Comandos do Git(Atualização do Sicoob)

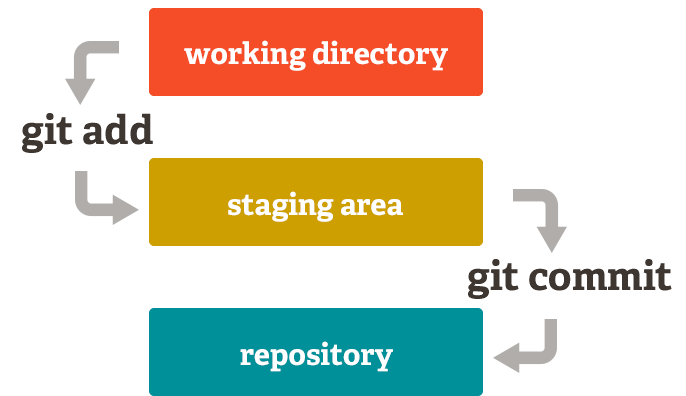
Comandos Gerais

**git version** = Indica a versão do git

**git status** = mostra arquivos fora do container

--Trabalhando inicialmente de maneira local

**git init** -- Cria um ponto inicial na pasta local da máquina criando um arquivo .git dentro do diretório de maneira centralizada.



As 3 fases acima de arquivo indicam o seu loca de armazenamento de acordo com as cores.

O staging área seria o container ou o index do commit, não se pode fazer commit sem os arquivos estarem nessa área.

O **git status** só mostra os dados dentro do staging área

**git add** -A coloca todos os arquivos prontos para o commit.

**Git commit** -m “Mensagem Para o Commit”(Colocar no repository)

**Git log** = Mostra a lista de commits realizados, uma opção legal é

**Git log –oneline** = facilita a visualização em uma linha

**Git log –oneline –decorate**  = facilita a visualização em uma linha e mostra (HEAD/Master/Branch)

**Git branch** = indica a branch selecionada

**Git branch “nome da branch”** = vão ser 2 grafos apontando para o mesmo commit, tanto o master quanto o branch

**Git checkout “nome da branch”=** vai chavear a branch para a nova branch e mostrar os arquivos dessa branch

**Git log –oneline –decorate**  **--all** = mostra o log da branch que estou chaveado e de todas as outras

**Git log –oneline –decorate**  **--all --graph** = mostra o log da branch que estou chaveado e de todas as outras além de mostrar as ilhas ou os caminhos de divergência entre as branches

Realização de Merge

Imaginemos que estou em master e quero atualizá-la de acordo com a branch feature1

**Git merge feature 1**

**q = dentro do merge é a atualização de referência onde não existem conflitos**

**Git branch -d “nome da branch” = Apaga a branch**

**Git checkout -b “nomedabranch”** = Cria a branch e chaveia.

Um git merge com **estratégia** **recursiva** indica o alinhamento entre a branch master e a branch sem divergências dentro do mesmo arquivo.

Trabalhando com Stash

Exemplo: Ele salva os arquivos que você ainda não tem certeza que vão subir, apenas para não perder os arquivos.

**Trabalho com Repositório remoto**

**Push** = Atualizar um sistema com referências remotas.

**Git remote add origin url**

**Git remote** = mostra o repositório remoto com parâmetro -v olha mais detalhes

**Git push origin master** (eu vou empurrar o meu grafo de master para o origin no servidor remoto)

**Pull** = Puxa do sistema remoto todas as referências para a local

**Git pull origin master(Cuidado o comando o comando faz merge automático)**

**Git clone url caminholocal**

**Normalmente para não usar o git pull usa-se o git fetch(realiza o pool pela metade) em**

**Pode-se verificar no status que existem diferenças entre a branch master e a sua origin/master normalmente em seguida usa-se git diff master origin/master e resolve os conflitos com git merge origin/master (rever a aula 4)**

**Desfazendo as atualizações**

**Git reset(soft,mixed,hard) terminar o ultimo vídeo 5 e realizar os testes no vídeo 4.**