**Comandos básicos para Git**

*(Es un software de control de versiones diseñado por Linux)*

$ git init : comando que permite iniciar git. Esto crea un nuevo subdirectorio llamado .git que contiene todos los archivos necesarios del repositorio. Crea un repositorio

$ git status : comando que permite saber el estado de nuestro repositorio. Determina que estado tienen los archivos es el comando git status.

$ git add : este comando actualiza el índice con el contenido actual que se encuentra en el árbol de trabajo, para preparar el contenido organizado para el próximo commit. Este se realiza antes de hacer un commit.

$ git add . : colocar todos los archivos modificados al índice.

$ git add <archivo> : coloca solo el archivo mencionado al índice.

$ git commit : permite guardar los cambios realizados en el registro del repositorio.

$ git commit –amend : permite guardar el cambio en el utimo registro guardado sin necesidad de crear uno nuevo registro en el repositorio.

$ git log : muestrar los commit registrados o historial de commit log.

$ git log –oneline : muestra los commit en una línea y su código de 7 digitos de identificación

$ git log –oneline | cat : para sistemas operativos unix

git log --oneline -–decorate : sirve para ver el nombre de los punteros en el repositorio o commit. Ejemplo: (HEAD -> master)

$ git diff : muestra una serie de cambios realizados a los archivos del proyecto, cambios que no estén en el staging o índice. Muestra los cambios entre el árbol de trabajo y el índice o un árbol, los cambios entre el índice y un árbol, los cambios entre dos árboles, los cambios entre dos objetos blob o cambios entre dos archivos en el disco.

$ git diff HEAD~<numero>: sintaxis para referir a los commit anteriores al HEAD ejemplo: $ git diff HEAD~1 HEAD ( ~ significa anterior )

$ git checkout : comando que permite realizar varias tareas

$ git checkout -- <archvio> : permite deshacer los cambios en el directorio de trabajo. Deshace la última modificación del archivo esto en el caso que el status este en rojo.

Ejemplo: $ git checkout -- index.html

$ git reset HEAD <archvio> : permite deshacer los cambios, esto a nivel que se haya ejecutado $ git add. Al hacer este cambio ya se puede utilizar el $ git checkout -- <archvio>

$ git reset <código del commit> : permite borrar un determinado commit, en este caso debe de exitir el commit, hay que tener cuidado porque este comando es dañino o destructivo y borra en caso de trabajar local no hay problema pero en caso de trabajar con un grupo en la red puede ocasionar problemas al sincronizar commit de otras personas para ello se pide utilizar otro comando.

$ git reset --hard <código del commit> : elimina el commit tanto del staging o índice como del cambio local

$ git reset --soft <código del commit> : elimina el commit y deja la información en el índice o staging

$ git revert HEAD : revierte los cambios de un commit especifico

$ git revert --no-commit HEAD : revierte el cambio y se muestra en el índice o staging dejando el mismo commit

$ git revert –-continue : comando que permite teminar la reversión realizada con el comando anterior.