

Estados de git

Working directory

Archivos sin seguimiento

Staging area

Archivos con seguimiento

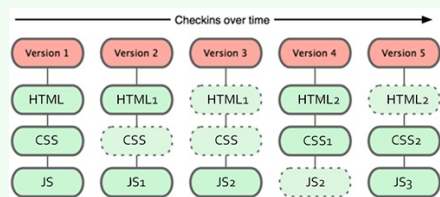
git directory (repository)

Archivos en el repositorio

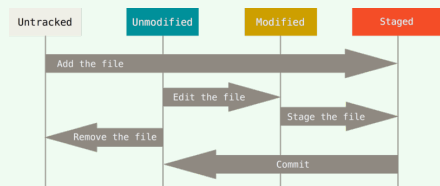
Conociendo el estado de los ficheros

```
git status
```

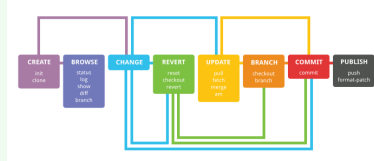
Ficheros en git



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
fichero del área de staging

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

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

```
git rm -f fichero1 fichero2  
directorio1...
```

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 stasth drop stash@{n}
```

Cojiendo los cambios del ultimo estado
guardado

```
git stash apply
```

Coger algún estado concreto

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



By Tlloc-Es

cheatography.com/tlloc-es/

Published 29th February, 2020.

Last updated 21st April, 2020.

Page 1 of 3.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

Guardando cambios temporalmente (cont)

Agregar los ficheros al stagin 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 ramas

```
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
```

Comprando 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 checkot fichero
```

Etiquetando estados

Etiquetando el commit actual

```
git tag -a "etiqueta" -m "-  
mensaje de la etiqueta"
```

Etiquetnado un commit concreto

```
git tag "etiqueta" HASH
```

Renombrando una etiqueta

```
git tag -f -a "etiqueta" -m "-  
mensaje de la etiqueta" HASH
```

Borrando una etiqueta

```
git tag -d "etiqueta"
```

Ver lista de tags

```
git tag -l
```

Uniendo 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_queremos_  
mezclar
```

Copiando un commit selectivamente

```
git cherry-pick HASH
```



By Tlaloc-Es

cheatography.com/tlaloc-es/

Published 29th February, 2020.

Last updated 21st April, 2020.

Page 2 of 3.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

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 trallendo todos los cambios

Agregar --all al final

Subiendo una etiqueta

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

Borrando una etiquetabarar

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

Desacer un push

```
git push -f origin ultimo_commi-  
t_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  
"editor --wait"
```

Estableciendo el usuario

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

Estableciendo el email

```
git config --global user.email =  
correo@correo.es
```

Estableciendo alias

```
git config --global alias.sup-  
erlog "log --graph --abbrev--  
commit --decorate --date=re-  
lative --format=format:'%C(bold  
blue)%h%C(reset) - %C(bold  
green)(%ar)%C(reset) %C(white)-  
%s%C(reset) %C(dim white)-  
%an%C(reset)%C(bold yellow)%d-  
%C(reset)' --all"
```

Trucos

Uniendo todos los commits en uno

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

Conociendo quien cambio un fichero

```
git blame file
```

