-Abrir Bash de GIT

-Comandos:

Pwd: te indica el directorio donde estas ubicado

cd: te lleva al directorio donde querés crear el repositorio

ls: lista los archivos que hay en el directorio seleccionado

git init: crea el área de trabajo (repositorio, terminal) y lo crea en donde estas ubicado (pwd). Esto crea un nuevo subdirectorio llamado .git que contiene todos los archivos necesarios del repositorio —un esqueleto de un repositorio Git.

git add seguido del nombre del archivo: lo agrega a staging área (al hacer git add, git lo/s tiene en seguimiento constante, y cada vez que hagamos un git status no indicara en que estado se encuentra)

git add . : para agregar todos los archivos masivamente, es decir sirve cuando tenemos muchos archivos para no tener que agregarlos uno por uno.

git status: indica los archivos que tenemos / trabajando y el estado de los mismos.

Luego de un git status:

- aquellos nombres que finalizan con un slash al final “/ “ son carpetas: ejemplo de una carpeta “nada/ “

- aquellos nombres que finalizan con punto(.)seguido de la extensión son archivos.

git ignore: Sirve para ocultar archivos. Si no quieres cargar algunos archivos y que git tampoco los reconozca podemos hacer 2 cosas:

1. A cada archivo podemos ignorarlo para que no los reconozca (ignore) ya que no tiene sentido o simplemente no queremos que los cargue.
2. Cuando tenemos 2 o más archivos / carpetas a ignorar podemos hacer lo siguiente para hacerlo más fácil: Crear un archivo o carpeta (Según) con el siguiente nombre .gitignore dentro de ella colocamos el/los nombres de todos los archivos y/o carpetas que queramos que ignore. Cuando hagamos un git status los archivos que nombramos dentro de este archivo los ignorará pero si traerá este archivo y para que tampoco lo traiga hay que hacer con este archivo lo del punto 1

para todo el sistema operativo, cada vez que hagamos una modificación se necesita indicar quien está haciendo la configuración

git config –global user.email : entre comillas el email

git config –global user.name : entre comillas el nombre

Pata hacer commit , necesitamos tene configurado el usuario y email , esto dará las indicaciones a git de quien es el que esta haciendo las modificaciones

git commit : realiza el snapshot (instantánea , foto ,punto de partida ) de los archivos agregados al staging área ( git add). Este git commit nos lleva al editor de Windows indicado al momento de la instalación , en este caso llamado vim

Luego nos pide que ingresemos un texto para identificar los cambios y guardarlos , para ingresar este texto debemos presionamos la “ i “ (i latina) para que nos permita escribir el texto en este editor que no es el del GIT , sino que es el editor de windos por defecti VIM. Luego de finalizar presionamos: ALT + (barra invertida) (nos permite acceder para tipear los comandos en el editor vit y grabar los cambios , para ello presionamos “:wq” (w (write) y q (quit))

Git commit – m “ el mensaje del commit” ( esto es como un atajo ya que no nos abre el editor de Windows y por lo tanto es mas rápido)

Git checkout - - el nombre del archivo: Deshacer cambios en un archivo:

Git diff nombre del archivo: Ver las diferencias entre lo ya cargado y lo modificado:

Git log nos trae el historial de commit

Crear versiones /ramas

Git Branch : indica en que versión estamos posicionados

Git Branch + nombre : crea la nueva versión / rama

Git checkout + nombre de la rama : Cambia la dirección en la cual estamos posicionados para hacer la nueva versión / actualización.

**GIT HUB**

**Git clone**

Para subir los proyectos a la nube tenemos que ingresar a :

Ingresamos a:

1. Your repositories
2. Create a new repository
3. Luego Obtenemos una dirección web que es el enlace al repositorio en Git hub
4. Para poder conectar la pc con el repositorio de Gut Hub en la nube demos tipear en el Bash de git en la PC lo siguiente:
5. git remote add origin + el enlace del repositorio en la nube , en este caso <https://github.com/monacog/git-course.git> ( con este paso ya esta conectado la pc con la nube Git Hub).
6. Luego , cualquier modificación que hagamos podemos hacer un Git commit -m ‘ message’, esto lo que hace es actualizaron en el repositorio de la PC y para subirlo a Git Hub (repositorio creado en la nube)hacemos lo siguiente:
7. git push -u + el Branch ej: origin master o versión2 o la rama que deseemos siempre y cuando la hayamos creado

Formato de escritura : MARKDOWN