**Comandos empleados en el aprendizaje**

**Brinda la versión de git instalada**

git –version

**Brinda ayuda sobre los comandos**

git help

**Repositorio local**

**Configura el email**

git config --global user.email [miguelguevaraalejandro@gmail.com](mailto:miguelguevaraalejandro@gmail.com)

**Configura el usuario**

git config --global user.name MiguelGuevara

**Lista la configuración del git**

git config –list

**Inicia un nuevo repositorio y crea la carpeta oculta. git**

git init

**Lista el estado de los archivos**

git status

**Agrega todos los archivos pendientes de cambios**

git add –all

o

git add .

**Captura estado del código y lo almacena en el repositorio local. Posterior a git add \***

git commit -m "<descripción>"

**Agregar y guardar cambios en commit**

git commit -am "<descripción>"

**Abre un editor de texto con los cambios del último commit y posibles modificaciones y realizar un commit reemplazando al último.**

git commit –amend

:wq (para salir)

**Modificar nombre de commit:**

git commit --amend -m “ingresa un nombre nuevo”

**Crea un tag**

git tag <nombreTag> -m "Release x.y.z"

**Lista tags**

git tag

**Borra un tag en específico**

git tag -d <nombreTags>

**Hace un tag en un commit anterior**

git tag -a <nombreTag> <código> -m "Release x.y.z"

**Mostrar información del tag**

git show <nombreTag>

**Deshace la captura del estado del código**

git reset

**Muestra los commit realizados hasta el momento**

git log

**Muestra los commit realizados hasta el momento en linea**

git log –oneline

**Lista todos los commits de todas las ramas de forma gráfica tomando como base la rama actual**

git log --oneline --graph –all

**Cambia a un commit en específico por su código con todos sus cambios**

git checkout <código>

**Cambiamos a un commit en específico perdiendo todos los cambios posteriores a este.**

git reset --hard <código>

**Crea una nueva rama**

git branch nombreRama

**Nos muestra en que rama estamos y lista las demás**

git branch

**Nos movemos de la rama actual a una específica**

git checkout nombreRama

**Crea y nos movemos a la nueva rama**

git checkout -b nombreRama

**Renombra la rama actual**

git branch -m nuevoNombre

**Eliminar una rama**

git branch -d nombreRama

**Permite juntar dos ramas. Trae los cambios de la rama específicada a la rama actual**

git merge nombreRama

**Permite juntar dos ramas, pero las mantiene. Genera un commit del merge en la rama actual**

git merge --no-ff develop

**Repositorio remoto**

**Vincular repositorio remoto con repositorio local**

git remote add origin <https://github.com/ElliotXLeo/AprendizajeGit.git>

**Cambiar URL del repositorio remoto**

git remote set-url origin https://github.com/ElliotXLeo/AprendizajeGit.git

**Nos muestra en que repositorio estamos enlazados remotamente.**

git remote –v

**Sube los cambios del repositorio local al remoto y especifica la rama principal**

git push -u origin <nombreRama>

**Sube los cambios del repositorio local al remoto de la rama principal**

git push

**Sube los cambios del repositorio local al remoto de una rama específica**

git push origin <nombreRama>

**Eliminar una rama remota**

git push origin --delete <nombreRama>

**Sube todos los tags locales al remoto**

git push –tags

**Eliminar un tag remoto**

git tag -d <nombreTag>

git push origin :refs/tags/<nombreTag>

**Descarga los cambios del repositorio remoto y actualiza el local en la rama por defecto**

git pull

**Descarga los cambios del repositorio remoto y actualiza el local en una rama específica**

git pull origin <rama>

**Clona un repositorio remoto en la rama por defecto**

git clone https://github.com/ElliotXLeo/AprendizajeGit.git

**Clona un repositorio remoto en una rama específica**

git clone --branch <rama> <https://github.com/ElliotXLeo/AprendizajeGit.git>