**Comandos**

**git config –-global user.name “nombre”**

**git config –-global user.email “correo”**

**Para que tenga registrado quien es el propietario**

**cls**

**Para limpiar la consola.**

**git config –-global -l**

**Para ver quien está registrado como propietario**

**cd \*ruta\***

**Para dirigirte a una parte especifica de la computadora**

**git init**

**Para crear un repositorio.**

**git status**

**Para ver el estado del estado de repositorio, si salen archivos en rojo significa que no hay un control de cambios**

**git add .**

**git add –A**

**git add --all**

**Le da el control a git para gestionar los archivos que no estaban gestionados, los archivos salen en verde.**

**Se encuentran en la zona “stage”**

**git commit –m “descripción de los cambios”**

**Para confirmar los archivos y agregarlos a la rama principal.**

**git diff**

**Para saber la modificación hecha de lo último confirmado**

**git checkout .**

**Se revierte la situación al último commit.**

**git log**

**Enseña el histórico de cambios hechos a lo largo del tiempo**

**git reset \*nombre o ruta del archivo\***

**Para quitar de la parte de stage un elemento en particular o un conjunto de archivos.**

**git commit --amend –m “descripción de los cambios”**

**Para cambiar el nombre del ultimo commit agregado.**

**git mv \*nombre actual del archivo\* \*nuevo nombre del archivo\***

**Para cambiar el nombre de un archivo que nosotros queramos cambiar.**

**git reset --soft \*id del commit\***

**Para hacer un paso atrás en los cambios, ubicándose en el lugar especificado.**

**git reset –hard \*id del commit\***

**Para recuperar archivos en algun punto de todos los commit guardados.**

**git reflog**

**Para mostrar el histórico de todos los commits.**

**git reset –mixed \*id del commit\***

**Para ir a un punto de los itos, evitando los commit que se hicieron desde ese punto en adelante pero con los cambios aun mantenidos.**