Branch ---> Nome dado á diferentes versões do sistema

master ---> Nome da versão principal do sistema

comit ---> Nome dados á ação de enviar alterações no sistema (geralmente envia junto com um comentario)

README.md ---> Um arquivo que existe em todos os repositorios, basicamente a primeira coisa que alguem vai ler qunaod entrar no seu repositorio então é importante ter informações como: comentarios, instruções e etc sobre seu projeto

-=-

git init ---> transformar projeto em um git repositorio

git commit (-m ou -am) "(comentario)" ---> envia as alterações feitas no sistema para o projeto hospedado no git

git status ---> lista todos os arquivos que sofreram alterações, ele faz isso comparando o projeto atual ao commit anterior

git add (nome do arquivo n monitorado) // git add -A ---> é usado para adicionar arquivos que não estão sendo monitorado pelo seu git (ou seja q estão fora do projeto) para dentro do repositorio do seu projeto

git log ---> lista quantas vezes houve um commit em uma branch que está sendo trabalhada

git branch ---> lista quantos branchs do seu projeto foram criados e em qual que você esta

-=-

git reset --soft (id do commit anterior) ---> retorna seu projeto para o comit anterior (ex commit 0), o projeto sofre esse retorno com o commit 0 já setado com os codigos do commit 1 prontos para serem commitados, ou seja ele simplesmente volta para o comit 0 só que com todas as alterações do commit 1 já feita, apénas retornando o projeto para antes do comando commit ser invocado só q com o comando git.add já adicionado

git reset --mixed (id do commit anterior) ---> volta para o commit anterior igual o git.reset -soft mas com o detalhe de que o comando git.add ainda não foi chamado ou seja: ainda não está pronto para ser commitado

git reset --hard (id do commit anterior) ---> volta para o commit anterior sem levar em conta nenhuma alteração ou add feita, simplesmente volta o projeto para o commit anterior

-=-

git branch (nome do novo branch) ---> cria um novo branch do projeto que se inicia apartir do ultimo commit do branch aterior Um branch novo só tem acesso aos seus proprios commits.

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git checkout (nome do branch) --> alterna entre os branchs e gera os arquivos do projeto exatamente como ele estava em seu ultimo commit (Lembrando que é possivel alternar entre os branchs livremente).

git checkout (nome do branch) -- (Nome do arquivo expecifico) --> Modifica o estado de algum arquivo expecifico para como ele estava em algum outro branch exemplo de ultilização da função acima: no branch pato eu tenho os arquivos 1 e 2 porem eu não gostei das modificações realizadas no arquivo 1 no meu branch atual e quero retornar apénas o arquivo 1 para como ele estava no branch anterior ai eu uso está função

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git diff ---> diferente do git status que só fala quais arquivos foram alterados comparado ao commit anterior o diff detalha exatamente quais mudanças foram feitas em cada um dos arquivos que foram modificados

git diff (nome do arquivo obtido pelo git diff --name-only) ---> saber absolutamente todas as alterações ocorrida em um arquivo expecifico do projeto

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

ssh-keygen -t rsa -b 4096 -C ("email da sua conta git") ---> para sincronizar o seu pc/seus repositorios locais com o github é necessario usar este comando, ele vai gerar duas chaves criptografadas com chaves diferentes porem que em seu final compartilham o mesmo codigo a chave publica e a privada. A chave publica deve ser inserida no github (O github vai bater o codigo que esta na chave com o msm codigo que está no pc e ver se é o msm e se for ele vai dar acesso)

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git remote add (nome da origem) (link de onde vai estar seu repositorio no github).git ---> Este comando está adicionando uma segunda origem ao seu repositorio local, está origem sera remota e interligada ao github

git remote -v ---> Retorna todos os detalhes e informações da conexão entre o repositorio local e remoto

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

origem ---> nome do canal de ligação entre o nosso repositorio local e o remoto

push ---> o ato de enviar alterações do repositorio local para o repositorio remoto

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git push -u (origem) (branch local) ---> Cria um branch no repositorio remoto do github com todas as informações do branch atual do nosso repositorio local.

git push (origem) (branch remoto) ---> Envia as alterações realizadas no nosso branch local para o branch remoto

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git revert ---> Diferente do git reset que apaga o seu commit quando retorna ele no tempo o git revert volta para o commit anterior porem ele salva o commit que você estava trabalhando para que você possa acessa-lo depois

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git push (nome de origem) :(Nome do branch remoto) ---> Deletar um branch remoto

git branch -D (Nome da branch) ---> É usado para deletar um branch local, lembrando que você não pode estar dentro da branch que você esta apagando

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git pull (Nome de origem) (branch local) ---> puxa as alterações do diretorio remoto para o local

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

git clone (Url do repositorio) ---> Para clonar algum repositorio publico do github para dentro do seu branch local (exemplo, você quer criar sua propria versão do linux, você entra no repositorio onde está o sistema linux e clona ele para seu branch local para poder trabalhar nele)]

-==-=-=-=--=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-