

## Comandos locales de git

### *Git init*

Comando inicial, sirve para decir que en esta carpeta vas a usar git, lo inicializas (crea una carpeta .git oculta)

### *Git status*

Comando de información, Permite ver los archivos que tenemos enlazados, o no, con git

Archivos en verde son los que están enlazados con git, los en rojo no

Para diferenciar entre carpetas y archivos mirar si al final tiene una /, lo cual va indicar que es una carpeta, de otro modo es un archivo

### *Git log*

Comando de uso, Para ver todos los commits que hemos creado. Para saber en cuál de todos estamos parados mirar donde dice: (HEAD->)

### *Git diff [archivo]*

Comando de información, sirve para ver las diferencias entre el punto de guardad y el archivo actual

- Con un menos delante y en rojo se muestra lo que se quitó o modifíco
- Con un más delante y en verde lo que se agrego

### *Git add [archivo o ruta]*

Comando de uso, Agrega un archivo al staging área (área de preparación). Se usa para agregar nuevos archivos y cuando se hizo cambios en los mismos.

Se puede usar git add . para añadir todos los archivos modificados

NOTA: Las carpetas que no tengan archivos dentro no las considera.

### *Git commit [-m "comentario"]*

Comando de uso, Para crear un punto de control de nuestro código

(Cuando no usas el parámetro -m) Presionar la tecla i para escribir un comentario, luego escape y :wq

### *Git checkout [-- nom\_archivo][nombre\_branch]*

Comando de uso,

- Si se usa con (nombre\_branch) permite cambiar de branch(rama)
- Si se usa con (-- nom\_archivo) permite revertir los cambios de los archivos al último punto de guardado (cuando usamos el git commit)

### *Git branch [nombre\_nueva\_branch]*

Comando de uso, permite crear nuevas branch(ramas) de código. Por defecto se tiene y usa la master.

### *Ignorar archivos*

Crear un archivo de texto que se llame ".gitignore" y hay listar todos los archivos y directorios a ignorar o no controlar mediante los puntos de guardado. Los archivos tienen que tener su extensión.

El archivo ".gitignore" tiene que añadirse con git add.

## Comandos online de git y github

*Git remote add origin [URL]*

Le dice a git donde vas a alojar tu código en la web

*Git push -u origin [branch]*

le dice a git que suba el código que está en el staging área

*git clone [URL]*

permite descargar el código con su control de versiones de una URL especificada