

Git desde cero

`git commit -m ""`

`git diff`: Compara lo que tenemos en el directorio del trabajo con lo que esta en el area de preparacion

Para hacer un commit desde mi editor podemos usar solo `git commit` y automaticamente se abra nuestro editor con el que configuramos nuestro git.

Configurando Git por primera vez

```
git config --global user.name "jdpoccorie"
git config --global user.email juandiego.poccori@gmail.com
git config --global core.editor nano
git config --list
```

.gitignore

Patrones de archivos que git ignorará

Saltar el area de preparación

`git commit -a -m` para saltar el area de preparación, tener en cuenta que solo funciona para archivos que ya estamos siguiendo.

`git rm`

`git rm` Elimina archivos rastreados del repositorio y de nuestro directorio de trabajo de manera que no aparezcan la próxima vez como archivos no rastreados.

si nosotros eliminamos el archivo, haciendo anticlick y mandar a la papelera como una simple eliminacion y ponemos `git status` en consola nos aparecera un mensaje que diga deleted y que si vamos a querer continuar con eso usemos `git add/rm` o que si queremos descartar esa eliminacion del archivo tenemos que usar `git checkout -- nombre-archivo` y todo regresa a la normalidad.

Renombrar archivos

`git mv` Renombrar archivos `git mv file_from file_to` su equivalente seria:

1. `git rm file_from`
2. `git add file_to`

git log

Muestra el historial de confirmaciones

Entre las opciones del comando pordemos encontrar:

- `--oneline` nos muestra el historial abreviado
- `--graph` añade un pequeño ASCII mostrando el historial de ramificaciones y uniones. Podemos combinarlo
- `git log --decorate` Nos muestra los puntos de las ramas
- `git log --oneline --all` Nos muestra todas las ramas
- `git log --oneline --all --graph` Nos muestra todas las ramas Para ver los ultimos dos log
- `git log -2` Para ver los dos ultimos commit
- `git log -3` Para ver los tres ultimos commit
- `git log --pretty=format:"%h - %an, %ar : %s"` Nos resultará algo así

```
8162bfe - JDiegoP, 9 minutes ago : agregando la parte para configurar nuestro git de manera global
df533c9 - JDiegoP, 17 minutes ago : Agregando los equivalentes para git mv
```

```
a2bb140 - JDiegoP, 20 minutes ago : Renombrando por el camino largo
a7dab93 - JDiegoP, 23 minutes ago : Se coloco en may<C3><BA>scula el archivo teoria.md a Teoria.md
a51237b - JDiegoP, 40 minutes ago : Eliminando el 01.mp4 y ademas aumente info a la teoria
eed3109 - JDiegoP, 47 minutes ago : archivos para que lo podamos eliminar
```

¿Cuales son los commit del día de ayer?

- `git log --after="YYYY-MM-dd"` para ver los commit despues de la fecha qe le demos
- `git log --after="YYYY-MM-dd 00:00:00"` para ver los commit despues de la fecha qe le demos ademas de la hora claro.
- `git log --before="YYYY-MM-dd"` para ver los commit antes de la fecha qe le demos.
- `git log --after="YYYY-MM-dd" --before="YYYY-MM-dd"` para ver los commit antes de la fecha qe le demos.

Comandos para deshacer

git commit --amend

`git commit --amend` Rehace la confirmacion ultima y ademas nos lo abre en el editor configurado.

Este comando utiliza el area de preparacion para la confirmacion.

Al final terminaras con una sola confirmacion - la segunda confirmacion reemplaza el resultado de la primera

git reset HEAD

`git reset HEAD <file>` Deshace la preparacion

El archivo o los archivos salen del area de preparacion y se quedan en el directorio de trabajo.

git checkout --

Descarta un archivo modificado

Este es un comando peligroso

Cualquier cambio que le hayas hecho a file desaparecerá.

Nunca utilices este comando a menos de que este absolutamente seguro de que ya no quieres los cambios del archivo

Primer vistazo para lo que es clonar

`git clone https://github.com/escueladigital/ED-GRID.git`: Ahi descarga todo el historial del proyecto ademas de los archivos.

`git clone https://github.com/escueladigital/ED-GRID.git mi-propio-grid` Para que lo creemos en una carpeta mi-propio-grid 😊

Etiquetas en Git

Poner un marcador a cambios importantes

git tag

Lista las etiquetas

Este comando lista las etiquetas en orden alfabetico; el orden en el que aparecen no tiene mayor importancia.

Nos lista todas las etiquetas segun orden alfabetico ya sean anotadas o ligeras.

Etiquetas ligeras

`git tag nombre-de-la-etiqueta`

Una etiqueta ligera no es mas que el checksum(apuntador) un alias de un commit guardado en un archivo, no incluye mas informacion. Para crear una etiqueta ligera, no pasamos las opciones `-a`, `-s` ni `-m`

Es mas recomendable usar las...

Etiquetas anotadas

```
git tag -a v1.0.0 -m "my version 1.0.0"
```

Se guardan en la base de datos de Git como objetos enteros. Tienen un checksum; contienen nombre del etiquetador, correo electronico y fecha y tienen un mensaje asociado, l

Si no le pasamos el `-m` se nos abrira el editor ya que el mensaje es obligatorio

- ¿Entonces como es que uno se guarda en la base de datos?
 - `git show nombre-etiqueta`

Para poner una etiqueta en un punto de la historia necesitamos el hash de este ejem `git tag ignorados-iniciales a20cf43`

Tener en cuenta de que es igual

- `git show a20cf43` y `git show nombre-de-la-etiqueta-en-a20cf43`

Filtrado de etiquetas

Tener en cuenta de que las etiquetas tienen que tener algo en comun en sus nombres como `v1.0.0 v2.0.0 ...`

- `git tag -l "v0.*"` la opcion `-l` nos ayuda para poner patrones y asi filtrarlos, esto hace que todas las versiones que epiezan con `v0.*` y que de ahi le siga lo que sea.

Nota: Para eliminar una etiqueta es `git tag -d nombre_etiqueta`

Ramas

`git branch nombre-de-la-rama`

Para crear una rama

`git checkout nombre-de-la-rama`

Para movernos de rama en rama

- Puedo crear todas las ramas que quiera y/o necesite
- Las ramas nuevas que se crean apuntan al commit en donde este ubicado el `HEAD`.
- Las ramas en git divergen es decir que cuando nosotros hagamos un commit en la rama master este apanzara a la par de la rama `testing`

OJO: Tambien podemos usar `git commit -am "message"` para poder saltar el area de preparaci3n

Fusiones

Para fusionar usamos

`git merge`

```
git merge otra_rama
```

Incorpora `otra_rama` en la rama actual

`git branch -d`

`git branch -d nombre_rama`

Elimina nombre_rama si ya ha sido fusionada con la rama actual

Solo funciona si ya ha sido fusionada con otra rama, si este no ha sido fusionada no funcionara.

git branch -D

`git branch -D nombre_rama`

Elimina nombre_rama este o no fusionada con la rama actual

Se fuerza el borrado, se pierden los cambios

Esta opción es muy peligrosa, tenemos que usarla con mucha precaución

Nota: Para saber que ramas no estan fusionadas aun podemos usar `git branch --no-merged` y para ver los fusionados podemos usar `git branch --merged`

GitHub

Para subir nuestros repositorios en la nube, esto es FREE si trabajamos open source.

`git remote add origin https://github.com/jdpoccorie/apuntes-git.git`

Con este comando vinculamos nuestro repositorio local con un repositorio en la nube