

Chuleta: Comandos Unix y subcomandos de Git

Comandos Unix

```
$ pwd
Muestra la ruta completa del directorio de trabajo actual.

$ cd <directory name>
Cambia el directorio de trabajo a <directory name>.

$ cd ..
Navega hacia el directorio padre.

$ ls
Lista el contenido del directorio actual.

$ ls -al
Muestra una vista detallada del directorio, incluyendo archivos ocultos.

$ mkdir <new directory name>
Crea un nuevo directorio llamado <new directory name>.
```

Crear repositorios

```
$ git init [repository name]
Crea un repositorio local en una carpeta con el nombre dado. Sin el nombre de la carpeta lo creo en el directorio actual.

$ git clone [repository url]
Descarga un repositorio con todas sus ramas e historial de versiones.
```

Hacer cambios locales

```
$ git status
Lista todos los archivos nuevos o modificados.

$ git diff
Muestra en que han cambiado los archivos no añadidos.

$ git add [file name]
Añade los archivos para que sean commiteados.

$ git add -A
Añade todos los archivos con cambios para que sean commiteados.

$ git reset [file name]
Desañade los archivos, pero preserva los cambios.

$ git commit -m "[message]"
Crea el commit (nueva versión) con los archivos añadidos, con el mensaje dado.
```

Trabajar con ramas

```
$ git branch -av
Lista todas las ramas (locales y remotas) de forma verbosa.

$ git branch [new branch name]
Crea una nueva rama a partir de la rama actual.

$ git checkout [branch name]
Cambia a la rama especificada, cambiando el contenido del directorio actual.

$ git branch -d [branch name]
Elimina la rama especificada.

$ git tag [tag name]
Marca el commit actual (de la rama actual) con una etiqueta.

$ git merge [branch name]
Combina la rama especificada en la actual, creando un nuevo commit.
```

Sincronizar con remotos

```
$ git push [remote] [branch]
Sube al repositorio remoto la rama indicada. Se pueden eliminar estas opciones.

$ git pull [remote] [branch]
Descarga del repositorio. No es elegante.

$ git fetch
Descarga del repositorio remoto, pero no lo añade al directorio de trabajo.

$ git rebase
Añade lo del descargado al directorio de trabajo.

$ git pull --rebase
git fetch y git rebase en un solo paso.
```

Deshacer cambios

```
$ git reset [commit id]
Vuelve al commit anterior y preserva todos los cambios como cambios sin commitear o añadir.

$ git reset --hard [commit id]
Vuelve al commit anterior pero elimina todos los cambios.

$ git reset --hard HEAD
Descarta todos los cambios en el directorio de trabajo.
```

Otros subcomandos

```
$ git mergetool
Lanza la herramienta para solucionar conflictos, si los hay.

$ git log
Lista el historial de versionado para la rama actual.

$ git stash
Guarda en una pila interna los cambios no añadidos. Es recomendable que el tamaño de la pila nunca sea mayor que 1.

$ git stash pop
Saca de la pila los últimos cambios stasheados y los añade al directorio de trabajo.

$ git stash drop
Saca de la pila los últimos cambios stasheados y los elimina.
```