Comando	Descrición
Crear repositorio e indicar usuario y correo electrónico	
git init	Xera no cartafol os arquivos necesarios para o control de versións usando Git.
git config --global user.name "Nome Apelido"	Modifica o nome e apelidos do usuario que vai crear versión nese proxecto.
git config --global user.email "omeu@correo.gal"	Modifica o correo electrónico do usuario que vai crear versión nese proxecto.
Cambio de estado nos arquivos	
git add <arquivo/directorio/regexp>	Engade á área de preparación as arquivos ou directorios que se lle indica. Para engadir todo o directorio: git add .
git status	Amosa o estado dos arquivos do directorio (untracked, staged, commited...)
git commit	Xera un commit cos arquivos presentes na área de preparación. Abre o editor por defecto para escribir a mensaxe explicativa.
git commit -m "Mensaxe explicativa"	Xera un commit cos arquivos presentes na área de preparación e asígnalle a mensaxe explicativa que se lle indica.
git commit -a -m "Mensaxe explicativa"	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 git add).
Etiquetar commits	
git tag -a <etiqueta> -m <mensaxe>	Etiqueta o último commit.
git tag -a <etiqueta> -m <mensaxe> <id_commit>	Etiqueta o commit con id id_commit.
Ver o historial de cambios	
git log **git log –oneline**	Amosa as distintas versións do proxecto coa súa información (a versión oneline é resumida nunha liña cada commit).
git log --graph	Amosa os distintas versións no repositorio (*commits*) en forma de grafo, no que se poden ver adecuadamente as distintas ramas.
Comparar (Buscar diferenzas)	
git diff	Amosa na propia consola as diferenzas entre o espazo de traballo e o repositorio local.
git diff –staged or --cached	Amosa as diferenzas entre o espazo staged e o repositorio local.
git difftool	Utiliza unha ferramenta para comparar (tkdiff, vimdiff)
git diff <id>	Amosa na propia consola as diferenzas entre o espazo de traballo e o commit de id indicado.
git diff <nome_arquivo>	Amosa na propia consola as diferenzas entre o espazo de traballo e o repositorio local no arquivo indicado.
git diff <id> <nome_arquivo>	Amosa na propia consola as diferenzas entre o espazo de traballo e o commit de id indicado no arquivo indicado.
Xestión das ramas	
Creación de ramas e cambio dunha rama a outra	
git branch <nova_rama>	Crea unha nova rama a partir da actual.
git branch <nova_rama> <RAMA_EXISTENTE>	Crea unha nova rama a partir de RAMA_EXISTENTE.
git checkout <RAMA>	Cambiamos á rama RAMA (que xa debe estar creada).
git checkout -b <nova_rama>	Crea unha nova rama a partir da actual e cámbiase a ela.
Fusión de ramas e comparar ramas	
git merge <OUTRA_RAMA>	Fusiona a rama na que nos atopemos con OUTRA_RAMA Pode haber que resolver conflitos manualmente, e despois: (1) git add . (2) git commit -m "Nova mensaxe"
git merge --abort	Desfacer o último merge
git diff <RAMA1> <RAMA2>	Ver as diferenzas entre ramas
Eliminar rama	
git branch -d <rama>	Borra unha rama.

Amosar versións do proxecto	
git log –all **gil log –graph** **git log –all –oneline --graph**	Amosa as distintas versións do proxecto en todas as ramas. Amosa as versións do proxecto como un grafo. Amosa as distintas versións de forma curta (unha soa liña) do proxecto en todas as ramas como un grafo.
Renombrar y borrar archivos	
git mv <orixinal> <novo>	Para renomear arquivos en git. Despois debemos facer commit para confirmar os cambios no HEAD.
git rm <archivo(s)> **git rm –cached <archivo(s)>**	Para eliminar arquivos do working directory y de stage. Para eliminar arquivos só de stage.
Revertindo cambios	
git checkout <arquivo(s)>	Volve o arquivo ou arquivos á situación na que estaban no último commit, eliminado os cambios feitos ata entón.
git checkout <id_commit>	Volve provisionalmente a un commit anterior, para botar un vistazo. A cabeza queda en estado desacoplado. Voltamos con **git checkout master**.
git checkout HEAD <archivo>	Recupera o arquivo na súa última versión.
git reset <arquivo(s)>	Sacar o ficheiro ou ficheiros da zona de preparación e voltar ao directorio de traballo.
git reset –hard <id_commit>	Volve a un commit anterior borrando todo o rastro dos posteriores.
git reset –hard HEAD~1	Volve á penúltima revisión eliminado commits previos.
git restore <arquivo(s)>	Para recuperar archivo(s) borrados do working directory que queremos recuperar.
git restore –staged <arquivo(s)> **git restore <arquivo(s)>**	Para recuperar archivo(s) borrados da zona staged que queremos recuperar.
git revert <id_commit>	Creará un novo commit descartando ata o commit do que se indica o id (incluído) pero sen descartar os cambios. Se hai conflitos haberá que resolvelos manualmente. No caso de conflito, unha vez resolto: (1) git add . (2) git revert –continue
Incorporar ficheiros a commit	
git commit --amend	Engade o que teñamos na área de preparación ao último commit ou o modifica, de ser o caso.
Instantáneas temporales	
git stash	Garda unha instantánea do que estamos facendo, sen facer commit e revirte a situación ao último commit.
git stash list	Lista as instantáneas creadas con stash.
git stash apply <id_instantánea>	Volve á instantánea indicada. Pode haber que resolver os conflitos, procedendo coma no merge.
git stash drop <id_instantánea>	Elimina a instantánea indicada.
Operacións con repositorios remotos	
Vincular repositorio remoto ao repositorio local	
git remote add origin <url>	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).
git remote remove <nome_repo>	Elimina este enlace ao repositorio en local.
Subir información ao repositorio remoto	
git push origin master	Sube a rama master ao repositorio remoto origin.
git push -u origin master	Selecciona por defecto a subida da rama master ao repositorio origin cada vez que se faga git push. (pódese substituír -u por –set-upstream)
Descargar información do repositorio remoto	
git pull origin master	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.
git fetch	Comproba os cambios feitos no repositorio ou rama pero non descarga arquivos.
git clone <url> .	Inicializa o repositorio e descarga un repositorio remoto. Realiza git init, git remote add e git pull dunha soa vez.