**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.**

**GIT IGNORE**

**Sirve para hacer que git ignore ciertos archivos que no quieres que este lleve control.**

1. **Se crea un documento de texto con el nombre “.gitignore” ; este tiene que estar a la misma altura que la carpeta principal de git “.git”.**
2. **Dentro del documento se pone el nombre de los archivos que quieres que sean ignorados, lo guardas y ya**

**git brach \*nombre de la nueva rama\***

**Para crear una nueva rama.**

**git checkout –b \*nombre de la rama\***

**Para crear una rama y que sea dirigido directamente a ella**

**git log - -oneline - -decorate - -all - -graph**

**Se muestra una forma más visual de donde se encuentra el usuario.**

**git checkot \*nombre de la rama\***

**Para cambiar de una rama a otra**

**git merge \*nombre de la rama a la cual se quiere unir\***

**Para unir dos ramas**

**git branch –d \*nombre de la rama\***

**Para eliminar una rama**

**git tag \*nombre de la etiqueta\* \*id de etiqueta\* (si no se pone nombre, será referente a la etiqueta head)**

**Para etiquetar un commit**

**git tag –a \*version\* \*identificador\* -m “mensaje”**

**Se agregar una etiqueta en el identificador**

**git tag –d \*nombre de la etiqueta\***

**Para eliminar una etiqueta**

**git tag –a \*nombre de etiqueta\* -m “detalles de la etiqueta”**

**Para indicar un mensaje adicional que represente que se hizo en esa etiqueta.**

**git show \*nombre de la etiqueta\***

**Para ver todo el detalle que contiene.**

**Github**

**Para crear repositorios**

**🡪Le da click al botón “nuevo repositorio”.**

**🡪Le pone nombre al repositorio.**

**🡪 Le pone descripción (opcional).**

**🡪 Escoge si será pública o privad.**

**🡪Click en “crear repositorio”.**

**Para enlazar un folder seleccionado en el repositorio de github**

**🡪 Se inicializa el folder (Si no está inicializado)**

* **Para enlazar el repositorio con el repositorio loca pon**

**git remote add origin \*enlace(solo si no se puso con anterioridad)\***

**git push –u origin master**

**🡪Una ves puesto bien eso te pedira el usuario y contraseña,lo ponen y si todo esta correcto se enviara todo al repositorio.**

**Para descargar todos los datos de ese repositorio (clonarlo)**

**🡪En donde está el repositorio se selecciona “clonar o descargar” y copian el link que aparece ahí**

**🡪Se dirigen a la carpeta donde quieren clonar los archivos, en la consola.**

**🡪Para descargarlo pon:**

**git clone \*enlace\***

**Para agregar archivos en github**

**🡪Ir a la carpeta donde se quiera crear el nuevo archivo**

**🡪 Dar click en “crear archivo”**

**🡪Se pone un nombre al archivo**

**🡪 Se agrega el contenido de ese archivo**

**🡪Se le agrega el nombre al commit y se agrega una descipción.**

**🡪Se seleccióna si se quiere agregar a la rama master o a una rama distinta.**

**🡪Se da Crick en “Commit new file”**

**Para agregar los archivos adicionados en nuestra rama principal**

**🡪Pon**

**git pull**

**Para ver el historial de un archivo en github**

**🡪Entras al archivo el cual quieres ver el historial**

**🡪Das click en el botón “Historia”**

**🡪Se da click en el identificador del commit que quieras ver**

**🡪 Dar click en “Busqueda de archivos”**

**🡪Se verá el directorio como era en ese instante del tiempo.**

**Para tener una copia de unos repositorios de github en nuestro repositorio**

**🡪Nos dirigimos al repositorio donde se encuentran los archivos**

**🡪Dar click en fork**

**Para solicitar que otro usuario emplee los cambios que nosotros hemos hecho en uno de sus repositorios (pull request)**

**🡪Dar click en “new pull request”**

**🡪Se mostrara una ventana resumen de cuales son los cambios, se da clic en “créate pull request” y llenas los campos.**

**Para aceptar los cambios que un usuario externo haya hecho**

**🡪Dar clic en “Merge pull request” y se confirma.**