

Comandos Git Bash			
pwd: nos muestra la carpeta actual en la que nos encontramos.	mkdir: nos permite crear carpetas, p. ej. mkdir NuevaCarpeta.	touch: nos permite crear archivos nuevos, p.ej. touch NuevoArchivo.txt	cat: nos permite ver el contenido de un archivo, p.ej. cat NuevoArchivo.txt
cd: nos permite cambiarnos de carpeta, p.ej. cd NuevaCarpeta.	cd .. : nos permite regresar al directorio o carpeta anterior.	ls: nos permite ver los archivos de la carpeta donde estamos actualmente.	ls -l: Ver todos los archivos como una lista en donde incluye el usuario, grupo, permisos sobre el archivo, tamaño, fecha y hora de creación.
ls-R: muestra el contenido de todos los subdirectorios de forma recursiva.	ls -S Ordena los resultados por tamaño de archivo.	rm: Nos permite borrar un archivo o carpeta ej: rm NuevoArchivo.txt	rmdir "nombre del directorio"/: borrar un directorio: Solo funciona con directorios vacíos.
rm -r 'nombre de la carpeta' :me permite eliminar la carpeta y los archivos dentro de ella de forma recursiva.	cp "nombre del archivo que quremos copiar" "nombre del directorio a donde lo queremos copiar": nos permite copiar un archivo.	mv "el directorio de donde queremos mover/el nombre del archivo" "el directorio hacia donde lo queremos mover": nos permite mover un archivo.	clear: nos permite limpiar la pantalla.
history: ver los últimos comandos que ejecutamos y un número especial con el que podemos volver a repetir el comando.			

Comandos Git			
git clone: realiza una copia idéntica de la última versión de un proyecto en un repositorio y la guarda en tu ordenador.	git branch: usamos el comando git branch para crear ramas, listarlas y eliminarlas.	git checkout: lo usamos para cambiarnos de rama.	git status: nos da toda la información necesaria sobre la rama actual.
git add: cuando creamos, modificamos o eliminamos un archivo, estos cambios suceden en local y no se incluirán en el siguiente commit (a menos que cambiemos la configuración). Necesitamos usar el comando git add para incluir los cambios del o de los archivos en tu siguiente commit.	git commit: una vez que se llega a cierto punto en el desarrollo, queremos guardar nuestros cambios (quizás después de una tarea o asunto específico). Git commit es como establecer un punto de control en el proceso de desarrollo al cual puedes volver más tarde si es necesario. También necesitamos escribir un mensaje corto para explicar qué hemos desarrollado o modificado en el código fuente.	git push: envía tus commits al repositorio remoto.	git pull: se utiliza para recibir actualizaciones del repositorio remoto.