[SEBASTIÁN KRAGLIEVICH](mailto:skraglievich@fi.uba.ar)

**Observaciones:**

“carpeta” = “directorio”

(así se llaman (así se llaman en Mac

en windows) o Linux)

**CTRL + L** → Comando para limpiar la terminal

Comandos básicos de GIT Bash

Comandos:

* **ls** **→** Permite listar todos los archivos y carpetas que se encuentran en un determinado directorio.
* **pwd** **→** Sirve para saber en qué carpeta/directorio nos encontramos actualmente (en el momento de la ejecución del comando).
* **cd <nombre de carpeta>** **→** Sirve para movernos entre nuestras distintas carpetas o directorios.
* **cd ..** **→** Sale del directorio actual, devolviendonos al directorio anterior (en la jerarquía).
* **mkdir** **→** Crea un directorio nuevo dentro del directorio en el que nos encontramos.
* **git init** **→** Sirve para inicializar un repositorio vacío dentro del directorio en el que nos encontremos (inicializa un proyecto que gestionaremos con git).
* **ls -a** **→** Para mostrar todos los archivos y directorios que se encuentran en el directorio actual (muestra TODO, hasta los archivos y directorios ocultos).
* **code . →** Abre el editor de texto. El “**.**” sirve para abrir la carpeta en la que nos encontramos (se abre en VS Code, que es el editor de texto que configuré por defecto.)
* **git status** **→** Muestra el estado actual de nuestro repositorio de GIT.
* **git add <nombre archivo> →** Selecciona el archivo mencionado y lo agrega a la estapa de “**Stage**”.
* **git add “<extensión>”** **→** Agrega todos los archivos que tengan dicha extensión a la etapa de “**Stage**”. Puede ser cualquier extensión. Por ej: git add “.txt”
* **git add . →** Agrega a la etapa “**Stage**” TODOS los archivos que aparecen en rojo enlistados al momento de ejecutar el comando “**git status**”. (Es una **mala práctica**, hay que usarlo con cuidado).
* **git commit -m “<mensaje>” →** Sirve para pasar los archivos de la etapa “**Stage**” a la etapa “**Commit**”. Es decir, sirve para comprometer los archivos.
* **rm <nombre de archivo> →** Sirve para eliminar el archivo indicado del repositorio GIT.
* **git rm <nombre de archivo> →** Sirve para eliminar un archivo del repositorio GIT agregándolo directamente a la etapa “**Stage**”. (Con este comando nos ahorramos un paso comparado con el comando “**rm <nombre de archivo>**”.
* **git restore <nombre de archivo> →** Recupera un archivo eliminado.
* **mv <archivo><destino> →** Sirve para renombrar un archivo.
* **git restore --stage <archivo> →** Saca un cambio agregado en la etapa “**Stage**”.
* **git mv <archivo><nuevo nombre> →** Sirve para cambiar el nombre del archivo, y agrega ese cambio inmediatamente a la etapa “**Stage**”.
* **git status -s →**  Muestra el estado actual del repositorio GIT de forma abreviada/resumida.
* **git diff →**  Muestra todos los cambios realizados sobres los archivos que todavía no hemos pasado/agregado a la etapa de “**Stage**”.
* **git diff --staged →** Muestra todos los cambios de los archivos que se encuentran en la etapa “**Stage**”. Este comando es muy útil cuando se desean ver sólo los cambios que se desean comprometer.
* **git log →** Muestra el historial de todo el repositorio (serían todos los commit realizados, junto con su código de identificación). A diferencia de “**git log --oneline**” muestra además, los datos de la persona que realizó el commit. Estos son el nombre, el email, y la fecha en que se realizó dicho commit.
* **git log --oneline →** Muestra el historial de todo el repositorio de forma abreviada.
* **git branch →** Muestra la rama en la cual estamos trabajando.
* **git checkout -b <nombre de la rama> →** Sirve para crear una rama nueva con el nombre indicado.
* **cat <nombre de archivo> →** Muestra el contenido del archivo.
* **git checkout <rama> →** Sirve para cambiar la rama en la que estamos.
* **git merge <rama> →** Sirve para traer los cambios (o los archivos mejor dicho) de la rama indicada hacia la rama en la que estamos actualmente.

**Trabajar en distintas ramas permite que c/programador desarrolle una funcionalidad por su parte y luego las mezclen (merge) con la rama principal (main).**