
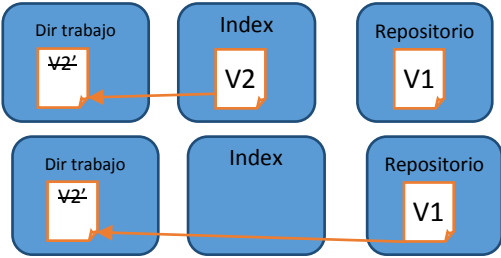


# GIT

Comandos iniciales	
<pre>git config [ámbito] [acción] [parametro1_accion] [parametro1_accion] [parametro3_accion]</pre>	<p>Ámbito:</p> <ul style="list-style-type: none"> <li>--system</li> <li>--global</li> <li>Si no ponemos nada, por defecto la de usuario</li> </ul> <p>Acción:</p> <ul style="list-style-type: none"> <li>--list: listar toda la configuración</li> <li>--get: obtener un valor de una variable.</li> <li>--add: añadir a una variable un valor.</li> <li>--unset: elimina la variable</li> </ul>
git --version	Devuelve la versión
git help <comando>	Para obtener ayuda
git <comando> --help	
Comandos básicos	
git init	Crea un repositorio
git add	Añade un archivo al área de preparación
git add -	Añade TODOS los archivos modificados al área de preparación
git commit -m "comentario"	Crea un commit con los archivos del área de preparación y agrega un comentario al commit
git tag -a nombreetiqueta -m "comentario" identificadorCommit	Crea una etiqueta asociada a un commit y le añade un comentario
git log	Muestra un histórico de commits almacenados
git log -graph	Lo muestra de manera gráfica (útil cuando trabajamos con ramas)
git status	Nos muestra el estado de las tres secciones
git cat-file -p identificadorobjeto	Muestra información sobre un objeto
git cat-file -p HEAD	Muestra información sobre el último commit
git cat-file -p HEAD:	Muestra el contenido del tree del último commit
git cat-file -p HEAD:nombrefichero	Muestra el contenido del archivo en el último commit
Comandos complementarios	
Comprobar diferencias	
git dif	nos proporciona las diferencias entre la copia del directorio de trabajo y la existente en el área de preparación
git dif --staged	nos proporciona las diferencias entre la copia existente en el área de preparación y el último <i>commit</i> .

# GIT

Deshacer cambios	
<b>git reset HEAD nombreArchivo</b> 	Quita del área de preparación el archivo SIN MODIFICAR el que está en el área de trabajo
<b>git checkout nombreArchivo</b> 	sustituye el fichero nombreArchivo que hay en el directorio de trabajo por el que hay en el área de preparación. o, en su defecto, en el repositorio.(si no está en el área de preparación)
<b>git commit -a -m "comentario"</b>	Con la opción -a se salta el área de preparación
<b>git commit --amend</b>	añade al último <i>commit</i> los ficheros que existen en el área de intercambio. Además con la opción -m podemos modificar el mensaje que aparecía en ese commit
Revisión de código anterior	
<b>git checkout idcommit</b> <b>git checkout HEAD~N</b>	Vuelve al commit especificado En este caso en lugar de identificar el commit con su código, utilizamos HEAD~N donde N es el número de commits anteriores a HEAD que quiero recuperar
<p>MUCHO CUIDADO AL VOLVER A UN COMMIT ANTERIOR</p> <p>En el momento en que se recupera un <i>commit</i> anterior Git crea una de rama virtual donde el usuario puede ir trabajando y haciendo <i>commits</i>. Pero esas modificaciones no afectan al árbol principal de trabajo. Es más, en cuanto volvamos a uno de los <i>commits</i> del grafo esas modificaciones se perderán. Para no hacerlo, siempre es posible crear una rama desde ese punto y trabajar sobre ella.</p> <p><i>Consultar el final del resumen sobre trabajo con RAMAS</i></p>	
RAMAS	
<b>git branch</b>	Muestra las ramas que hay y marca con un * la actual
<b>git branch nombreRama</b>	Crea una nueva rama con el nombre proporcionado, pero no cambiamos de rama, seguimos en la original
<b>git checkout nombreRama</b>	Cambiamos a la rama indicada
<b>git log --graph --decorate --pretty=oneline --all</b>	Imagen gráfica del grafo d commits y ramas
<b>git branch -d nombreRama</b>	Elimina la rama indicada. No es posible hacerlo si nos encontramos en ella
Fusión de ramas	
<b>git checkout ramaDestino</b> <b>git merge ramaAFusionar</b>	1. Nos situamos en la rama en la que queremos incorporar los cambios 2. Fusionamos
<b>git checkout ramaAMover</b> <b>git rebase ramaoriginal</b>	1. Nos situamos en la rama a reestructurar 2. Hacemos un rebase sobre la rama original En este momento la rama a fusionar es descendiente de la rama original y podemos ejecutar un fast-forward