

Basic commands in GIT

- **GIT CLONE**

es un comando para descargar el código fuente existente desde un repositorio remoto; realiza una copia idéntica de la última versión de un proyecto en el repositorio y la guarda de manera local.

- **GIT BRANCH (Nombre de la rama)**

Trabajar con ramas es la mejor opción, una de las recomendaciones es: nunca hacer cambios en la rama *main* ya que si llegamos a perder el proyecto, se perderá todo el trabajo, la recomendación es crear una rama trabajar en ella y cuando los cambios han sido aprobados a través de un *pull request* procedemos a unir los cambios con la rama *main*.

- **GIT BRANCH**

Comando para ver las ramas existentes

- **GIT CHECKOUT (Nombre de la rama)**

Este comando es nos sirve para cambiarnos de rama

- **GIT INIT**

Este comando es uno de los más importante ya que inicializa la carpeta o ambiente de trabajo, git crea un archivo oculto con el cual git contiene lo necesario para guardar los commit y poder viajar entre las versiones, dicho archivo no se recomienda borrar ya que git no podrá dar seguimiento a los cambios.

- **GIT ADD .**

Este comando nos permite subir los archivos al stash a los cuales se les realizara el commit.

- **GIT COMMIT**

Este comando nos sirve para ir creando las diferentes versiones de un proyecto a las cuales podemos acceder.

- **GIT LOG**

Git log nos sirve para darnos información acerca de los commit que se hayan realizado.

- **GIT STATUS**

Este comando nos sirve para darnos información general del proyecto ejemplo: si hay archivos no añadidos al stash, si hay commit por realizar etc.

- **GIT PUSH**

Este comando es usado para subir los cambios realizados en el proyecto, estos cambios se guardan en un repositorio remoto.

- **GIT PULL**

Comando que nos sirve para obtener las actualizaciones del proyecto remoto, asía nuestro proyecto local

- **GIT REVERT**

Este comando nos ayuda a revertir algún cambio que no queremos en el último commit.

- **GIT MERGE**

Este comando es usado para fusionar las ramas.

- **GIT FECH**

Comando para actualizar los cambios que se realizaron en el servidor (nube) asía el repositorio local

- **GIT PUSH -SET-UPSTREAM ORIGIN "Nombre de la rama"**

Comando para subir una rama al repositorio

- **GIT REMOTE -V**

Comando para ver las direcciones del push y pull

- **GIT PUSH –TAGS**

Comando para subir los tags al repositorio remoto

- **GIT TAG -A (versión) -M “nombre del tag”**

Comando para crear un tag del ultimo commit realizado de esta forma podemos localizar con más facilidad algún commit importante al que se quiera acceder.

- **GIT COMMIT --AMEND -M (mensaje correcto)**

Comando para cambiar el nombre del ultimo commit realizado.

- **GIT CHECKOUT -- .**

Comando para restaurar el proyecto.

- **GIT RESET –SOFT “numero de anterior commit”**

Este comando nos sirve para modificar el ultimo commit realizado

Working with repositories on GitHub (Branches, Merge, Conflicts)

En GitHub también podemos crear ramas para trabajar en ellas y no afectar a la rama principal, de igual forma podemos eliminar una rama cuando ya no se use

Cuando se fusionan las ramas podemos entrar conflictos estos conflictos son porque GitHub no sabe cuál cambio será el correcto para guardar de tal manera que debemos decidir cuál será el cambio que permanecerá de esta manera se da la solución a los conflictos.