

# Comandos de la terminal Bash

## Mostrar

<b>\$ ls</b>	Lista archivos en el directorio o carpeta
<b>\$ ls -a</b>	Lista todos los archivos, incluyendo los archivos ocultos
<b>\$ ls -l</b>	Muestra toda la información de una carpeta: usuario, grupo, permisos, tamaño, fecha y hora de creación.
<b>\$ ls -R</b>	Muestra las carpetas y los archivos contenidos en ellos de manera recursiva
<b>\$ pwd</b>	Muestra la carpeta en la que se está trabajando actualmente
<b>more [Nombre del archivo]</b>	Muestra el contenido de un archivo

## Crear

<b>\$ mkdir [Carpeta]</b>	Crea una nueva directorio o carpeta
<b>\$ touch [Nombre del archivo]</b>	Crea un nuevo archivo

## Eliminar

<b>\$ rm [Nombre del archivo]</b>	Elimina un archivo
<b>\$ rmdir [Nombre de la carpeta]</b>	Elimina una carpeta vacía
<b>\$ rm -r [Nombre de la carpeta]</b>	Elimina una carpeta y su contenido

## Navegación entre carpetas

<b>\$ cd ..</b>	Sube un nivel de carpeta
<b>\$ cd</b>	Cambia de carpeta
<b>\$ cd/chosen/directory</b>	Cambia a una carpeta específica

## Otros comandos

<b>\$ clear</b>	Limpia la pantalla de la terminal
-----------------	-----------------------------------

Copiar/Mover/Renombrar	
<b>\$ mv</b> <b>[ruta/archivo1]</b> <b>[ruta/archivo2]</b>	Renombra archivos (archivo2 no debe existir o será sobrescrito)
<b>\$ mv</b> <b>[ruta/carpeta1]</b> <b>[ruta/carpeta2]</b>	Renombra la carpeta1 como carpeta2 (carpeta 2 no debe existir)
<b>\$ mv</b> <b>[ruta/carpeta1]</b> <b>[ruta/carpeta2]</b>	Mueve contenido de carpeta1 a carpeta2 (carpeta 2 debe existir)
<b>\$ cp</b> <b>[ruta/archivo1]</b> <b>[ruta/archivo2]</b>  <b>\$ cp</b> <b>[ruta/archivo1]</b> <b>[ruta/archivo2]</b>	Copia un archivo o carpeta
<b>opción: -r</b>	Indica que copie recursivamente el contenido de las subcarpetas

### Git branch

Para crear una nueva rama utilizamos el comando git branch. Git branch nos permite crear,

enumerar, cambiar el nombre y eliminar ramas. No permite cambiar entre ramas o volver a

unir un historial bifurcado.

#### ● git branch

Enumera todas las ramas de tu repositorio, es similar a git branch --list.

#### ● git branch <branch>

Crea una nueva rama llamada <branch>.

```
git checkout -b <nombre-de-tu-rama>
```

Este comando crea una nueva rama en local (-b viene de rama (branch)) y te cambia a la rama que acabas de crear.

#### ● git branch -d <branch>

Elimina la rama llamada <branch>. Git evita que eliminemos la rama si tiene cambios

que aún no se han fusionado con la rama Main.

#### ● git branch -D <branch>

Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.

3

### Git checkout

Para moverse de una rama a otra, se ejecuta el comando

#### ● git checkout nombre\_rama

Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si

tenemos cambios, para cambiarnos de rama, debemos:

1. Eliminarlos (deshaciendo los cambios).
2. Confirmarlos (haciendo un git commit).

Guardar cambios y subirlos al repositorio remoto

Una vez que terminamos de realizar los cambios que queremos en nuestra branch,

ejecutamos los mismos comandos que vimos hasta ahora: git add, git commit, git status y git

log. Pero cuando queramos subir esos cambios, debemos utilizar git push con el nombre de la

rama en que estamos posicionados:

#### ● git push origin <branch>

Así también, para traer los cambios de esa rama utilizamos el git pull agregando desde donde

queremos traer los cambios:

#### ● git pull origin <branc

