HEAD = Ultimo commit en el Branch actual.

Cat = ver contenidos de un archivo

Pwd: Ver la carpeta en donde te encuentras ahora.

Ls -al = Ver lista de archivos en la carpeta

Rm -rf [archivo | carpeta] = borra lo seleccionado

Npp ~/.gitconfig = abre el archivo gitconfig. (Después de asignar un editor de texto)

git config –global user.name [nombre] = Configura globalmente el nombre del usuario

git config –global user.name [[email]](mailto:eroscamposg@gmail.com) = Configura globalmente el email.

Git init = Inicializa un nuevo repositorio en el folder actual.

Git add [] = Agrega archivos al “Staging Area”

Git add -A = Agrega todos los archivos al “Staging Area” y además comprueba los archivos que han sido movidos, renombrados o borrados.

Git branch = Muestra los branch locales.

Git branch -m [nombre\_branch] [nuevo\_nombre] = Cambia el nombre de un branch.

Git branch -d [nombre\_branch] = Borra el Branch seleccionado.

Git checkout [branch] = Cambia al branch especificado.

Git checkout -b [branch] = Crea y Cambia al branch especificado.

Git merge [branch] = Junta los cambios hechos en el Branch especificado hacia el Branch donde te encuentras actualmente.

Git merge [branch] –no-ff = Lo mismo, pero sin fast forward.

Git mergetool = PARA RESOLVER COMFLICTOS.

Git rebase [branch] = Manda los cambios que ahí en el actual Branch donde te encuentras, hacia el Branch especificado en el comando.

Git rebase –abort = Cancela el rebase actual.

Git pull –rebase origin [branch] = Hace un pull pero haciendo rebase antes.

Git commit -m [mensaje] = Se hace commit para salvar los cambios en el repositorio local de Git.

Git commit -am “[mensaje]”= Hace commit a cualquier archivo modificado.

Git checkout – [archivo] = Regresa los cambios hechos en el archivo

Git reset HEAD [archivo] = Sacar un archivo del Stage Area.

Git status = Ver el estado actual del repositorio.

Git clone url [http | ssd] = Clona el repositorio en la carpeta actual.

Git push [repositorio] [Branch] = Manda los cambios a GitHub/Bitbucket.

Git ls-files = Ver que archivos esta trackeando git

Git log = Muestra todos los cambios en el repositorio.

Git log abbrev-commit = Muestra los cambios con el id commit abreviado.

Git log –oneline – graph –decorate = Muestra los cambios con más ordenadamente.

Git log [id\_commit]…[id\_commit] = Te muestra el historial de un rango de commits.

Git log –since =” [fecha]” = Muestra los cambios desde una fecha específica.

Git log – [archivo] = Muestra la historia de un archivo en específico.

Git log –follow – [archivo] = Muestra la historia de un archivo incluyendo las veces que fue renombrada.

Git show [commit id] = Muestra los cambios que se hicieron en un commit.

Git diff(tool) = Muestra las diferencias entre lo que está en el Staging área y el Working Directory.

Git diff(tool) HEAD = Muestra las diferencias entre el Repositorio Local (. git) y el Working directory.

Git diff(tool) –staged HEAD = Muestra las diferencias entre el Repositorio local (. git) y el Staging Area.

Git diff(tool) – [archivo] = Solo muestra los cambios del archivo especificado.

Git diff(tool) [commit 1] [commit 2 || HEAD] = Muestra las diferencias desde un commit especificado hasta otro más nuevo. [Repositorio 1 – Repositorio 2].

Git diff(tool) HEAD^ =

Git diff(tool) [branch\_local] [repositorio/branch] = Compara diferencias entre cambios locales y remotos del repositorio.

ALIAS

Git config –global alias. [nombre\_del\_alias] “[comando\_que\_ejecutara]” = Crea un alias para un comando y salvar tiempo de tipeo.

Git mv [archivo] [nuevo nombre del archivo | carpeta] = Renombra un archivo o lo coloca en una nueva carpeta. Lo coloca en el Staging Area.

Git rm [archivo] = Remueve un archivo que está siendo trackeado por git.