

Comandos principales Git

ls	Lista los archivos contenidos en el directorio.
ls -a	Lista los archivos contenidos en el directorio, mostrando también los ocultos.
ls -t	Muestra los archivos ordenados según el ultimo modificado.
ls -h	Muestra mejor el tamaño de los archivos.
rm -rf	Para remover los archivos dentro de un directorio y subdirectorios, la f es de forzar para que no pregunte confirmación por cada archivo o carpeta.
cat	Muestra el contenido del archivo
tail -f	Muestra el archivo y su contenido en tiempo real, lo vigila.. si se modifica el archivo te muestra tambien. opción -n 5 por ejemplo, muestra las últimas 5 líneas del archivo.
nano	Es para editar el archivo.
find	-iname es la opción para buscar por nombre sin distinguir mayúsculas, sino sin la i
	-size por tamaño 25c (25 bytes). k kilobytes M megabytes G Gigabytes.
	Según modificación. -time (24hs) -min (minutos) -m modificado -a accedido -c modificado
	-find -mtime 0 (entre ahora y hace 1 día) find -mtime -1 menos de un día find -mtime 1 entre 24 y 48 hs find -mtime +1 más de 48 hs.
	-type según tipo de archivo, hay varios.
grep	Busca texto dentro de los archivos.
	Se usa como grep (texto a buscar) (archivo donde buscar, si se pone asterisco busca en todos los archivos del directorio).
	-l no muestra el detalle de la cantidad de veces que se repite, sino solo muestra el nombre del fichero que lo contiene, por mas que lo tenga mas de una vez.
	-r buscar recursivamente en carpetas
git --version	Para ver la versión de git instalada.
git init	Crea un repositorio local (en mi compu). Distinto si fuera remoto, que se hospeda en GitHub.
git config user.name "dmartin07"	Para configurar usuario en el repositorio.
git config --global user.name "dmartin07"	Lo configura para todos los repositorios de la pc.
git config user.name	Para ver el nombre de usuario.
\$ git config user.email "dariomartinbenavidez@hotmail.com"	Para indicarle el mail que usamos en github, mismo comando sin mail para que nos devuelva el mail indicado anteriormente.
git config --global user.email	Para todos los repositorios.
git add nombre archivo	Para agregar a git archivo específico.
git add .	Para agregar todos los archivos de la carpeta.
git rm --cached nombrearchivo	Para sacar archivo del repositorio.
git status	Muestra los archivos que estan agregados al repositorio y los que no.
git commit -m "nombre de la versión o instancia"	Para que guarde un commit. Solo guardará aquellos archivos a los que se les haya hecho add anteriormente.
Para volver a un x commit y el ultimo dejarlo para revisión.	Copiar el identificador del commit (parece una wallet, tambien puede ser los primeros 7 digitos de ese commit) desde log, y usar el comando: git checkout identificador del commit y lo que hace es una rama con ese commit, dejando el main con el commit anterior.
git log	Muestra todos los commit generados, con nombre de usuario con fecha y hora.
git remote add origin + el link del repositorio remoto	Para vincular el repositorio remoto con el local.
git remote -v	Para verificar vinculación.

git push origin main	Actualizar repositorio remoto, previamente tiene que estar commiteado.
git push -u origin main	Hace que se correspondan las ramas locales con las de github. sirve para la primera vez para subir todo lo local a github.
git clone urldelrepositorio	Es para clonar los archivos en una nueva pc, por primera vez. Luego para descargar actualizaciones del mismo se usa otro comando.
git pull origin main	Es para actualizar el repositorio local.
git rm archivo	Para remover archivo.
git rm -r directorio	Remueve el directorio y los archivos que contiene.
git log	Muestra el historial.
git log --	Muestra el historial de un archivo específico.
git log --author=usuario	Muestra el historial de un usuario en particular.
git log --oneline	Solo se ven los primeros digitos del commit.
git branch	Enumera las ramas.
git branch rama	Crea una nueva rama llamada rama. Esta rama ya tiene los archivos que estan en main, no duplicados, pero estan ahi, los tiene como referencia.
git branch -m nuevonombre	Para cambiar nombre de la rama.
git branch -d rama	Elimina la rama llamada rama. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.
git branch -D rama	Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.
git checkout nombre_rama	Para moverse de una rama a otra.
git checkout -b nombre_rama	Crea una rama y cambio a esa rama.
git merge rama_origen rama_main(destino)	Para fusionar (parado en la rama destino):
git fetch	Descarga lo que esta en remoto.

Dario Martín Benavidez