**Branch > são versões diferentes do seu sistema;**

* **git branch** lista em qual branch você está. A opção irá ficar marcada com um asterisco. \*

**Após criar um arquivo, dentro do repositório do git você deve inserir o comando:**

* **git status** para vizualizar os aruivos na pasta

**Para adicionar os arquivos não traqueados vc utiliza:**

* **git add** *arquivo\_a\_ser\_usado*
* **git add -A** para adicionar todos os arquivos.
* **git add** meuarquivo\* só adiciona os arquivos que comecem com o que vem antes do asteristico (Ex: meuarquivo.html, meuarquivo/fotos/iamgem.png e etc.

**Para voltar mudanças feitas antes de vc adicionar os arquivos:**

* **git checkout** - -. ou - - \* ou - - nome-do-arquivo

**Para desfazer o add de um arquivo:**

* **git restore** - -staged nome-do-arquivo

**Realizar um commit:**

* **git commit -m** **"**frase desejada para identificara as alterações ou inserções realizadas**"**

**Para adicionar modificações ao ultimo commit:**

* **git commit - - amend** –m **"** dfsdgfdsggfdd**"** ele vai adicionar os últimos arquivos add ao ultimo commit já realizado. Ou simplesmente para mudar o titulo do ultimo commit que pode etr sido digitado ou especificado de erroneamente.

**realizar um commit e já adicionar os arquivos não trackeados:**

* **git commit -am** **"**frase desejada para identificar as alterações realizadas**"**

**Vizualizar todos os commits feitos o branch:**

* **git log**

**Para reverter uma modificação:**

* **git reset - -soft** 00000000000000000000000000000 (código da versão) volta para a versão antes do commit (você ainda terá acesso às informações editadas).
* **git reset - -mixed** 00000000000000000000000000000 (código da versão) semelhante ao soft mas não vai estar preparado para commitar.
* **git reset - -hard** 00000000000000000000000000000 (código da versão) vai deletar totalmente o arquivo para a versão anterior.

**Criar um novo branch:**

* **git branch** *nome-do-novo-branch*

**Mudar de branch:**

* **git checkout** *nome-do-branch*

**Mudar de branch criando uma nova branch:**

* **git checkout** –b *nome-da-nova-branch*

**Excluir branch:**

* **git branch** -d *nome-da-branch*

**Exibir o que realmente foi alterado** (detalha o que foi removido ou alterado) (recomendado antes de realizar um commit se já tiver um tempo que vc não usa o branch):

* **git diff**

**Exibir somente o nome dos arquivos que foram modificados:**

* **git diff - -name-only**

**Exibir somente o que foi modificado em determinado arquivo:**

* **git diff** *nome-do-arquivo*

**Voltar somente um arquivo atrás:**

* **git checkout HEAD - - nome-do-arquivo**

**Para enviar o projeto para o remoto:**

* **git push** *nome-do-master* **master**

**Para excluir um remote:**

* **git remote –v** (para vizualizar os remotes).
* **git remote rm destination** (destination se refere ao inicio do project (**origin** http://....... , **old-origin** http://..............). EX: **git remote rm origin**
* **git remote –v** (para conferir se a exclusão foi bem sucedida).

**Para ignorar um arquivo no git:**

* Deve-se criar um arquivo .gitignore .
* Dentro do arquivo colocar o nome dos arquivos que deseja ocultar (ex: estilo.css, index.html, banco-de-senhas).
* Para ignorar todos os arquivos de uma extensão deve-se usar o seguinte \*.html \*.txt \*.css .

**Para reverter um arquivo no git sem apagar o commit novo:**

* git revert --no-edit *código-do-commit*

**Para excluir um branch remoto:**

* **git push :** *nome-do-branch*

**Para excluir um branch local:**

* **git branch -D** *nome-do-branch*

**Para fazer um pull:**

* **git pull origin máster**

**Para clonar um repositório remoto:**

* **git clone** *url-do-repositório*

**Para mesclar duas branchs locais:**

* **git merge**

**Para colaborar com outro projeto:**

* Ir até a página aonde o projeto esta
* Clicar em fork
* Copiar a url que vai ser gerada do projeto em sua própria conta
* Cria um clone através dessa url gerada
* Realiza as edições necessárias
* Faz um push ( necessário visualizar o servidor remoto dele através do git remote -v)
* E observar o nome do branch (possivelmente estará modificado)
* Vc deverá ir em New pull rquest
* Visualizar as informações
* Create pull request
* Colocar os argumentos de confirmar.

Tu pode da git pull origin master para a sua branch local, e depois dar git push origin suabranch -f para sobrescrever tudo. É recomendado que este comando seja rodado apenas na branch do usuário, e não na máster ou em outra branch compartilhada.

Quando vc faz um pull request, ele vai mesclar o que esta no repositório remoto com o conteúdo do repositório local, isso gera um commit novo pra descrever o porquê desta ação

Criar uma pasta qualquer , criar uma branch, clonar o repositório, realizar as alterações no repositório, subir as alterações com git push

**Estudar os comandos:**

**Git merge**

**Git stash**

**Git rebase**

**Git chery-pick**

**Git bysect**