Creo un directorio de trabajo

- mkdir ~/ejemplo/
- cd ~/ejemplo/
- git status
- ls -la

Inicializo el repositorio. “git init” o “git clone”

- git init
- ls
- ls -la
- git status

Creo un archivo index.htm

Creo un archivo nuevo en ese directorio

- touch primer_archivo.txt
- ls
- git status

Lo añado al repositorio, para que sea seguido:

- git add primer_archivo.txt
- Podía hacer git add .
- git status

Hago el commit:

- git commit -m “Añado el primer archivo vacío”.
- git status

Creo dos archivos nuevos

- touch segundo_archivo.txt
- touch tercer_archivo.txt
- git status
- git add segundo_archivo.txt
- git status
- git commit -m “Añado el segundo archivo”
- git status
- git add tercer_archivo.txt
- git status
- git commit -m “Añado el tercer archivo”
- git log
- git log --oneline

Edito el primer archivo

- echo “Creo una primera línea en el primer archivo” >> primer_archivo.txt
- git status
- git add primer_archivo.txt

- git status
- echo “Creo una primera línea en el segundo archivo.” >> segundo_archivo.txt
- echo “Creo una primera línea en el tercer archivo.” >> tercer_archivo.txt
- git status
- git add segundo_archivo.txt
- git status
- git commit -m “Introduzco una línea en el primer y en el segundo archivo”
- git status
- git log
- git log --oneline
- git add tercer_archivo.txt
- git commit -m “Introduzco una línea en el tercer archivo”
- git status

Ver cambios con diff

- echo “Creo una segunda línea en el primer archivo” >> primer_archivo.txt
- cat primer_archivo.txt
- git status
- git diff
- echo “Creo una segunda línea en el segundo archivo” >> segundo_archivo.txt
- cat segundo_archivo.txt
- git status
- git diff
- git diff segundo_archivo.txt

Ver cambios en el índice

- git add segundo_archivo.txt
- git status
- git diff
- git diff --staged
- git diff --cached
- git diff HEAD
- git add primer_archivo.txt
- git status
- git diff
- git diff --staged
- git commit -m “Añado líneas en el primer y en el segundo archivo”
- git status
- git diff
- git diff --staged

Eliminar archivos

- touch temp1.txt
- touch temp2.txt
- git status

- git add .
- git status
- git commit -m “Añado dos archivos de prueba para borrar”
- rm temp1.txt
- git status
- git add temp1.txt
- git rm temp1.txt
- git status
- git commit -m “Borro el archivo temp1.txt”
- git status
- git log –oneline
- git rm temp2.txt
- git status
- git commit -m “Borro el archivo temp2.txt”
- git status
- git log --oneline

Mover y renombrar archivos

- mv primer_archivo.txt archivo_primero.txt
- git status
- git add archivo_primero.txt
- git rm primer_archivo.txt
- git status
- git mv segundo_archivo.txt archivo_segundo.txt
- git status
- mkdir dir1
- git mv tercer_archivo.txt dir1/archivo_tercero.txt
- git status
- git commit -m “Cambios los nombres de los archivos y la estructura de los directorios”
- git status
- git log --oneline

Deshaciendo cambios en la zona de trabajo

- git status
- echo “Añado una tercera línea al primer archivo” >> archivo_primero.txt
- cat archivo_primero.txt
- git status
- git diff
- git checkout archivo_primero.txt
- git checkout – archivo_primero.txt
- git status

Quitando archivos del índice

- git status
- echo “Añado una tercera línea al primer archivo” >> archivo_primero.txt
- cat archivo_primero.txt
- git status
- git add archivo_primero.txt
- git status
- git reset HEAD archivo_primero.txt
- git status

Cambiando el último commit

- git add archivo_primero.txt
- git commit -m “Añado una tercera línea en el archivo primero”
- git log
- git log --oneline
- echo “Añado una cuarta línea al primer archivo” >> archivo_primero.txt
- git status
- git add archivo_primero.txt
- git commit --amend -m “Añado dos líneas en el archivo primero”

Recuperando versiones antiguas de archivos

- git log --oneline
- cat archivo_primero.txt
- git checkout $SHA_commit archivo_primero.txt //Un commit antes
- cat archivo_primero.txt
- git diff --staged
- git checkout – archivo_primero.txt // git checkout HEAD archivo_primero.txt
- git status

Revirtiendo un commit

- git log –oneline
- git revert HEAD
- git status

Deshaciendo commits: reset

- Soft
- Mixed
- Hard

Copio los SHA-1 a un archivo temporal, porque los voy a necesitar

Soft reset

- git status
- cat .git/HEAD
- cat .git/refs/heads/master

- git reset --soft $id_commit_2 //4º último commit
- cat .git/refs/heads/master
- git log
- git status
- git diff --staged
- git reset --soft $id_commit_1 //commit donde estábamos al inicio
- git log –oneline

Mixed reset //5cb18dc

- git status
- cat .git/refs/heads/master
- git reset –mixed $id_commit_2 //4º último commit
- cat .git/refs/heads/master
- git log
- git status
- git diff –staged
- git diff
- git reset --mixed $id_commit_1 //commit donde estábamos al inicio
- git status
- git log –oneline

Hard reset

- git status
- cat .git/refs/heads/master
- git reset --hard $id_commit_2 //4º último commit
- cat .git/refs/heads/master
- git log --oneline
- git status
- git reset --hard $id_commit_1 //commit donde estábamos al inicio
- git status
- git log –oneline

Borrar archivos no seguidos

- touch basura1.txt
- touch basura2.txt
- touch basura3.txt
- git status
- git clean -n //Lo que debería de borrar
- git add basura1.txt
- git status
- git clean -f
- git status
- git reset HEAD basura1.txt
- git clean -f
- git status

Ignorar archivos

- touch archivo_temporal.txt
- git status
- touch .gitignore
- git status
- echo “archivo_temporal.txt” >> .gitignore
- git status
- mkdir log
- touch log/untrack01.log
- mkdir images
- touch images/untrack01.jpg
- echo “*.zip” >> .gitignore
- echo “*.gz” >> .gitignore
- echo “log/*.log” >> .gitignore
- echo “log/*.log[0-9]” >> .gitignore
- echo “images/” >> .gitignore
- echo “! images/logo.png” >> .gitignore
- git status
- git add .gitignore
- git commit -m “Añadido el archivo .gitignore”
- git status
- git log --oneline
-

Ignorar archivos seguidos

- touch archivo_temporal2.txt
- git status
- git add archivo_temporal2.txt
- git commit -m “Añado el archivo_temporal_2.txt”
- git status
- echo “archivo_temporal2.txt” >> .gitignore
- cat gitignore
- echo “Primera línea del archivo_temporal2.txt” >> archivo_temporal2.txt
- git status
- ///// git rm archivo_temporal2.txt ///// No me interesa borrarlo
- git rm --cached archivo_temporal2.txt //// No existe --staged
- ls
- git status
- git commit -am “Actualizo el archivo .gitignore, incluyendo archivo_temporal2.txt”
- git status
- git log --oneline

Seguir directorios vacíos

- git status
- mkdir media/
- git status
- touch media/.gitkeep
- ls -la media/
- git status
- git add media/
- git commit -m “Añado el archivo .gitkeep en el directorio media “
- git status
- git log –oneline

Crear una rama

- git branch
- ls -la .git
- cat .git/HEAD
- ls -la .git/refs/heads/
- cat .git/refs/heads/master
- git log --oneline -2
- git branch desarrollo
- git branch
- ls -la .git/refs/heads/
- cat .git/refs/heads/master
- cat .git/refs/heads/desarrollo

Cambio de rama

- git branch
- cat .git/HEAD
- git checkout desarrollo
- cat .git/HEAD

Edición en esta nueva rama

- echo “Inserto una línea en el archivo primero” >> archivo_primero.txt
- git status
- git commit -am “Introduzco una nueva línea en el archivo_primero.txt de la rama de desarrollo”
- git log –oneline -5
- git checkout master
- git log –oneline -5
- cat archivo_primero.txt
- git checkout desarrollo
- cat archivo_primero.txt
- git log –oneline -5
- cat .git/HEAD
- git show HEAD

Crear una rama y cambiarse a ella

- git checkout master
- git checkout -b experimento
- echo “Experimento con una nueva línea en el archivo primero” >> archivo_primero.txt
- git status
- git commit -am “Experimento con una nueva línea en el archivo_primero.txt de la rama experimento”
- git log –oneline -5
- cat archivo_primero.txt
- git checkout master
- git log –oneline -5
- cat archivo_primero.txt
- git checkout desarrollo
- cat archivo_primero.txt
- git log –oneline -5

Renombrando una rama

- git branch
- git branch -m experimento prueba
- git branch

Borrando una rama

- git branch rama_temporal
- git branch
- git branch -d rama_temporal
- git checkout desarrollo

Visualizando gráficamente las ramas

- git log --oneline -6 --graph
- git log --oneline -6 --graph --all
- git log --oneline -6 --graph --all --decorate

Otros filtros de git log

- git log --oneline --since=2014-01-24
- git log --oneline --until=2014-01-25
- git log --oneline --author=”Jesus”
- git log --oneline --grep=”gitignore”
- git log --stat --summary
- git log --oneline --stat --summary

Mostrar información de un commit (u otro objeto)

- git branch
- git log --oneline
- git show HEAD

- git show --format=oneline HEAD
- git show --oneline HEAD
- git show --oneline HEAD^
- git show --oneline HEAD^^
- git show --oneline HEAD~2
- git show --oneline HEAD~6
- git show --oneline HEAD~5
- git show --oneline $SHA1_commit

Comparando commits

- git diff prueba..desarrollo
- git diff desarrollo..prueba
- git diff prueba^..desarrollo
- git diff prueba^..desarrollo^ //// Vacío
- git log --oneline -6 --graph --all --decorate
- git diff prueba..desarrollo archivo_segundo.txt /// Vacío
- git log --oneline
- git diff --stat --summary $SHA1_commit_inicial..HEAD // Trabajo realizado
- git diff --stat --summary $SHA1_commit_inicial.. .gitignore
- git diff $SHA1_commit_inicial.. .gitignore

Fusionando ramas

- git log --oneline -6 --graph --all --decorate
- git checkout master
- git merge desarrollo ///// fast-fordward

Conflicto

- git merge prueba
- cat archivo_primero.txt

Soluciones

- Abortar
- Resolver manualmente
- Herramientas de fusión

Abortar

- git merge --abort
- git status
- cat archivo_primero.txt

Resolver el problema de forma manualmente

- git merge prueba
- cat archivo_primero.txt
- Edito el archivo y resuelvo los cambios
- git status

- git add archivo_primero.txt
- git commit -m “Resuelto el conflicto en la línea 4 entre las ramas master y prueba”
- git log --oneline -6 --graph --all --decorate

Etiquetas

- git tag
- git tag -a v1.0 $SHA1-commit -m “Versión 1.0” //4º último commit
- git tag
- git show v1.0
- git tag -a v1.1. -m “Versión 1.1. Fusionadas las ramas de desarrollo y master”
- git tag
- git show tag v1.1
- git tag -l “v1.*”
- git tag -l “v*.1”