



# Tecnológico de Monterrey

## **Actividad Evaluable: Repositorio**

**Herramientas computacionales: el arte de la analítica (Gpo 2)**

**Profesor:**

Javier Mauricio Antelis Ortíz

**Estudiante:**

Samuel Alejandro Díaz del Guante Ochoa - A01637592

**Fecha:**

26/10/2021

## Comandos de terminal

- **ls:** Es uno de los comandos más básicos en los que lista los archivos y carpetas que existen dentro del directorio actual.
- **cd:** Al igual que “ls”, este nos permite desplazarnos entre carpetas. Esto puede ser carpetas que están dentro de la actual para lo cual basta escribir su nombre después de “cd” o puede ser seguido por “..” para salir de la carpeta actual hacia la que lo contiene.
- **pwd:** Son las siglas, Print Working Directory, que imprime a pantalla el camino del directorio en el que se encuentra desde la raíz de la unidad de memoria.
- **clear:** Es una manera muy sencilla de borrar todo el texto que se encuentra en la pantalla de la terminal sin tener que cerrarla o dar muchos espacios. Conserva instrucciones previamente escritas o configuraciones en proceso.
- **rm:** Seguido por el nombre de un archivo dentro de la carpeta, permite remover dicho archivo sin tener que entrar al explorador de archivos y hacerlo manualmente. Existen variantes como **rmdir** para eliminar carpetas de la misma forma.
- **ni:** Similar a “rm” te permite hacer algo que comúnmente tendrías que hacer desde el explorador de archivos, en este caso siendo crear un nuevo ítem como archivos de texto. También existe **mkdir** para crear nuevas carpetas en el directorio actual.

## Comandos de git

- **git:** Todos los comandos de git deben de preceder la palabra “git”. Si se escribe por sí sola sirve como una clase de “-help” que muestra las acciones que puede ejecutar el programa acompañados de una pequeña instrucción.
- **clone:** Es una de las primeras cosas que se deben hacer para empezar a trabajar con Git y GitHub. En este caso te permite adquirir una copia de los archivos que se encuentran en ese momento de un repositorio remoto. Para ello podemos crear un repositorio desde la aplicación de GitHub, copiar su url y situarnos en la carpeta en la que queremos clonar el repositorio de forma local en nuestra máquina.
- **add:** Antes de poder realizar cualquier cambio localmente hacia el repositorio remoto se tienen que hacer un par de cosas. Primero utilizar la línea “git add .” con esto, git puede escanear la carpeta local e identificar las partes que han sido modificadas y que están listas para ser puestas en el repositorio remoto.
- **commit:** Ya casi podemos ver reflejados nuestros cambios en el repositorio remoto pero primero se tiene que hacer un “commit” que va acompañado de una pequeña descripción en donde podemos describir en un par de palabras lo que hicimos en esa ocasión. Estos deben ser claros ya que nos pueden ayudar a identificar con mayor facilidad las pasadas versiones en caso de que se tenga que restaurar alguna de ellas.
- **push:** Es el paso final para publicar nuestros cambios a la rama en que nos encontramos, generalmente siendo el “main”. Es importante que no existan conflictos entre tu versión local y la que se encuentra en GitHub porque de otra forma tendrá que ser resueltos de manera manual y no siempre termina funcionando como esperamos.
- **pull:** Por ello es importante siempre hacer “pull” antes de trabajar de forma local y verificar que nuestra versión esté al par de la remota. En pocas palabras esta función es como un “update” de la versión remota hacia nuestras computadoras.