GIT: MÁS COMANDOS

- Ignorar ficheros del control de versiones.

  Creamos 1 fichero

  echo “fichero 1” > fich1.tmp

  git status

  Se crea un fichero .gitignore

  y se le añade una máscara

  *.tmp

  No se agregará nunca al área de staging.

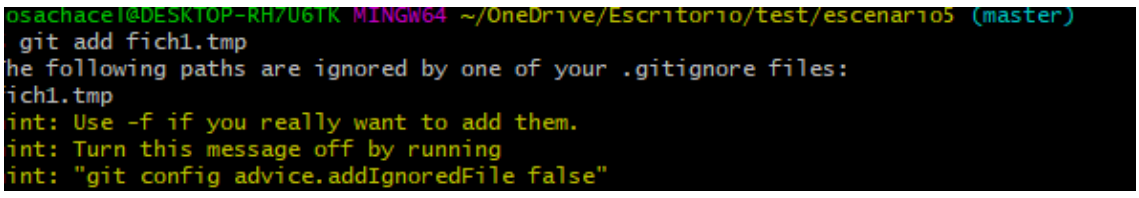
  git status

  Incluyo todos los ficheros del Working directory para pasarlos al área de Staging o preparados.

  git add .

  git status

  Se hago git add fich1.tmp da el siguiente error:



Para añadir .gitignore lo tengo que hacer con

git add .gitignore.

- Ver los cambios de un fichero modificado

Me creo un nuevo fichero readme2.md y modifico readme.md.

Ahora tendré dos advertencias en rojo.

Subo todo al área de Staging

git add .

git commit -m “revisión 3”

Ahora modifico el fichero readme.md

Hacemos git status y nos dice que hay un cambio y para ver en que difiere hacemos

git diff

Hace la diferencia entre el Working Directory y el repositorio remoto.

git difftool

me aparece vimdiff. Pero vamos a instarlar tkdiff

Copiar tkdiff en la siguiente ruta de git

C:\Program Files\Git\usr\bin

git difftool

ahora nos reconoce tkdiff

- Borrado de ficheros

Borrado de fichero

rm readme.md

git status

la forma de hacerlo debería ser

git rm readme.md

Lo borra y lo pasa directamente al área de staging.

- Movimiento de ficheros

Lo mismo para mv.

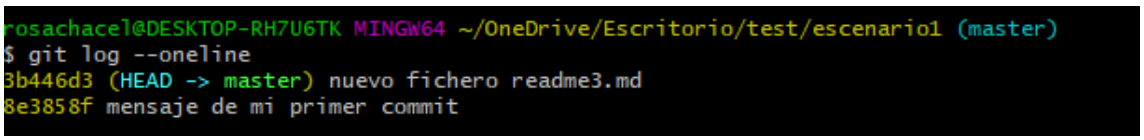
Hace dos cosas git considera que se ha borrado un fichero y se ha creado uno nuevo, luego hay que hacer un git add . para guardar los dos cambios. Lo mejor es hacer los siguiente

git mv fich1 fich2

- Mas opciones de git log

--oneline

Devuelve el identificador de la versión y la etiqueta.



el id de la versión es un hash, por eso se suele llamar también “hash id”.

HEAD es la etiqueta de la revisión más actual, se va a utilizar para hacer búsquedas: desde HEAD tres versiones atrás, etc.

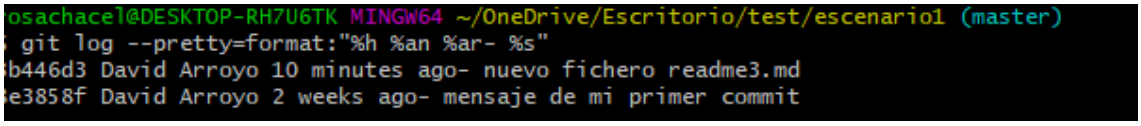
git log –pretty=”%h %an %ar %s”

h -> hashid

an -> autor name

ar -> de cuando es el cambio

s -> nombre de la revisión



NOTA: format se puede omitir.

- Chequear revisiones

git show hashid

Muestra los cambios de una versión respecto a la anterior

Paso de no tener nada /dev/null a tener un fichero.

Para ver las diferencias entre un fichero en el área de staging y el repositorio local

git diff –staged