**Mostrar la versión de git**: git --version

**Mostrar la ayuda de git**: git help

**Mostrar la ayuda del comando commit de git**: git help commit

**Configurar el nombre de usuario**: git config --global user.name "fulano"

**Configurar Email**: git config --global user.email [fulano@gmail.com](mailto:fulano@gmail.com)

**Configurar el editor**: git config --global core.editor "notepad"

**Mostrar las dos credenciales que hemos configurado**: git config --global -l

**Activar el marco de colores para los comandos**: git config --global color.ui true

**Iniciar repositorio**: git init

**Añadir todos los archivos para el commit**: git add .

**Añadir un archivo para el commit**: git add archivo

**Añadir todos los archivos con la extensión especificada**: git add \*.txt

**Añadir todos los archivos dentro de un directorio y de una extensión especifica:** git add docs/\*.txt

**Añadir todos los archivos dentro de un directorio**: git add docs/

**Consultar el estado del stage:** git status

**Retirar los cambios en un archivo del stage:** git reset archivo

**Confirmar cambios en el repositorio**: git commit -m “Texto que identifique por que se hizo el commit”

**Actualizar el stage con los cambios de archivos y confirmar cambios en el repositorio:** git commit -am "Texto que identifique por que se hizo el commit"

**Mostrar los cambios realizados en un archivo**: git diff

**Mostrar los logs de los commits**: git log

**Mostrar los logs de los commits con sus identificadores**: git log --oneline

**Restaurar el estado del repositorio al almacenado en el último commit**: git checkout -- .

**Deshacer el último commit anulando los cambios en el repositorio, permaneciendo estos en la carpeta y también en el *stage****:* git reset --soft HEAD~1 (Nota: HEAD~1 desharía los dos últimos commits y así sucesivamente con otros números)

**Deshacer el último commit anulando los cambios en el repositorio, permaneciendo estos en la carpeta y pero no en el *stage****:* git reset --mixed HEAD~1

**Deshacer el último commit anulando los cambios en el repositorio, deshaciendo estos cambios de la carpeta y también del *stage****:* git reset --hard HEAD~1

**Deshacer el commit (hard, mixed o soft) apuntando a su identificador***:* git reset --soft identificador

**Ver el histórico de commits del repositorio incluyendo los que se han eliminado***:* git reflog

**Crear una rama**: git branch rama

**Cambiar de rama**: git checkout rama

**Eliminar una rama**: git branch -d rama

**Fusionar el contenido de una rama con la principal**: git merge rama

**Mostrar todas las ramas del repositorio**: git branch

**Crear un tag para versionar un repositorio**: git tag v1.0.0

**Mostrar todos los tags:** git tag

**Incorporar cambios desde la rama master a otra rama, una vez ubicados en la rama de destino:** git checkout master

**Actualizar cambios desde la rama master a otra rama, una vez ubicados en la rama de destino:** git rebase master

**Actualizar cambios desde la rama master a otra rama a partir de un commit, una vez ubicados en la rama de destino:** git cherry-pick master identificadordelcommit

**Clonar un repositorio remoto:** git clone urlrepositorio

**Subir cambios a un repositorio remoto**: git push

**Subir cambios a una rama del repositorio**: git push origin nombrerama

**Actualizar nuestro repositorio local desde el repositorio remoto**: git pull

**Actualizar una rama concreta nuestro repositorio local desde el repositorio remoto**: git pull nombrerama

**Borrar una rama del repositorio remoto:** git push origin -d rama

**Actualizar todas las ramas de nuestro repositorio local a nuestro repositorio remoto**: git push –all

**Crear un repositorio nuevo en Github**: https://github.com/new

**Eliminar un repositorio de Gihub**: Entrar al repositorio, ir a la opción *Settings,* buscar la opción *Delete this repository* y confirmar la eliminación escribiendo lo que señala la ventana.