**Git/GitHub**

El comando git show nos muestra los cambios que han existido sobre un archivo y es muy útil para detectar cuándo se produjeron ciertos cambios, qué se rompió y cómo lo podemos solucionar. Pero podemos ser más detallados.

Si queremos ver la diferencia entre una versión y otra, no necesariamente todos los cambios desde la creación del archivo, podemos usar el comando git diff commitA commitB.

Recuerda que puedes obtener el ID de tus commits con el comando git log.

Comandos para analizar cambios en GIT

* git init: inicializar el repositorio
* git add nombre\_de\_archivo.extensión: agregar el archivo al repositorio
* git commit -m “Mensaje”: Agregamos los cambios para el repositorio
* git add: Agregar los cambios de la carpeta en la que nos encontramos agregar todo
* git status: visualizar cambios
* git log nombre\_de\_archivos.extensión: histórico de cambios con detalles
* git push: envía a otro repositorio remoto lo que estamos haciendo
* git pull: traer repositorio remoto
* ls: listado de carpetas en donde me encuentro. Es decir, como emplear dir en windows.
* pwd: ubicación actual
* mkdir: make directory nueva carpeta
* touch archivo.extensión: crear archivo vacío
* cat archivo.extensión: muestra el contenido del archivo
* history: historial de comandos utilizados durante esa sesión
* rm archivo.extensión: Eliminación de archivo
* comando --help: ayuda sobre el comando
* git checkout: traer cambios realizados
* git rm --cached archivo.extensión: se utiliza para devolver el archivo que se tiene en ram. Cuando escribimos git add, lo devuelve a estado natural mientras está en *staging*.
* git config --list: muestra la lista de configuración de git
* git config --list --show-origin: rutas de acceso a la configuración de git
* git log archivo.extensión: muestra la historia del archivo