

Git

Deshaciendo cambios en Git

Indice

- Reset commit
 - Hard
 - Soft
 - Mixed
- Reset commit file
 - vs git rm
- Checkout commit
- Checkout Commit file
- Revert

Commit v1

- `git commit -am "v1"`

HISTORY

f1.txt
v1

STAGED

f1.txt
v1

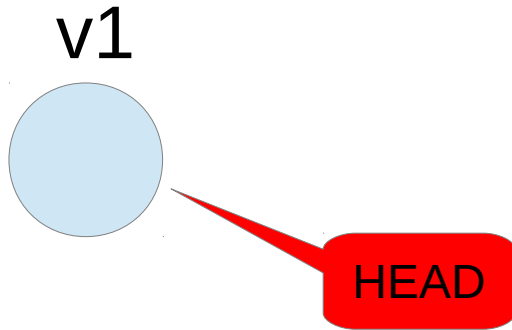
CARPETA LOCAL

f1.txt
v1

Commit v1

- `git log --pretty="format:%h %s"`

0d588ed v1



Commit v2

- `git commit -am "v2"`

HISTORY

f1.txt
v1 v2

f2.txt
v2

STAGED

f1.txt
v1 v2

f2.txt
v2

CARPETA LOCAL

f1.txt
v1 v2

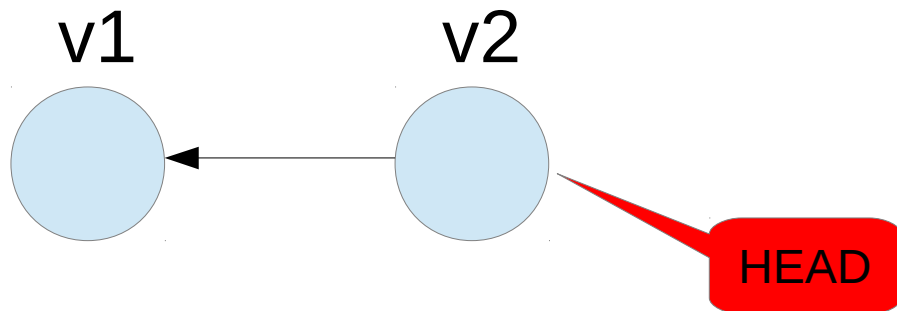
f2.txt
v2

Commit v2

- `git log --pretty="format:%h %s"`

a3aeb2a v2

0d588ed v1



Commit v3

- `git commit -am "v3"`

HISTORY

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

CARPETA LOCAL

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

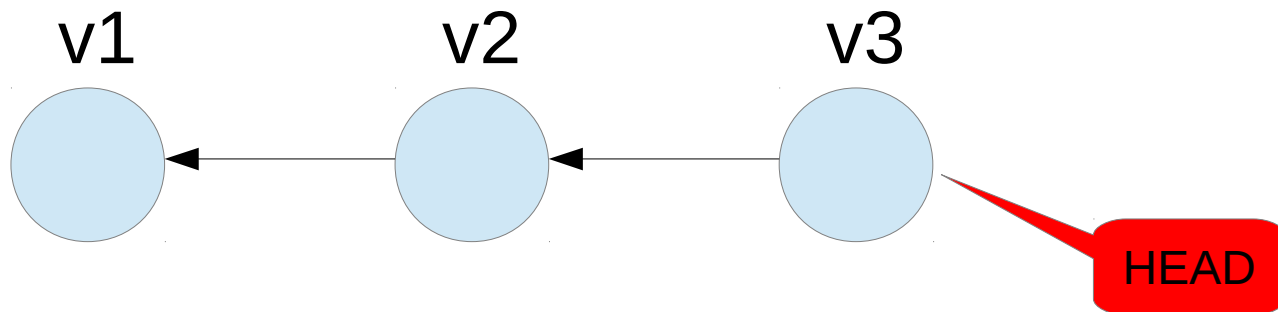
Commit v3

- `git log --pretty="format:%h %s"`

5687d92 v3

a3aeb2a v2

0d588ed v1





```
git reset --hard <commit>
```

git reset --hard

- git reset --hard 0d588ed

HISTORY

f1.txt
v1

STAGED

f1.txt
v1

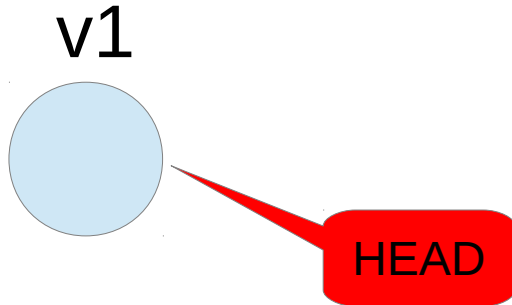
CARPETA LOCAL

f1.txt
v1

git reset --hard

- `git log --pretty="format:%h %s"`

0d588ed v1



git reset --hard

- git status

On branch master
nothing to commit, working tree clean



```
git reset --soft <commit>
```

git reset --soft

- git reset --soft 0d588ed

HISTORY

f1.txt
v1

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

CARPETA LOCAL

f1.txt
v1 v2 v3

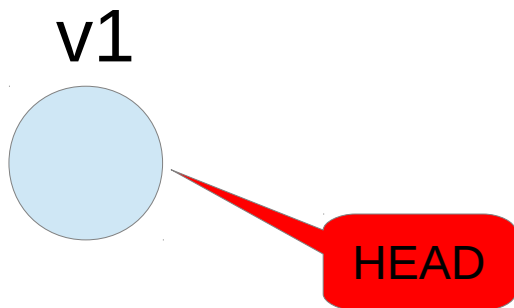
f2.txt
v2 v3

f3.txt
v3

git reset --soft

- `git log --pretty="format:%h %s"`

0d588ed v1



git reset --soft

- git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: f1.txt

new file: f2.txt

new file: f3.txt



```
git reset –mixed  
<commit>
```

git reset --mixed

- git reset --mixed 0d588ed

HISTORY

f1.txt
v1

STAGED

f1.txt
v1 v2 v3

CARPETA LOCAL

f1.txt
v1 v2 v3

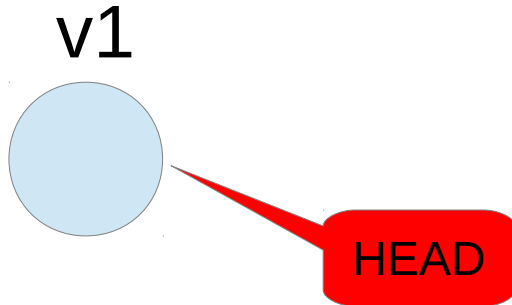
f2.txt
v2 v3

f3.txt
v3

git reset --mixed

- `git log --pretty="format:%h %s"`

0d588ed v1



git reset --mixed

- **git status**

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: f1.txt

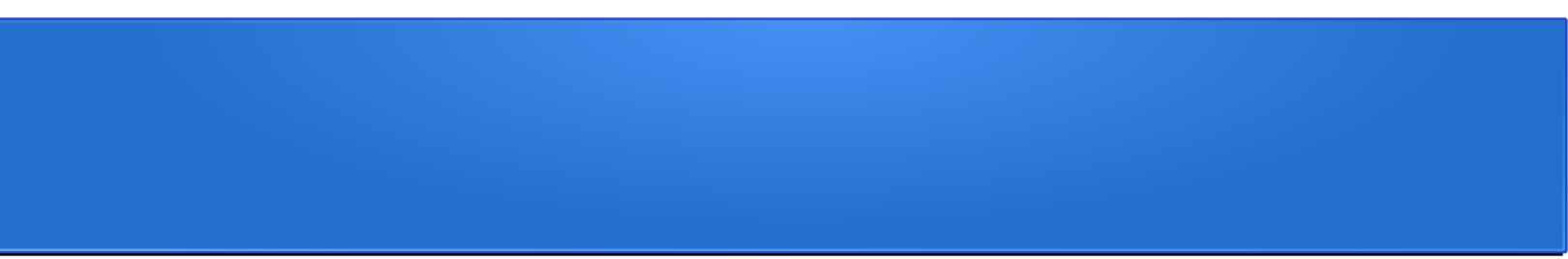
Untracked files:

(use "git add <file>..." to include in what will be committed)

f2.txt

f3.txt

no changes added to commit (use "git add" and/or "git commit -a")



```
git checkout <commit>
```

git checkout commit

- git checkout 0d588ed

HISTORY

f1.txt
v1

STAGED

f1.txt
v1

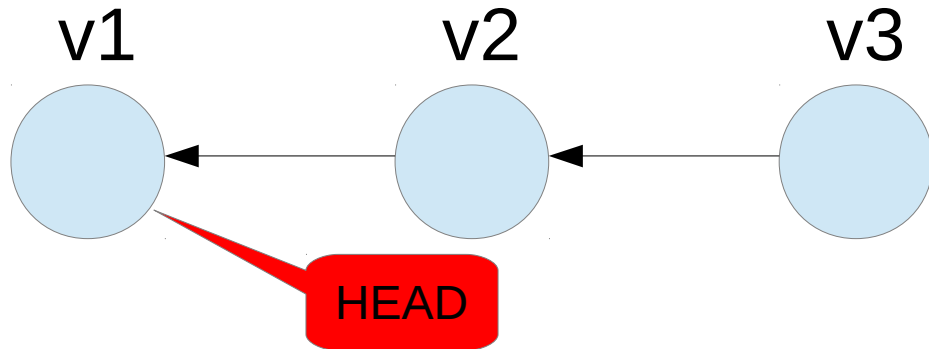
CARPETA LOCAL

f1.txt
v1

git checkout commit

- `git log --pretty="format:%h %s"`

0d588ed v1



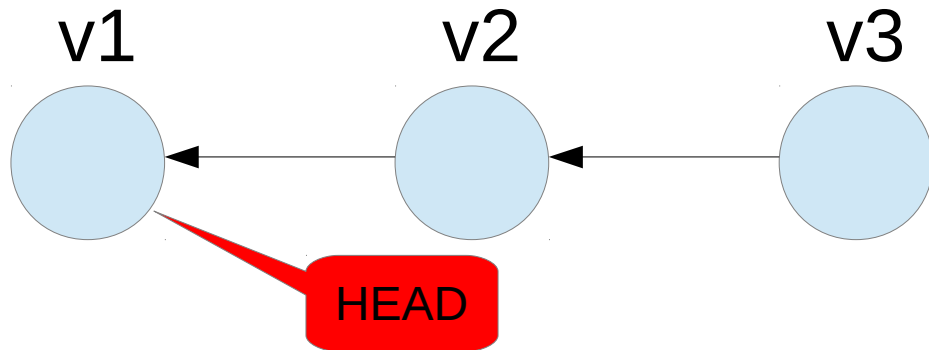
git checkout commit

- `git log --pretty="format:%h %s" --all`

5687d92 v3

a3aeb2a v2

0d588ed v1



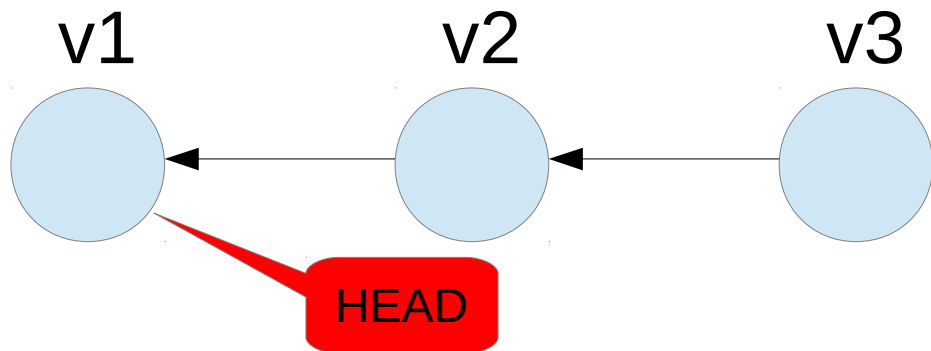
git checkout commit

- `git log --oneline --all`

5687d92 (master) v3

a3aeb2a v2

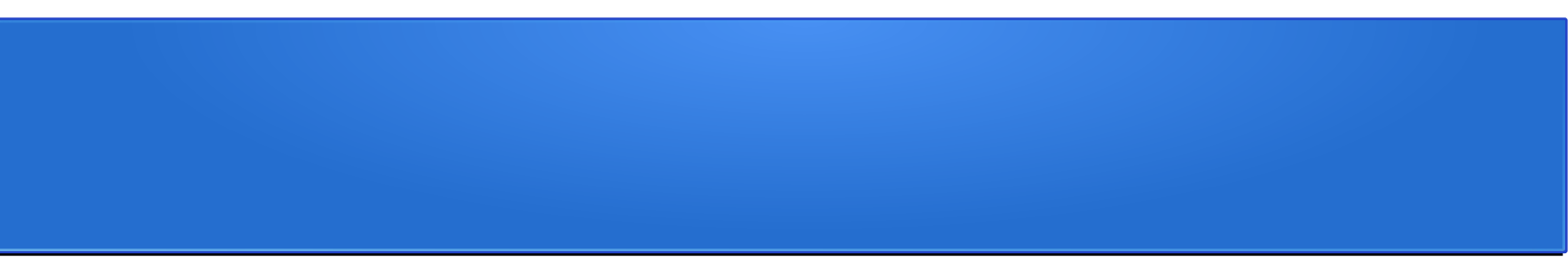
0d588ed (HEAD) v1



git checkout commit

- git status

```
HEAD detached at 0d588ed  
nothing to commit, working tree clean
```



```
git checkout <commit>  
          <file>
```

git checkout commit file

- git checkout 0d588ed **f1.txt**

HISTORY

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

CARPETA LOCAL

f1.txt
v1

f2.txt
v2 v3

f3.txt
v3

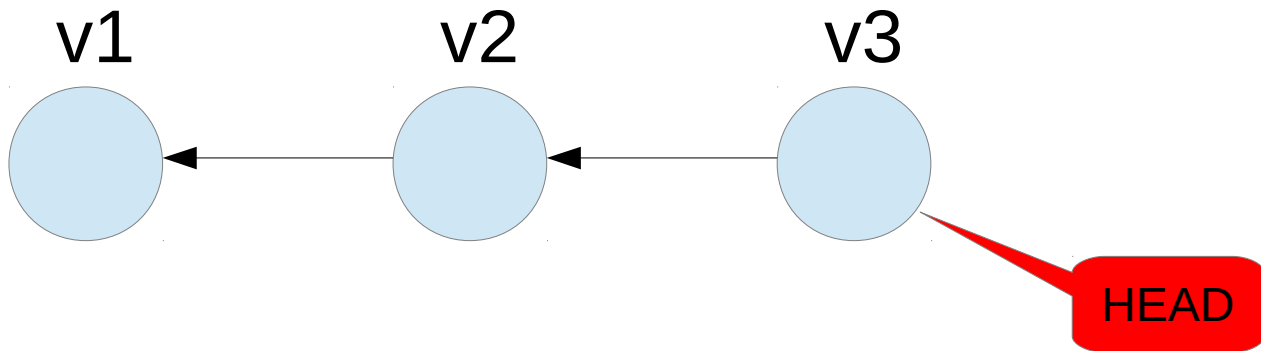
git checkout commit file

- `git log --pretty="format:%h %s"`

5687d92 v3

a3aeb2a v2

0d588ed v1



git checkout commit file

- git status

On branch master

Changes to be committed:

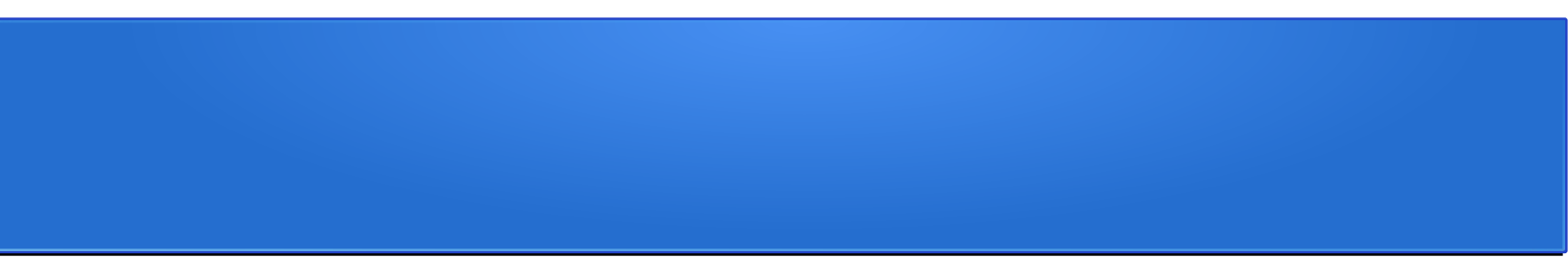
(use "git reset HEAD <file>..." to unstage)

modified: f1.txt

git checkout commit file

- `git checkout HEAD f1.txt`
`git checkout f1.txt`

Deshace los cambios en la carpeta local que hayas hecho desde el último commit del fichero f1.txt



```
git reset <commit>  
      <file>
```


git reset commit file

- git reset 0d588ed **f3.txt**

HISTORY

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3

CARPETA LOCAL

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

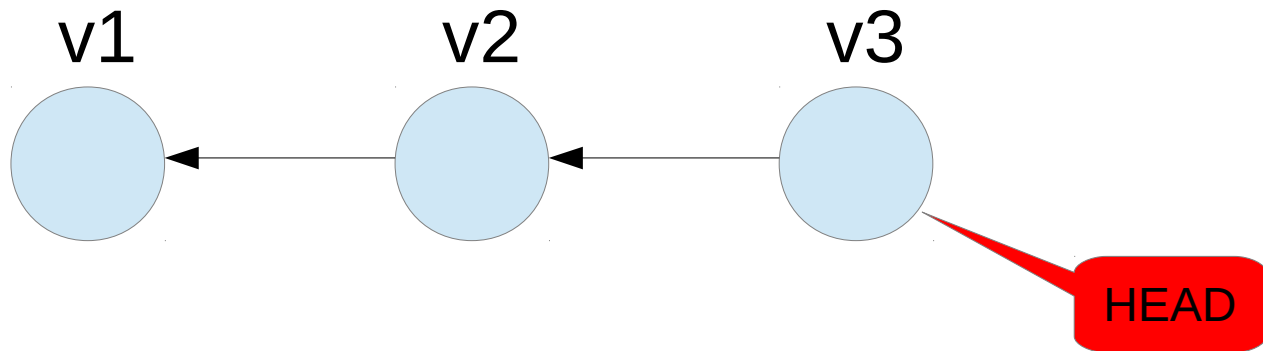
git reset commit file

- `git log --pretty="format:%h %s"`

5687d92 v3

a3aeb2a v2

0d588ed v1



git reset commit file

- git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

deleted: f3.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

f3.txt

git reset commit file

- `git reset HEAD f4.txt`
`git reset f4.txt`

Modifica el STAGED para que se quede como estaba en el último commit, pero no modifica la carpeta local.

Realmente es como deshacer la orden “`git add f4.txt`”

No es útil si el fichero ya existe en el commit



git reset vs rm

git reset commit file

- git reset HEAD **f3.txt**

HISTORY

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3

CARPETA LOCAL

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

git reset commit file

- git status

On branch master
nothing to commit, working tree clean

git rm file

- `git rm --cached f3.txt`

HISTORY

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

STAGED

f1.txt
v1 v2 v3

f2.txt
v2 v3



CARPETA LOCAL

f1.txt
v1 v2 v3

f2.txt
v2 v3

f3.txt
v3

git rm file

- git status

On branch master

Changes to be committed:

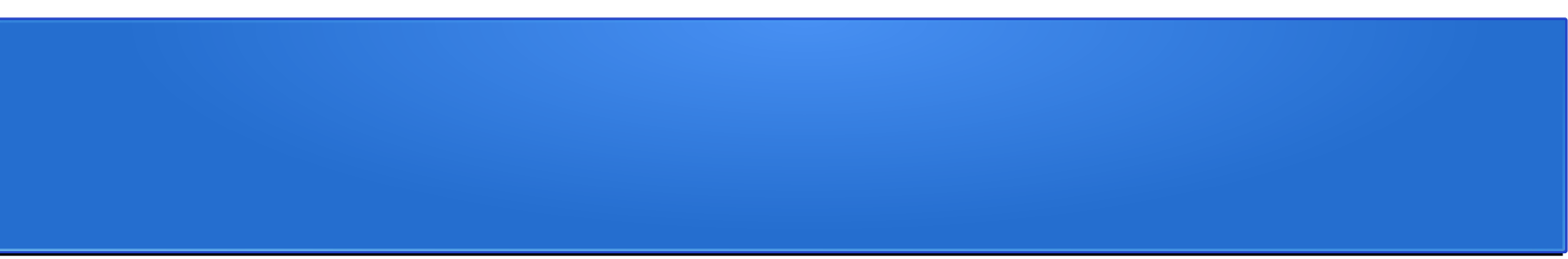
(use "git reset HEAD <file>..." to unstage)

deleted: f3.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

f3.txt



```
git revert <commit>
```

git revert commit

- `git revert 5687d92` #Revertir v3 y volver a v2

HISTORY

f1.txt
v1 v2

f2.txt
v2

STAGED

f1.txt
v1 v2

f2.txt
v2

CARPETA LOCAL

f1.txt
v1 v2

f2.txt
v2

git revert commit

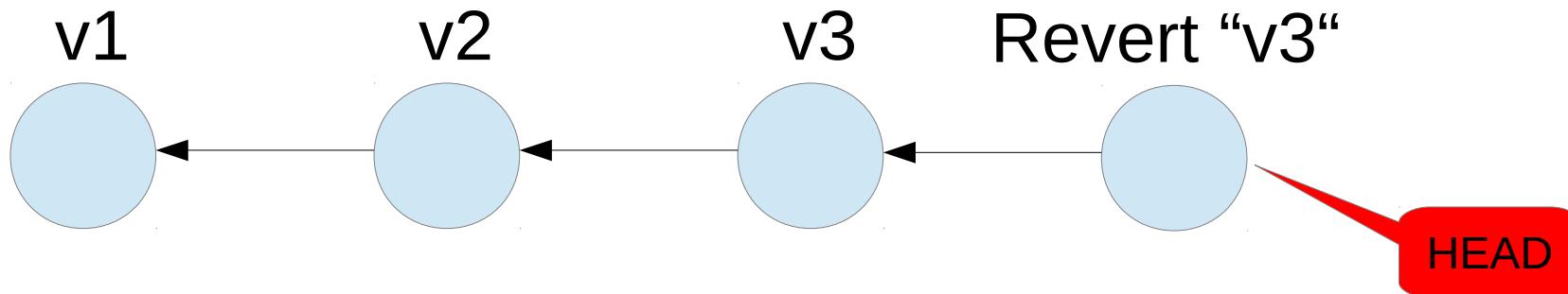
- `git log --pretty="format:%h %s"`

a1d5431 Revert "v3"

5687d92 v3

a3aeb2a v2

0d588ed v1



git revert commit

- `git status`

On branch master
nothing to commit, working tree clean

git revert commit

```
git revert 5687d92
```

```
#Revertir v3
```

```
git revert a3aeb2a
```

```
#Revertir v2
```

- Si queremos volver a un commit anterior (Por ejemplo:v1), hay que hacer varios “revert” seguidos y revertir cada uno de los commits.
- Si NO deshacemos el último commit se pueden producir conflictos ya que no se deshace **hasta** ese commit sino que se deshace **únicamente** ese commit.