***COMANDOS BASICOS***

***DE GIT***

git init // crea el repositorio local

git config user.name "nombreUsuario" // agrega nuestra identidad

git config user.email "emailUsuario" // agrega nuestro e-mail

\*\*(se puede poner git config –global user.name "nombreUsuario" y git config –global user.email "emailUsuario" para logear siempre en esa cuenta lo de esa pc)

git remote add origin http://... // apunta al repositorio remoto

git remote –v // para verificar que estamos apuntando correctamente al repo remoto

git add . // agrega todos los archivos

git commit -m "mensaje" // commitea los cambios hechos

git push origin master // envía los cambios al repositorio remoto

\*\*se puede usar main en lugar de master si no lo toma

git status // realiza un seguimiento de los estados de los archivos

git pull origin master // descarga los cambios que existen en el repositorio remoto

* Cuando quiero descargar un repositorio remoto en mi pc:

git clone *link Del Repo* // Clonamos el repositorio en nuestra máquina

\*\*Tengo que tener invitación del propietario para poder acceder

* Cuando termino de modificar:

git add . // + // git commit -m "mensaje"// + // git push origin master //

\*\*Si alguien que trabaja sobre lo mismo lo modifica y actualiza con push antes que yo, debo primero:

git pull origin master // + // git add . // + // git commit -m "mensaje"// + // git push origin master //

**CONFLICTOS: FLUJO COMPLETO**

⇒ Tenemos listos los cambios que queremos subir al repositorio.

⇒ Hacemos el add, el commit y el push.

⇒ La consola nos devuelve un mensaje de error. No se pudo hacer el push porque en

el repositorio remoto existen cambios que todavía no tenemos en nuestro

repositorio local y no coinciden las versiones.

⇒ Hacemos un pull para actualizar nuestro repositorio local y traer esos cambios.

⇒ Si no hay conflictos entre lo que teníamos y lo que trajimos, Git va a fusionar todos

los cambios de forma automática.

◆ En este caso, la consola nos va a pedir que escribamos un mensaje de

commit para explicar el merge que acaba de suceder. Podemos escribir un

mensaje, o salir de la consola apretando las teclas :q o Ctrl+x .

◆ Solo nos queda hacer el push para mandar nuestros cambios al repositorio

remoto.

⇒ Si no se puede resolver de forma automática, la consola va a devolver un nuevo

mensaje dejándonos saber que existen conflictos que tenemos que resolver a

mano antes de poder subir nuestros cambios al repositorio remoto.

⇒ Vamos al editor de texto para ver cuáles son los archivos conflictuados.

⇒ Git nos deja saber la zona del conflicto de la siguiente manera:

<<<<<<< HEAD

Todo lo que esté entre estas dos líneas serán nuestros cambios locales, lo que estuvimos

trabajando y queremos integrar con el resto del proyecto.

=======

Todo lo que esté entre estas dos líneas serán los cambios que trajimos desde el repositorio

remoto con el pull.

>>>>>>> 3216f3fd5ca65cfd3252ae76808d8f659a715fa6

⇒ Por cada conflicto que encontremos, tendremos que decidir si:

◆ Nos quedamos con nuestro cambio y borramos lo que trajimos.

◆ Nos quedamos con lo que trajimos y borramos nuestro cambio.

◆ Nos quedamos con los dos.

⇒ Resolvemos el conflicto tomando alguno de estos tres caminos y borramos las

líneas de texto que agregó Git: <<<<<<< HEAD, ======= y >>>>>>>

3216f3fd5ca65cfd3252ae76808d8f659a715fa6

⇒ Una vez resueltos todos los conflictos, hacemos el add, el commit y el push para

subir estos nuevos cambios al repositorio remoto.