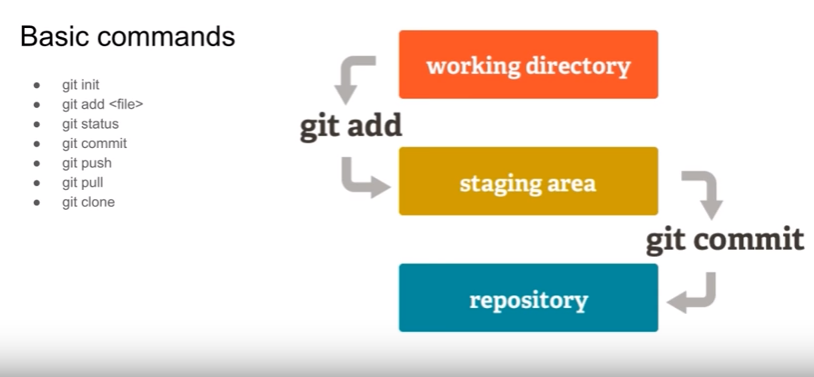
**GIT RESUMEN COMANDOS Y PROCESO** **https://www.youtube.com/watch?v=HiXLkL42tMU**



**Working Directory** = directorio donde se está trabajando

**Staging Area** = son todos los archivos que después se van a guardar con el commit

**Repository** = Repositorio donde finalmente se guardar los archivos

**Git init** = iniciar un proyecto

**Git add <file>** = para agregar a la staging area (el nombre del archivo debe ser tal cual respetando extensión y mayúsculas / minúsculas)

**Git status** = para ver en status están los archivos, si están en el working directory o staging area

**Git commit** = para pasar del staging area al repositorio, se crea el snapshot

**Git push** = para subirlo a un repositorio remoto (github)

**Git pull** = para bajar los cambios de un repositorio remoto (github)

**Git clone** = clona desde el servidor central para comenzar a trabajar

**Git log** = para ver todos los commits hechos

**git diff =** cambios de archivo modificado

git branch = muestras las ramas de archivos e indica en color en que branch estoy

Desde la consola (bash):

Ls = listar directorios y archivos

Cd = cambiar directorio

Pwd = para ver en que directorio estamos actualmente

Proceso:

// Se puede iniciar git con doble clic derecho sobre carpeta donde está el proyecto

Una vez que estoy en la carpeta del proyecto, para **comenzar** usar git:

Git init

Con ls vemos los archivos, y con

Git status vemos en que estado están los archivos, si están en working directory o staging area

Para comenzar a seguirlos git add NombreArchivo.html y asi con todos lo que se quieran dps versionar ( para agregar a todos de una vez usamos git add . //punto)

Para crear una **primera versión: git commit**

// la primera vez pide que se defina el usuario con su mail y nombredeusuario

hp@LAPTOP-NKOG29TJ MINGW64 /c/jquery214 (master)

$ git config --global user.email "dgale11@gmail.com"

hp@LAPTOP-NKOG29TJ MINGW64 /c/jquery214 (master)

$ git config --global user.name "DiegoD"

Si solo escribo git commit me lleva a otra pantalla donde con cuidado pongo el nombre del commit, como por ejemplo MiPrimerCommit (para escribir tengo que presionar primero la **letra i**) dps presiono Esc y escribo al final **:qw** para volver a la pantalla principal

(// para evitar ir a la 2da pantalla puedo usar **git commit –m “MiPrimerCommit”** )

Si modifico un archivo que ya estoy siguiendo (staging area) y luego chequeo el status con git status me va a indicar que fue modificado e indicar dos opciones: descartar o agregar para futuro commit

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

Si uso checkout vuelve el archivo a como estaba antes y no se puede deshacer si el archivo no fue commiteado antes, es decir si no hay una versión anterior.

Si queremos ver los cambios hechos en el archivo solo ingresamos **git diff**

Para agregar los cambios hacemos git add nombrearchivo

Si creamos un archivo con nombre **.gitignore** en el podemos colocar todos los nombres de archivos o carpetas que se van a ignorar porque no nos interesa conocer su status (agregamos git add .gitignore)

**Branch** :

Git branch me muestras las ramas, por defecto la primera es master

Si quiero crear otra rama git branch NombreDeBranch

Si quiero pasar a esa rama git checkout NombreDeBranch

**GITHUB**

Es un repositorio online open source, no se puede tener un repositorio privado en forma gratuita pero sí se puede tener un repositorio publico gratuitamente en el que pueden participar colaboradores en forma ilimitada.

**Git remote add origin** https github/dgale11/usandoGit.git

Para ingresar los archivos al repositorio

**Git push –u origin** master