GIT: DESHACER CAMBIOS

- Recuperar un fichero del área de commit al working directory.

Con checkout si queremos recuperar un fichero siempre será del área de commit al working directory, nunca se pasa por el área de preparados.

Crearse un nuevo directorio, vincularlo a git y creamos tres ficheros, más un readme.md

echo “version1 – fichero1” > fichero1.txt

echo “version1 – fichero2” > fichero2.txt

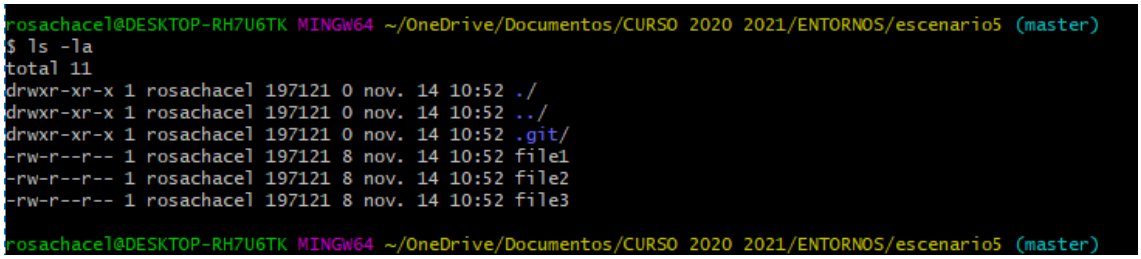
echo “version1 – fichero3” > fichero3.txt

echo “version1 – fichero4” > fichero4.txt

git add .

git comit -m “IC”

git satus



git checkout admite dos parámetros la rama y un fichero, para distinguirlos se antepone dos guiones “— “ para indicar que vamos a hacer checkout de fichero, es decir:

git checkout master (Rama) (Es para cambiar de rama lo veremos más adelante).

git checkout – master (fichero o grupo de fichero -- .)

También me puedo traer el fichero de una versión concreta con

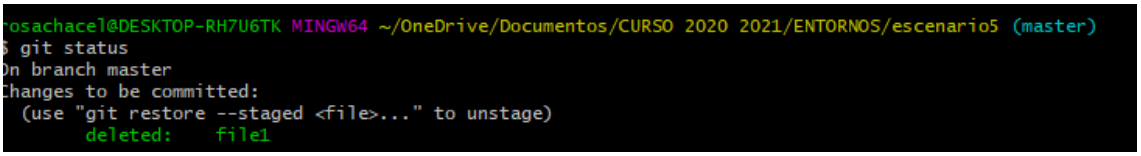
git checkout HEAD fichero (puede ser la etiqueta como HEAD el HASHID)

Puedo traer versiones anteriores a la actual con HEAD~1

git checkout HEAD~1 fichero

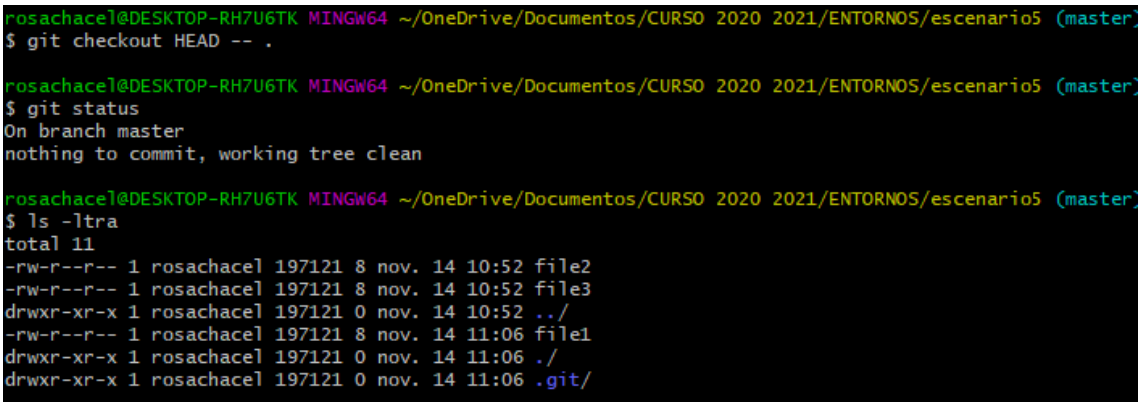
Vamos a ver un ejemplo, de como recuperarnos de un problema:

git rm file1



Como lo recupero

git checkout HEAD -- .



Vemos que tengo file1, file2 y file3.

- Recuperar un fichero del Área de Staging

  echo “añado una nueva línea al final” >> fichero1.txt

  git add fichero1.txt

  git status

```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   fichero1.txt
```

  git restore –staged fichero1.txt

  git status

```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   fichero1.txt

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

  git reset HEAD fichero1.txt

- Resetear todo desde el área de commit al Working Directory borrando todo lo que está en el área de staging.

  echo “cabecera” > file1

  echo “cabecera” > file2

  echo “cabecera” > file3

  git add .

  git commit -m “cabecera”

  echo “ultima linea” > file1

  echo “ultima linea” > file2

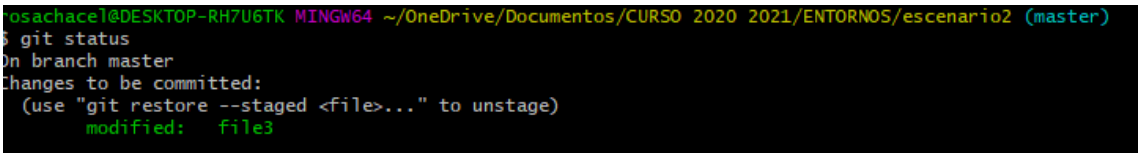
  echo “ultima linea” > file3

  git add .

  git commit -m “ultima línea”

  git “cambio” >> file3g

  git add file3



  git reset –hard HEAD~1

  ¿Qué obtendré?

Desparece el cambio del área de preparado y el contenido en el Working Directory de los ficheros.

```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git reset --hard HEAD~1
HEAD is now at 5dcc0e4 cabecera
```

git status

```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git status
On branch master
nothing to commit, working tree clean

rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ cat file1
cabecera

rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
```

git log –oneline

```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git log --oneline
5dcc0e4 (HEAD -> master) cabecera
9a59f6d version 2
cc7eac6 version 1
```

Se ha borrado todo del área de preparado y del Working directory.

He perdido la versión con la etiqueta última línea

!!CUIDADO CON ESTE COMANDO!!!!

- Realizar el cambio pero sin perder los cambios ->Revertir.

  echo “ultima línea equivalente” > file1

  echo “ultima línea equivalente” > file2

  echo “ultima línea equivalente” > file3

  git add .

  git commit -m “he incluido una última línea en todos mis ficheros”

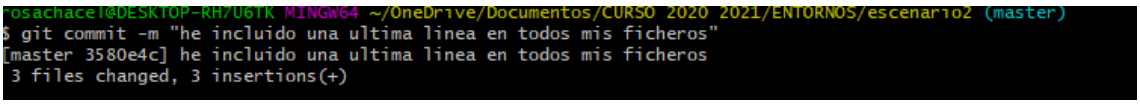
```
rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git status
On branch master
nothing to commit, working tree clean

rosachacel@DESKTOP-RH7U6TK MINGW64 ~/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2 (master)
$ git log --oneline
3580e4c (HEAD -> master) he incluido una ultima linea en todos mis ficheros
5dcc0e4 cabecera
9a59f6d version 2
cc7eac6 version 1
```

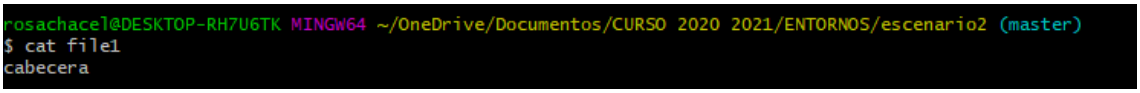
  No tengo nada que aprobar y tengo una nueva revisión.

  git revert 3580e4c

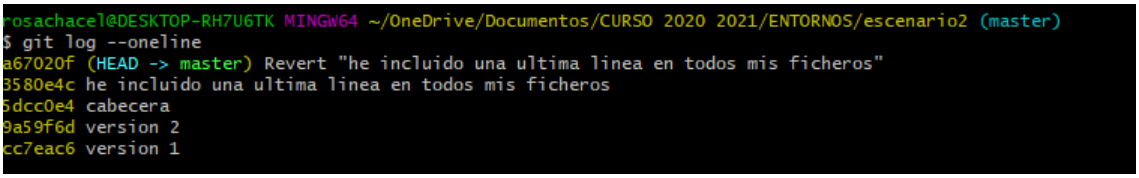
```
MINGW64:/c/Users/david/OneDrive/Documentos/CURSO 2020 2021/ENTORNOS/escenario2
  GNU nano 4.9.3            C:/Users/david/OneDrive/Documentos/CURSO 2020 2021/ENTORNO
Revert "he incluido una ultima linea en todos mis ficheros"

This reverts commit 3580e4c59126374a7447291753a1901a5407004c.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   file1
#       modified:   file2
#       modified:   file3
#
```

Me dice que las líneas que empiezan con # serán ignoradas, pero como yo quiero revertir los cambios las quito.





No he perdido los cambios, los tengo en una versión anterior.



Podía haber revertido solo un fichero (no quitando todos los #).

Por último, se pueden revertir todos los cambios sobre una franja dada de versiones, con el siguiente comando:

git revert HEAD. . .HEAD~2 –no-edit

el no-edit es para que no pregunte que quiero cambiar.