

### Estados de git

Working directory

Archivos sin seguimiento

Staging area

Archivos con seguimiento

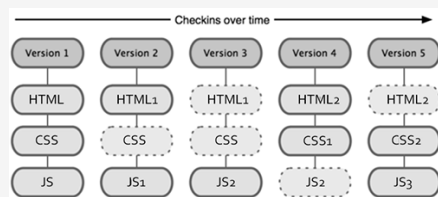
git directory (repository)

Archivos en el repositorio

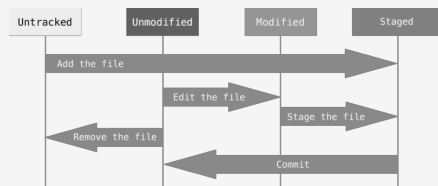
Conociendo el estado de los ficheros

```
git status
```

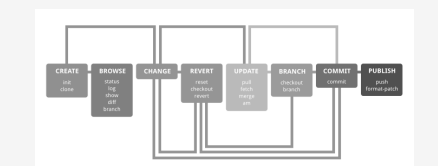
### Ficheros en git



### Estados de ficheros en git



### Workflow



### Creación de repositorios

Desde un servidor

```
git clone url [nombre carpeta]
```

En una carpeta con código

```
git init
```

En una carpeta nueva

```
git init nombre carpeta
```

Clonar todo el repositorio

```
git clone --mirror
```

### De working directory a staging area

Un fichero concreto

```
git add ruta-fichero
```

Todos los ficheros creados, modificados y eliminados

```
git add -A
```

Todos los ficheros creados y modificados

```
git add .
```

Todos los ficheros modificados y eliminados

```
git add -u
```

### De staging area a working directory

Quitar uno o varios ficheros o directorios del área de staging

```
git rm --cached fichero1 fichero2 directorio1
```

```
..
```

Quitar uno o varios ficheros del área de staging y borrarlos del disco

```
git rm -f fichero1 fichero2 directorio1...
git branch -m "rama" "nombre"
```

### De staging area a git directory

Guardando los cambios en el repositorio (Hacer un commit)

```
git commit -m "comentario"
```

Modificando el ultimo cambio (commit) guardado

```
git commit --amend
```

### Guardando cambios temporalmente

Haciendo un stash con los cambios actuales

```
git stash
```

Ver la lista de estados guardados

```
git stash list
```

Borrar algun estado

```
git stash drop stash@{n}
```

Cojiendo los cambios del ultimo estado guardado

```
git stash apply
```

Coger algún estado concreto

```
git stash apply stash@{n}
```

### Guardando cambios temporalmente (cont)

Agregar los ficheros al staging area y guardar todo como estado

```
git stash -u
```

Limpiar los estados guardados

```
git stash clear
```

### Trabajando con ramas

Creando una rama

```
git branch rama
```

Crear rama y moverte a ella

```
git checkout -b rama
```

Cambiar a una rama

```
git checkout rama
```

Cambiar a un commit

```
git checkout HASH
```

Renombrar una rama

```
git branch -m "rama" "nombre"
```

```
"
```

Listando ramas

```
git branch -l
```

Borrando una rama

```
git branch -d <nombre de rama>
```

Borrando una rama con commits

```
git branch -D <nombre de rama>
```

### Revisando cambios entre commits

Compara staging area con el ultimo commit

```
git diff
```

Comparando dos commits

```
git diff HASH1 HASH2
```

Comparando dos tags

```
git diff TAG1 TAG2
```

Comparando un archivo

```
git diff Nombre_archivo
```

Diferencia desde staged

```
git diff --staged
```



### Reseteando cambios

Borras todos los commit asta cierto punto pero mantienes los cambios en el area de stagin

```
git reset --soft HASH
```

Igual que --soft pero dejando los cambios en working directory

```
git reset --mixed
```

Borra los commits y no mantiene los cambios

```
git reset --hard HASH
```

Deshader todos los cambios del working directory

```
git reset --hard
```

Deshadcer los cambios de un fichero

```
git checkout fichero
```

### Etiquetando estados

Etiquetando el commit actual

```
git tag -a "eti que ta" -m "mensaje de etiqueta"
```

Etiquetnado un commit concreto

```
git tag "eti que ta" HASH
```

Renombrando una etiqueta

```
git tag -f -a "eti que ta" -m "mensaje de etiqueta" HASH
```

Borrando una etiqueta

```
git tag -d "eti que ta"
```

Ver lista de tags

```
git tag -l
```

### Uniando commits

Rebasando commits desde el actual asta el n-esimo, sera necesario hacer push con -f

```
git rebase -i HEAD~n
```

Rebasando comit actual con otra rama

```
git rebase rama
```

Mezclando ramas en la actual

```
git merge rama_que_que_remos_mezclar
```

Copiando un commit selectivamente

### Uniando commits (cont)

```
git cherry -pick HASH
```

### Trabajando en remoto

Agregando un servidor remoto

```
git remote add nombre url
```

Quitar un servidor remoto

```
git remote remove nombre url
```

Traer cambios sin sobrescribir

```
git fetch
```

Traer cambios y sobrescribir

```
git pull
```

Subiendo cambios

```
git push -u remote rama
```

Forzando subida

```
git push -f remote rama
```

Borrar una rama remota

```
git push remote --delete rama
```

Subiendo o trayendo todos los cambios

Agregar --all al final

Subiendo una etiqueta

```
git push -u remote etiqueta
```

Borrando una etiqueta

```
git push --delete remote tag
```

Desacer un push

```
git push -f origin ultimo_commit_correcto:rama
```

### Ver la historia

Ver los commits del repositorio

```
git log
```

En una sola linea

```
git log --oneline
```

Con grafico

```
git log --online --graph
```

Los ultimos n commits

```
git log --online --graph -n
```

### Configurando git

Modificando el editor

```
git config --global core.editor "
```

Estableciendo el usuario

```
git config --global user.name = "
```

Estableciendo el email

```
git config --global user.email = c
```

Estableciendo alias

```
git config --global alias.s up erl w )%d %C( reset)' --all"
```

### Trucos

Uniando todos los commits en uno

```
git rebase -i HEAD --root
```

Conociendo quien cambio un fichero

```
git blame file
```