

Comandos Git
Configurar Nombre
<i>git config --global user.name Kike-Gomez</i>
Configurar Email
<i>git config --global user.email kike.gomez.uy@gmail.com</i>
Iniciamos GIT en la carpeta donde está el proyecto
<i>git init</i>
Clonamos el repositorio de github
<i>git clone &lt;url&gt;</i>
Lista el estado actual del repositorio con lista de archivos modificados o agregados
<i>git status</i>
Hacemos el primer commit
<i>git commit -m "Texto que identifique por que se hizo el commit"</i>
Subimos al repositorio
<i>git push origin master</i>
Añadimos todos los archivos para el commit
<i>git add .</i> <i>git add -A</i>
Añadimos el archivo para el commit
<i>git add &lt;archivo&gt;</i>
Añadimos todos los archivos con la extensión especificada
<i>git add *.txt</i>
Añadimos todos los archivos dentro de un directorio y de una extensión específica
<i>git add docs/*.txt</i>
Añadimos todos los archivos dentro de un directorio
<i>git add docs/</i>
Cargar en el HEAD los cambios realizados
<i>git commit -m "Texto que identifique por que se hizo el commit"</i>
Agregar y Cargar en el HEAD los cambios realizados
<i>git commit -a -m "Texto que identifique por que se hizo el commit"</i>
De haber conflictos los muestra
<i>git commit -a</i>
Agregar al último commit, este no se muestra como un nuevo commit en los logs. Se puede especificar un nuevo mensaje
<i>git commit --amend -m "Texto que identifique por que se hizo el commit"</i>
Crea un nuevo tag
<i>git tag -a &lt;version&gt; -m "esta es la versión x"</i>
Subimos al repositorio
<i>git push &lt;origin&gt; &lt;branch&gt;</i>
Subimos un tag
<i>git push --tags</i>
Busca los cambios nuevos y actualiza el repositorio
<i>git pull origin &lt;nameBranch&gt;</i>
Muestra una lista de todos los tags
<i>git tag</i>
Muestra los logs de los commits
<i>git log</i>
Muestra los cambios en los commits
<i>git log --oneline --stat</i>
Muestra gráficos de los commits

<i>git log --oneline --graph</i>
Muestra los cambios realizados a un archivo
<i>git diff</i>
<i>git diff --staged</i>
Saca un archivo del commit
<i>git reset HEAD &lt;archivo&gt;</i>
Devuelve el ultimo commit que se hizo y pone los cambios en staging
<i>git reset --soft HEAD^</i>
Devuelve el ultimo commit y todos los cambios
<i>git reset --hard HEAD^</i>
Devuelve los 2 ultimo commit y todos los cambios
<i>git reset --hard HEAD^^</i>
Rollback merge/commit
<i>git log</i>
<i>git reset --hard &lt;commit_sha&gt;</i>
Agregar repositorio remoto
<i>git remote add origin &lt;url&gt;</i>
Cambiar de remote
<i>git remote set-url origin &lt;url&gt;</i>
Remover repositorio
<i>git remote rm &lt;name/origin&gt;</i>
Muestra lista repositorios
<i>git remote -v</i>
Muestra los branches remotos
<i>git remote show origin</i>
Limpiar todos los branches eliminados
<i>git remote prune origin</i>
Limpiar todos los branches eliminados
<i>git remote prune origin</i>
Crea un branch
<i>git branch &lt;nameBranch&gt;</i>
Lista los branches
<i>git branch</i>
Comando -d elimina el branch y lo une al master
<i>git branch -d &lt;nameBranch&gt;</i>
Elimina Branch sin preguntar
<i>git branch -D &lt;nameBranch&gt;</i>
Quita del HEAD un archivo y le pone el estado de no trabajado
<i>git checkout -- &lt;file&gt;</i>
Crea un branch en base a uno online
<i>git checkout -b newlocalbranchname origin/branch-name</i>
Cambiar de branch
<i>git checkout &lt;nameBranch/tagname&gt;</i>
Une el branch actual con el especificado
<i>git merge &lt;nameBranch&gt;</i>
Verifica cambios en el repositorio online con el local
<i>git fetch</i>
Borrar un archivo del repositorio
<i>git rm &lt;archivo&gt;</i>
Descargar remote de un fork

<i>git remote add upstream &lt;url&gt;</i>
Merge con master de un fork
<i>git fetch upstream</i> <i>git merge upstream/master</i>