

Después de haber creado el diario.txt le agregamos la siguiente información con cada línea agregando un commit respectivo

```
1  Día 1: Explorando Git
2  Día 2: Aprendiendo sobre commits
3  Día 3: Descubriendo ramas
```

```
Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
● $ git log
commit fa00e31d226a956e00de76c19e2bf4ca9309aba5 (HEAD -> main)
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:56 2025 +0000

    Third Commit: Branchs

commit 8191eb0a2ace1d4221408d9589a531157208fea2
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:27 2025 +0000

    Second Commit: Learning Commits

commit f86bc584c65de2b50c8c31a01161273250b6b17e
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:10:53 2025 +0000

    First Commit: Exploring Git
```

Utilizamos git checkout con el hash del primer commit

```
Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
● $ git checkout f86bc58
Note: switching to 'f86bc58'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at f86bc58 First Commit: Exploring Git
```

Quedando el diario.txt de esta forma al volver a un commit anterior.

```
Git > Tareas > Tarea1 > ≡ diario.txt
1   Día 1: Explorando Git
```

Y ahora para volver a la actualidad utilizamos lo siguiente

```
Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind ((f86bc58...))
● $ git checkout main
Previous HEAD position was f86bc58 First Commit: Exploring Git
Switched to branch 'main'
Your branch and 'origin/main' have diverged,
and have 3 and 1 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)
```

Volviendo a tener el diario.txt de antes.

Despues añadimos una entrada mas al diario y le hacemos el respectivo add y commit.

```
1   Día 1: Explorando Git
2   Día 2: Aprendiendo sobre commits
3   Día 3: Descubriendo ramas
4   Día 4: Retrocediendo en el tiempo con checkout
```

```
Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
● $ git add .

Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
● $ git commit -m "Fourth Commit: Time Change"
[main 8766ac6] Fourth Commit: Time Change
```

Quedandose el log de commits tal que asi

```

$ git log
commit 8766ac60c365784c0a223a8240f71d2d9a0b8a35 (HEAD -> main)
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:17:45 2025 +0000

    Fourth Commit: Time Change

commit fa00e31d226a956e00de76c19e2bf4ca9309aba5
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:56 2025 +0000

    Third Commit: Branchs

commit 8191eb0a2ace1d4221408d9589a531157208fea2
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:27 2025 +0000

    Second Commit: Learning Commits

commit f86bc584c65de2b50c8c31a01161273250b6b17e
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:10:53 2025 +0000

    First Commit: Exploring Git

```

Ahora para “Retroceder en el tiempo” haremos lo siguiente.

```

Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
$ git reset --hard fa00e3

```

Quedando el mismo diario que el principio y teniendo el log tal que así.

```

1 Día 1: Explorando Git
2 Día 2: Aprendiendo sobre commits
3 Día 3: Descubriendo ramas

```

```
Cesar@DESKTOP-QINDB7V MINGW64 /e/Apuntes MasterMind (main)
● $ git log
commit fa00e31d226a956e00de76c19e2bf4ca9309aba5 (HEAD -> main)
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:56 2025 +0000

    Third Commit: Branchs

commit 8191eb0a2ace1d4221408d9589a531157208fea2
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:11:27 2025 +0000

    Second Commit: Learning Commits

commit f86bc584c65de2b50c8c31a01161273250b6b17e
Author: Cesardrom <cesardguezr20@gmail.com>
Date: Sat Jan 25 17:10:53 2025 +0000

    First Commit: Exploring Git
```

¿Qué sucedió cuando usaste git checkout para ir a un commit anterior? ¿Pudiste ver los cambios realizados hasta ese punto?

Lo que sucede es que los archivos volvieron al estado que tenían en el commit del principio siendo así capaz de modificar desde ese punto si quería o tener esa información aislada.

Cuando deshiciste el último commit, ¿qué pasó con los cambios en diario.txt? ¿Y con la historia de tus commits?

Se eliminaron los cambios pero porque al ponerle en el comando el --hard antes del hash del commit hace que se borre de forma permanente en git los cambios realizados después del commit al que te acabas de movilizar. Tambien se podria hacer el desplazamiento sin eliminar solo con el reset que eliminaría el commit del log pero no la información del Working Tree