

Usar git diff desde la consola siempre ha sido una operación muy compacta y a veces los mensajes son algo crípticos . Vamos a explicar este comando a detalle para poder ver como funciona. Imaginemonos que tenemos el fichero3.txt

```
hola1  
hola2  
hola3  
hola4  
hola5  
hola6
```

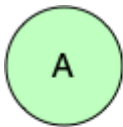
se trata de un fichero de texto muy sencillo con 6 mensajes de hola. Vamos a modificar un mensaje de hola6 a adios .

```
hola1  
hola2  
hola3  
hola4  
hola5  
adios
```

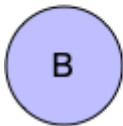
Solicitamos un git diff:

```
diff --git a/fichero3.txt b/fichero3.txt
index 9b5b79e..dd74324 100644
--- a/fichero3.txt
+++ b/fichero3.txt
@@ -3,4 +3,4 @@ hola2
 hola3
 hola4
 hola5
-hola6
\ No newline at end of file
+adios
```

Lo primero que vemos es que al fichero viejo le denomina a y al fichero nuevo b



Viejo



Nuevo

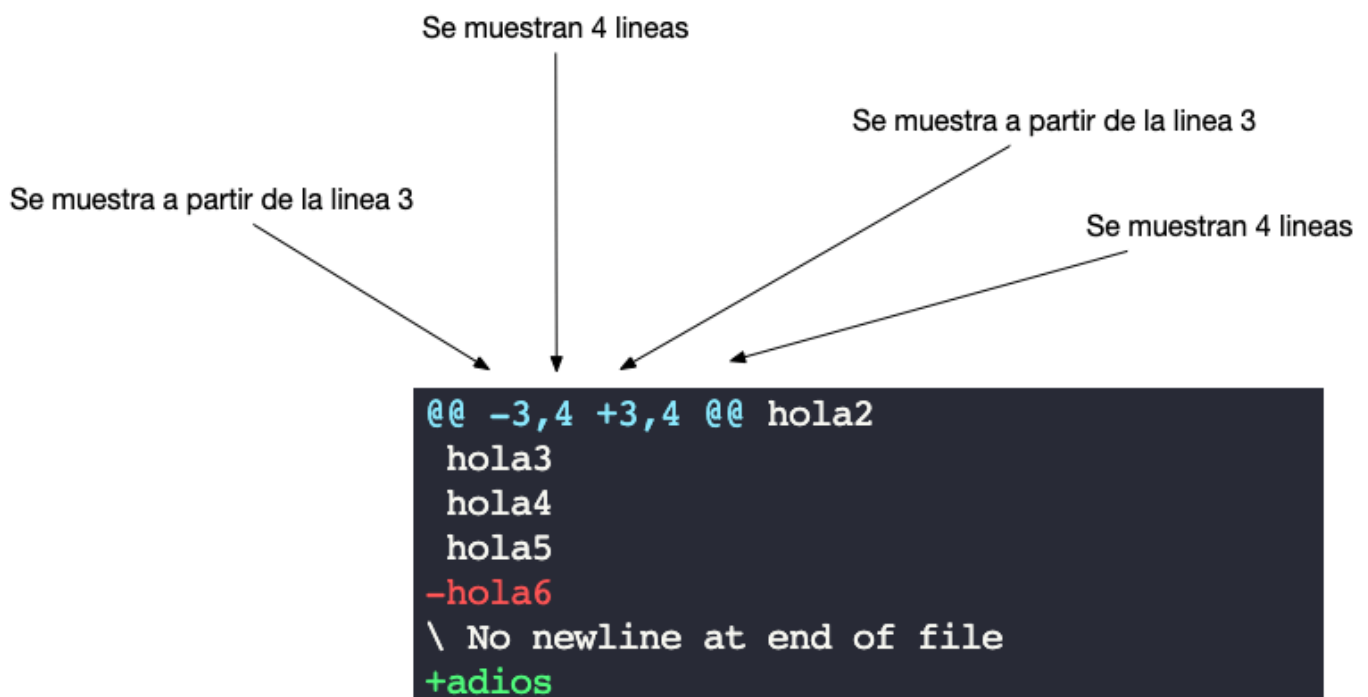
Una vez tenemos esto muy claro lo siguiente que nos dice es que los simbolos de - estan ligados al fichero viejo y los simbolos de + estan ligados al fichero nuevo.

```
--- a/fichero3.txt
+++ b/fichero3.txt
```

Esto es algo también importante de entender . Es momento de empezar a revisar las diferencias . En este caso las diferencias se dibujan así:

```
@@ -3,4 +3,4 @@ hola2
 hola3
 hola4
 hola5
-hola6
\ No newline at end of file
+adios
```

Esto quiere decir que el viejo fichero tiene una línea afectada que es la línea del hola6 y el fichero nuevo pues tiene una nueva línea que se denomina adios. Hasta aquí todo muy claro . Ahora en lo que tenemos mas dudas es en los numeros -3,4 +3,4 .



En este caso nos dice que es a partir de la línea 3 que estamos mostrando y que a partir de esa línea se muestran 4 líneas adicionales. De esta manera podemos entender que es la línea del hola 6 la que cambia por adios ya que la línea hola6 contiene un menos (-) que nos dice que es del fichero viejo y la línea adios contiene un (+) del nuevo fichero.

Vamos a ver una segunda versión parecida del fichero partiendo de esta situación inicial:

hola1  
hola2  
hola3  
hola4  
hola5  
hola6  
hola7  
hola8  
hola9  
hola10  
hola11

Pero teniendo esta situación final:

adios  
hola2  
hola3  
hola4  
hola5  
hola6  
hola7  
hola8  
hola9  
adios  
hola11

```
diff --git a/fichero3.txt b/fichero3.txt
index 9668e3b..6d7ecd1 100644
--- a/fichero3.txt
+++ b/fichero3.txt
@@ -1,4 +1,4 @@
-hola1
+adios
 hola2
 hola3
 hola4
@@ -7,5 +7,5 @@ hola6
 hola7
 hola8
 hola9
-hola10
+adios
 hola11
```

En este caso nos dice que desde la línea 1 se muestran 4 líneas con sus diferencias tanto del fichero viejo como del nuevo y vemos como un hola1 esta en el viejo y un adios esta en el nuevo. Una vez hecho esto git se da cuenta que existe otro bloque de diferencias que empieza en la línea del hola7 y que nos muestra 5 líneas con los holas del fichero viejo y el adios del nuevo.

Es de esta forma con la cual git trabaja con sus bloques de diferencias o hunks

- [Flux vs Mono ,Spring y la programación reactiva](#)
- [Eclipse Git , Repositorios locales y remotos](#)
- [Java Stream File y manejo de ficheros](#)
- [Git Alias y como usarlos](#)
- [Eclipse GIT commit y como crear un repositorio](#)
- [git](#)