

GIT REFLOG

Git lleva seguimiento de cambios a la punta de las ramas usando el mecanismo reflog. Esto volver a los grupos de cambios incluso si no están referidos por ninguna rama o etiqueta. Después de reescribir la historia, reflog contiene la información del estado antiguo de ramas y permite volver a ese estado si fuera necesario.

Uso

```
git reflog
```

Muestra el reflog para el repositorio local.

```
git reflog --relative-date
```

Muestra reflog con la información relativa de la cuenta (p.ej. hace dos semanas).

Explicación

Cada vez que el HEAD actual se actualiza (al cambiar ramas, al recibir cambios, reescribir historia o simplemente añadir nuevos commits) se añadirá una nueva entrada al reflog.

Ejemplo

Miremos un ejemplo para entender git reflog.

```
0a2e358 HEAD@{0}: reset: moving to HEAD~2
0254ea7 HEAD@{1}: checkout: moving from 2.2 to master
c10f740 HEAD@{2}: checkout: moving from master to 2.2
```

El reflog anterior muestra un checkout de la rama master a la rama 2.2 y vuelta atrás. Desde ahí, reinicializa (hard reset) a un commit antiguo. La última actividad se presenta al principio del HEAD@{0} etiquetado.

Si accidentalmente has vuelto atrás, el reflog tendrá el commit master apuntando a (0254ea7) antes de que descartaras accidentalmente dos commits.

```
git reset --hard 0254ea7
```

Ó

```
git reset --hard HEAD@{1}
```

Al usar git reset es posible cambiar master al commit que era antes. Esto proporciona una red de seguridad en caso de que se cambie la historia de forma accidental. Es importante tener en cuenta que reflog solo proporciona una red de seguridad si los cambios se han confirmado al repositorio local, y solo hace seguimiento de movimientos.