

Comando	Descrición
<b>git config --global user.name "Nome Apelido"</b>	Modifica o nome e apelidos do usuario que vai crear versión nese proxecto.
<b>git config --global user.email "omeu@correo.gal"</b>	Modifica o correo electrónico do usuario que vai crear versión nese proxecto.
<b>git init</b>	Xera no cartafol os arquivos necesarios para o control de versións usando Git.
<b>git add &lt;arquivo/directorio/regexp&gt;</b>	Engade á área de preparación as arquivos ou directorios que se lle indica. Para engadir todo o directorio: <code>git add .</code>
<b>git status</b>	Amosa o estado dos arquivos do directorio (untracked, staged, committed...)
<b>git commit</b>	Xera un commit cos arquivos presentes na área de preparación. Abre o editor por defecto para escribir a mensaxe explicativa.
<b>git commit -m "Mensaxe explicativa"</b>	Xera un commit cos arquivos presentes na área de preparación e asígnalle a mensaxe explicativa que se lle indica.
<b>git commit -a -m "Mensaxe explicativa"</b>	Xera un commit con todos os arquivos que se modificaron, aínda que non se pasaran á área de preparación (non os arquivos novos aos que nunca se lles fixo un <code>git add</code> ).
<b>git log</b> <b>git log --oneline</b>	Amosa as distintas versións do proxecto coa súa información (a versión oneline é resumida nunha liña cada commit).
<b>git diff</b>	Amosa na propia consola as diferenzas entre o espazo de traballo e o repositorio local.
<b>git diff &lt;id&gt;</b>	Amosa na propia consola as diferenzas entre o espazo de traballo e o commit de id indicado.
<b>git diff &lt;nome_arquivo&gt;</b>	Amosa na propia consola as diferenzas entre o espazo de traballo e o repositorio local no arquivo indicado.
<b>git diff &lt;id&gt; &lt;nome_arquivo&gt;</b>	Amosa na propia consola as diferenzas entre o espazo de traballo e o commit de id indicado no arquivo indicado.
<b>git restore &lt;arquivo&gt;</b>	Desfacer os cambios e voltar ao que hai no último commit. (EXPERIMENTAL)
<b>git checkout &lt;arquivo&gt;</b>	Desfacer os cambios e voltar ao que hai no último commit.
<b>git log --graph</b>	Amosa os distintas versións no repositorio ( <i>commits</i> ) en forma de grafo, no que se poden ver adecuadamente as distintas ramas.
<b>git tag -a &lt;etiqueta&gt; -m &lt;mensaxe&gt;</b>	Etiqueta o último commit.
<b>git tag -a &lt;etiqueta&gt; -m &lt;mensaxe&gt; &lt;id_commit&gt;</b>	Etiqueta o commit con id <code>id_commit</code> .
<b>Xestión das ramas</b>	
<b>git branch &lt;nova_rama&gt;</b>	Crea unha nova rama a partir da actual.
<b>git branch &lt;nova_rama&gt; &lt;RAMA_EXISTENTE&gt;</b>	Crea unha nova rama a partir de RAMA_EXISTENTE.
<b>git checkout &lt;RAMA&gt;</b>	Cambiamos á rama RAMA (que xa debe estar creada).
<b>git switch &lt;RAMA&gt;</b>	Cambiamos á rama RAMA (que xa debe estar creada). (EXPERIMENTAL)
<b>git checkout -b &lt;nova_rama&gt;</b>	Crea unha nova rama a partir da actual e cámbiase a ela.
<b>git switch -c &lt;nova_rama&gt;</b>	Crea unha nova rama a partir da actual e cámbiase a ela. (EXPERIMENTAL)
<b>git merge &lt;OUTRA_RAMA&gt;</b>	Fusiona a rama na que nos atopemos con OUTRA_RAMA. Pode haber que resolver conflitos manualmente, e despois: <code>git add .</code> <code>git commit -m "Nova mensaxe"</code>
<b>git branch -d &lt;rama&gt;</b>	Borra unha rama.
<b>git log --all</b>	Amosa as distintas versións do proxecto en todas as ramas.
<b>git log --graph</b>	Amosa as versións do proxecto como un grafo.
<b>git log --all --oneline --graph</b>	Amosa as distintas versións de forma curta (unha soa liña) do proxecto en todas as ramas como un grafo.

Revertindo cambios	
<b>git reset --hard &lt;id_commit&gt;</b>	Volve a un commit anterior borrando todo o rastro dos posteriores.
<b>git revert &lt;id_commit&gt;</b>	Utilízase para volver a un commit anterior pero mantendo os commits intermedios. O <id_commit> refírese ao do último commit que revirte (voltará ao commit anterior a este). Trata de facer un merge do commit ao que queremos voltar e o último da rama, polo que adoita haber conflitos e haberá que resolvelos manualmente (normalmente aceptando os cambios entrantes). No caso de conflito, unha vez resolto: git add . git revert --continue (ou simplemente facer commit)
<b>git checkout &lt;id_commit&gt;</b>	Volve provisionalmente a un commit anterior, para botar un vistazo. A cabeza queda en estado desacoplado. Voltamos con <b>git checkout master</b> .
<b>git checkout &lt;arquivo(s)&gt;</b>	Volve o arquivo ou arquivos á situación na que estaban no último commit, eliminado os cambios feitos ata entón.
<b>git reset &lt;arquivo(s)&gt;</b>	Sacar o ficheiro ou ficheiros da zona de preparación e voltar ao directorio de traballo.
Incorporar ficheiros a commit	
<b>git commit --amend</b>	Engade o que teñamos na área de preparación ao último commit ou o modifica, de ser o caso.
<b>git stash</b>	Garda unha instantánea do que estamos facendo, sen facer commit e revirte a situación ao último commit.
<b>git stash list</b>	Lista as instantáneas creadas con stash.
<b>git stash apply &lt;id_instantánea&gt;</b>	Volve á instantánea indicada. Pode haber que resolver os conflitos, procedendo coma no merge.
<b>git stash drop &lt;id_instantánea&gt;</b>	Elimina a instantánea indicada.
Operacións con repositorios remotos	
<b>git remote add origin &lt;url&gt;</b>	Engade un repositorio remoto chamado origin coa URL indicada (ollo co copia/pega, usa o botón dereito para que non engada caracteres non visibles que provocan erros).
<b>git remote remove &lt;nome_repo&gt;</b>	Elimina este enlace ao repositorio en local.
<b>git push origin master</b>	Sube a rama master ao repositorio remoto origin.
<b>git push -u origin master</b>	Selecciona por defecto a subida da rama master ao repositorio origin cada vez que se faga git push. (pódese substituír -u por --set-upstream)
<b>git pull origin master</b>	Baixa a master o existente no repositorio remoto e o mestura coa rama master. Poden existir conflitos que se resolven coma no caso do merge.
<b>git fetch</b>	Comproba os cambios feitos no repositorio ou rama pero non descarga arquivos.
<b>git clone &lt;url&gt;</b>	Inicializa o repositorio e descarga un repositorio remoto. Realiza git init, git remote add e git pull dunha soa vez.