Git - Comandos explicados

git config

git config --global user.name "nome_do_usuario"

Faz a configuração do nome de usuário.

git config --global user.email "email_do_usuario"

Faz a configuração do e-mail do usuário.

git config --global --list

Faz a listagem dos valores atribuídos na configuração.

git init

git init

Inicializa repositório git. O diretório usado para a criação não precisa necessariamente estar vazio, pode existir arquivos antigos. Isso torna possível versionar um projeto mesmo que já esteja criado.

git add

git add --all

Adiciona todos os arquivos (modificados, alterados e removidos) na área de preparação (staging area) e os deixa preparado para o commit e indica ao git que esses arquivos serão rastreados.

git add <nome_do_arquivo.extensao>

Adiciona arquivo único na área de preparação (staging area).

git commit

> *git commit -m "mensagem_commit"*

Salva mudanças no repositório local e adiciona mensagem de commit.

git status

> *git status*

Verifica quais os estados que os arquivos estão. Se existe algum arquivo modificado, adiciona ou removido. Além disso, caso esteja rastreado com uma branch no servidor, verifica se a versão está a frente ou atrás da versão do servidor.

git log

> *git log*

Mostra histórico de alterações em ordem cronológica.

> *git log --stat*

Mostra histórico de alterações em ordem cronológica e quais arquivos foram alterados.

git diff

> *git diff <commit_1> <commit_2>*

Faz a comparação entre dois commits. Mostra quais foram os arquivos alterados, novos e removidos. Além disso, mostra também quais foram as linhas alteradas.

git checkout

> *git checkout <name>*

Caso o <name> seja um branch:

- Muda o código para a branch selecionada. Alterações devem ser commitadas antes de fazer a troca de branch.

Caso o <name> seja um arquivo:

- Desfaz as alterações no arquivo, porém arquivo já deve ter sido adicionado nas mudanças.

```
git checkout -b <branch_name>
```

Cria uma branch e muda o código para a branch selecionada.

git reset

```
git reset --hard
```

Desfaz todas as alterações que aconteceram em arquivos rastreados.

```
git reset HEAD~1
```

Desfaz o último commit, porém mantém as alterações feitas.

```
git reset --hard HEAD~1
```

Desfaz o último commit feito e também desfaz todas as alterações feitas.

git clean

```
git clean -n
```

Mostra uma lista de arquivos que serão apagados.

```
git clean -f
```

Apaga todos os novos arquivos não rastreados na alteração.

```
git clean -n
```

Mostra uma lista de opções para apagar somente alguns arquivos.

git clone

```
git clone <origem>
```

Faz a clonagem do repositório para a pasta corrente. O valor da origiem pode ser uma pasta local ou pode ser uma URL para o código ser baixado.

git pull

git pull

Atualiza repositório local com a última versão da origem da branch. Necessário commitar as mudanças para executar essa ação.

Se for no servidor:

- Branch já deve estar rastreado no servidor. Assim, o git sabe que deve comparar a versão local com o servidor.

Se for no repositório local:

- Deve ter sido clonado de outra pasta para a pasta destino. Só assim, o repositório terá um 'pai' que poderá ser verificado para atualização de novas informações.

git branch

git branch

Faz a listagem de todas as branchs locais.

git branch -r

Faz a listagem de todas as branchs remotas.

git branch -a

Faz a listagem de todas as branchs locais e remotas.

git branch -d <branch_name>

Faz a remoção de um branch local.

git branch -D <branch_name>

Força a remoção de um branch local. Esse comando é necessário, pois caso exista uma branch que ainda não foi feito o merge, o git notifica e não permite apagar a branch somente com o comando 'git branch -d'.

git push

git push

Faz o envio das mudanças comitadas localmente para a origem da branch rastreada.

git push origin <nome_da_branch>

Faz o envio das mudanças comitadas localmente para o servidor pela primeira vez. Caso a branch que está sendo enviada não exista no servidor, é criada e então a branch está configurada para ser rastreada com essa origem no servidor.

A partir desse primeiro comando, para versionar próximas mudanças basta que seja feito o comando 'git push' descrito acima.

git merge

git merge <nome_da_branch>

Mescla as mudanças presentes na <nome_da_branch> na branch corrente.

git tag

git tag

Faz a listagem de tags.

git tag <nome_da_tag>

Faz a criação de uma tag no último commit da branch corrente.

git tag <nome_da_tag> <hash_do_commit>

Faz a criação de uma tag no commit específico.

git push --tags

Faz o envio das tags locais para o servidor.

git tag -d <nome_da_tag>

Faz a remoção da tag localmente.

git push origin --delete <nome_da_tag>

Faz a remoção de tag no servidor.

git checkout -b <branch_a_partir_da_tag> <nome_da_tag>

Faz a criação de uma branch a partir de uma tag.

alias

git config --global alias.<abreviacao> <comando_git>

Cria uma abreviação para o comando do git.

Ex: git config --global alias.st status

Cria um atalho para o comando status. Dessa maneira esse comando pode ser executado como 'git st'.

git config --global --unset alias.<abreviacao>

Faz a remoção do alias cadastrado.

git config --global alias.<abreviacao>

Verifica o comando que foi cadastrado.

git config --get-regexp

Faz a listagem de todos os alias configurados.