restore <arquivo></code>	Restaura um arquivo modificado para a versão do último commit.
<code>git restore --stage <arquivo></code>	Retira um arquivo do staging sem perder as alterações.
<code>git checkout <arquivo></code>	Restaura um arquivo para a versão do último commit.
<code>git clean -f</code>	Remove arquivos não rastreados do diretório de trabalho.
<code>git reset --hard</code>	Descarta todas as alterações locais e volta ao último commit.

HISTÓRICO

<code>git log</code>	Mostra o histórico de commits.
<code>git log --oneline</code>	Exibe um resumo dos commits (um por linha).
<code>git log --patched</code>	Mostra detalhes das mudanças feitas em cada commit.
<code>git log --stat</code>	Exibe estatísticas das alterações realizadas.



ALTERAÇÃO DE COMMITS

<code>git commit --amend</code>	Modifica o último commit, permitindo adicionar arquivos ou alterar a mensagem.
<code>git commit --no-edit</code>	Faz um amend sem alterar a mensagem do commit anterior.

REPOSITÓRIOS REMOTOS

<code>git remote -v</code>	Exibe as URLs dos repositórios remotos associados ao repositório local.
<code>git remote add origin <url></code>	Adiciona um repositório remoto com o nome <i>origin</i> e a URL especificada.
<code>git remote set-url origin <url></code>	Modifica a URL de um repositório remoto já existente, no caso, <i>origin</i> .
<code>git push</code>	Envia as alterações locais para o repositório remoto.
<code>git pull</code>	Baixa e mescla alterações do repositório remoto para o repositório local.
<code>git push --force</code>	Força o envio das alterações para o repositório remoto, sobrescrevendo o histórico de commits no remoto.
<code>git push --force-with-lease</code>	Força o envio das alterações, mas verifica se o repositório remoto não foi alterado antes de sobrescrever o histórico.

MESCLAGEM DE BRANCHES

<code>git merge <branch></code>	Mescla as alterações de uma branch especificada para a branch atual.
---------------------------------------	--

BRANCHES

<code>git branch --list</code>	Exibe a lista de branches locais no repositório.
<code>git branch <nova_branch></code>	Cria uma nova branch com o nome especificado, sem trocar para ela.
<code>git checkout <branch></code>	Altera para a branch especificada.
<code>git checkout -b <nova_branch></code>	Cria uma nova branch e já muda para ela.
<code>git checkout -f <branch></code>	Força a troca para a branch especificada, descartando alterações não commitadas.
<code>git switch <branch></code>	Alterna para a branch especificada, alternativa ao <i>git checkout</i> .
<code>git switch -</code>	Volta para a última branch onde estava.
<code>git switch -c <nova_branch></code>	Cria uma nova branch e já muda para ela, equivalente ao <i>git checkout -b</i> .
<code>git branch -d <nome_branch></code>	Exclui a branch local especificada, se ela já tiver sido mesclada.
<code>git push --delete origin <nome_branch></code>	Remove a branch especificada do repositório remoto.
<code>git branch -m <novo_nome></code>	Renomeia a branch atual para o novo nome.
<code>git branch -m <nome_antigo> <novo_nome></code>	Renomeia a branch especificada de <nome_antigo> para <novo_nome>.
<code>git branch -a</code>	Exibe todas as branches, locais e remotas.



TAGS

<code>git tag <nome></code>	Cria uma tag do tipo lightweight (sem metadados), associada ao último commit.
<code>git tag -a -m "mensagem" <nome></code>	Cria uma tag annotated (com metadados) e adiciona uma mensagem de anotação.
<code>git show <tag></code>	Exibe informações sobre uma tag específica, incluindo o commit ao qual está associada.
<code>git tag --list</code>	Exibe todas as tags presentes no repositório.
<code>git tag -n</code>	Exibe todas as tags com suas respectivas mensagens (se houver).
<code>git tag -d <nome></code>	Remove uma tag localmente.
<code>git push --delete origin <nome_tag></code>	Remove uma tag do repositório remoto.

MODIFICAÇÃO DE HISTÓRICO

<code>git revert <commit></code>	Cria um novo commit que desfaz as alterações de um commit anterior.
<code>git revert HEAD --no-edit</code>	Reverte o último commit sem modificar a mensagem de commit.
<code>git rebase <branch></code>	Reaplica commits da branch atual sobre a branch especificada.
<code>git rebase --abort</code>	Aborta o processo de rebase em andamento, retornando ao estado anterior.
<code>git rebase --continue</code>	Continua o rebase após a resolução de conflitos.

STASH

<code>git stash</code>	Armazena temporariamente alterações não commitadas, deixando o diretório de trabalho limpo.
<code>git stash list</code>	Exibe a lista de stashes armazenados.
<code>git stash apply</code>	Aplica o stash mais recente ao diretório de trabalho, sem removê-lo da lista de stashes.
<code>git stash apply stash@<posicao></code>	Aplica um stash específico da lista, indicado pela posição.
<code>git stash pop <posicao></code>	Aplica e remove o stash especificado da lista de stashes.
<code>git stash drop <posicao></code>	Remove um stash específico da lista, sem aplicá-lo.

BISECT (bissecção)

<code>git bisect start</code>	Inicia uma sessão de bisecting para identificar o commit que introduziu um erro.
<code>git bisect good <hash></code>	Marca o commit especificado como "bom" durante a bissecção.
<code>git bisect bad <hash></code>	Marca o commit especificado como "ruim" durante a bissecção.
<code>git bisect good</code>	Marca o commit atual como "bom" no processo de bissecção.
<code>git bisect bad</code>	Marca o commit atual como "ruim" no processo de bissecção.



FETCH

<code>git fetch</code>	Baixa as atualizações de repositórios remotos, mas não as mescla com a branch atual.
------------------------	--

APLICAÇÃO DE COMMIT

<code>git cherry-pick <commit></code>	Aplica as mudanças de um commit específico da outra branch à branch atual.
---	--

ALIAS

<code>git config --global alias.<abreviação> <comando></code>	Cria um alias global, permitindo usar <abreviação> no lugar de <comando>.
<code>git config --global --unset alias.<abreviacao></code>	Remove o alias global especificado.