

Práctica 04

Contenido

1	Creando alias para nuestros comandos.....	2
2	Cambios en los archivos.....	3
2.1	Comparar cambios en archivos con visual studio code	5
3	Actualizar mensajes de commits y deshacer commits	5
3.1	Actualizar el mensaje de un commit	5
3.2	Borrar un commit	6
4	Preparando un repositorio para viajes en el tiempo	7

1 Creando alias para nuestros comandos

Para mostrar el estado de los archivos del repositorio con una descripción corta podemos usar el siguiente comando:

```
git status
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git status
On branch main
nothing to commit, working tree clean
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git status --short
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> █
```

Podemos crear un alias para este comando de forma que podamos ejecutarlo escribiendo menos texto:

```
git status -short
git config --global alias.s "status -short"
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git status
On branch main
nothing to commit, working tree clean
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git status --short
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> █
```

```
PS C:\git\cristina\02-bases> git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        fonts/
        images/
        js/.DS_Store
        scss/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\git\cristina\02-bases> git status --short
?? .DS_Store
?? fonts/
?? images/
?? js/.DS_Store
?? scss/
PS C:\git\cristina\02-bases> █
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git config --global alias.s "status --short"
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> git s
PS C:\NIEVES\DAW\ENDES\Práctica GIT\02-bases> █
```

Como vemos para crear un alias usamos la configuración global y ponemos alias. y después del punto ponemos el texto que servirá de alias. A continuación, escribimos entre comillas el comando que queremos abreviar sin poner git delante.

2 Cambios en los archivos

Creamos un nuevo repositorio que se llame 03-instalaciones y lo abrimos en visual studio.

Hacemos el **git init** para inicializar el repositorio y creamos un archivo que se llame instalaciones.md con el siguiente contenido:

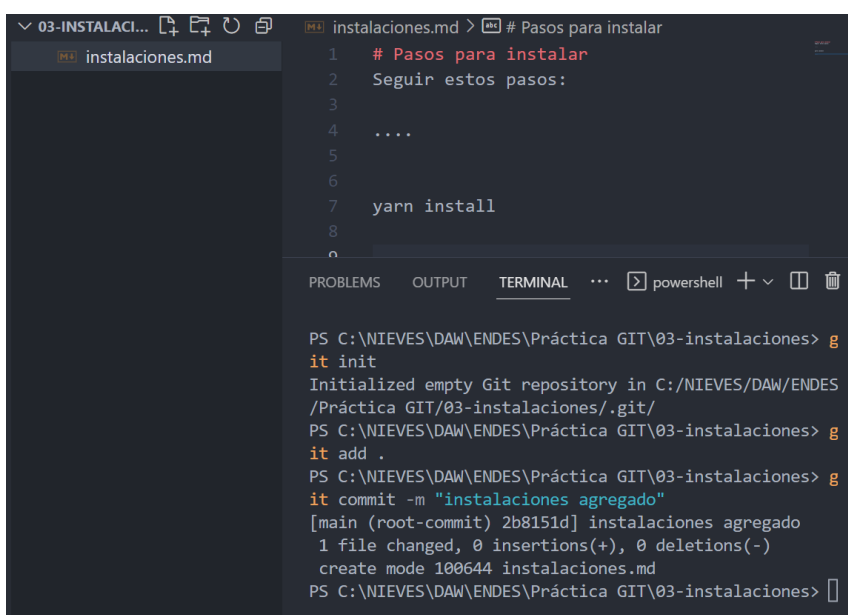
Lo agregamos al stage:

```
git config --global alias.s "status --short"
```

Y hacemos commit:

```
git commit -m "instalaciones agregado"
```

Deberíamos ver algo así en nuestro visual studio code:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a file named 'instalaciones.md'. The main editor area displays the content of this file, which includes a heading '# Pasos para instalar', a subheading 'Seguir estos pasos:', and a code block for 'yarn install'. At the bottom, the TERMINAL pane shows the following commands and output:

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git init
Initialized empty Git repository in C:/NIEVES/DAW/ENDES/Práctica GIT/03-instalaciones/.git/
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git add .
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git commit -m "instalaciones agregado"
[main (root-commit) 2b8151d] instalaciones agregado
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 instalaciones.md
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> █
```

Modificamos el archivo instalaciones y lo dejamos así:

(Hemos modificado una línea y añadido 3 líneas vacías debajo de la segunda línea con ...)

Guardamos los cambios, pero no hacemos add ni commit.

Vamos a comparar las modificaciones realizadas en este archivo, para ellos usamos el siguiente comando:

git diff

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> g
it diff
diff --git a/instalaciones.md b/instalaciones.md
index e69de29..c143a8a 100644
--- a/instalaciones.md
+++ b/instalaciones.md
@@ -0,0 +1,9 @@
+# Pasos para instalar
+
+Seguir estos pasos:
+
+....git
+
+yarn install
+
+....
```

La versión a del archivo instalaciones nos dice que tiene menos cosas que la versión b del mismo archivo

```
PS D:\Git\03-instalaciones> git diff
diff --git a/instalaciones.md b/instalaciones.md
index 4920948..d7266ac 100644
--- a/instalaciones.md
+++ b/instalaciones.md
@@ -3,5 +3,7 @@ Seguir estos pasos:
...
-npm install
...
\ No newline at end of file
+yarn install
+.
```

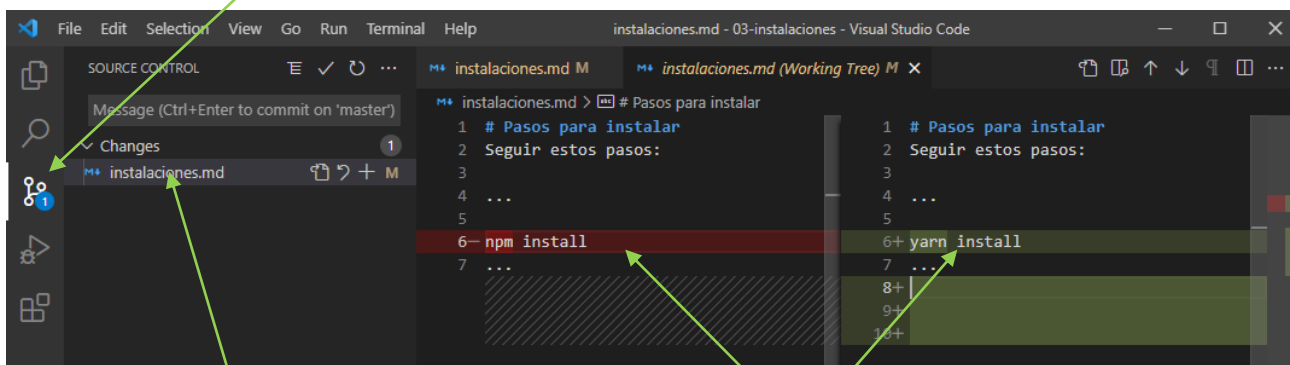
Las líneas en color rojo y precedidas por un - nos indican cosas que la primera versión no tiene respecto al segundo y las cosas en verde precedidas por un + nos indican cosas que la segunda versión tiene y la primera no.

El comando `git diff` nos compara las modificaciones de los archivos que no están en el escenario. Si queremos usar el comando con un archivo que hemos añadido al escenario usaremos lo siguiente:

```
git diff --staged
```

2.1 Comparar cambios en archivos con visual studio code

Desde la opción de source control de visual studio podemos comparar las modificaciones realizadas en un archivo de formas más visual:



Hacemos click en el archivo del cual queremos ver las diferencias

3 Actualizar mensajes de commits y deshacer commits

3.1 Actualizar el mensaje de un commit

Para cambiar el mensaje del último commit realizado usaremos el siguiente comando:

```
git commit --amend -m "nuevo texto para mi commit"
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git commit --amend -m "nuevo
texto para mi commit"
[main bf2c8a5] nuevo texto para mi commit
Date: Thu Jan 12 13:09:44 2023 +0100
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 instalaciones.md
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git log
commit bf2c8a5005dc972b93dba9a3424d73b4a62456f4 (HEAD -> main)
Author: Nieves <sevein_cd@hotmail.com>
Date: Thu Jan 12 13:09:44 2023 +0100

    nuevo texto para mi commit
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> █
```

3.2 Borrar un commit

Modificamos el archivo instalaciones.md (escribimos cualquier cosa) y hacemos commit:

```
git commit -am "segundo commit"
ó bien
git add instalaciones.md
git commit -m "segundo commit"
```

Para borrar el último commit realizado usaremos el siguiente comando:

```
git reset --soft HEAD^
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git commit -am "segundo commit"
[main d2380c1] segundo commit
1 file changed, 10 insertions(+)
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git log
commit d2380c1a9d1db7e2eb310294b18f5dc885920438 (HEAD -> main)
Author: Nieves <sevein_cd@hotmail.com>
Date: Thu Jan 12 13:20:25 2023 +0100

    segundo commit

commit bf2c8a5005dc972b93dba9a3424d73b4a62456f4
Author: Nieves <sevein_cd@hotmail.com>
Date: Thu Jan 12 13:09:44 2023 +0100

    nuevo texto para mi commit
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git reset --soft HEAD^
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> █
```

Esto eliminará el commit y nos dejará el repositorio como estaba antes de hacer dicho commit. Es decir si teníamos archivos modificados volverán a aparecer como modificados. El mantener los cambios se debe al parámetro --soft.

(*--hard para no mantener los cambios)

```
1 file changed, 10 insertions(+)
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git log
commit 2d15ee6ec79691e56b95feff16e1dca4f57eb498 (HEAD -> main)
Author: Nieves <sevein_cd@hotmail.com>
Date: Thu Jan 12 13:23:53 2023 +0100

    commit para borrar

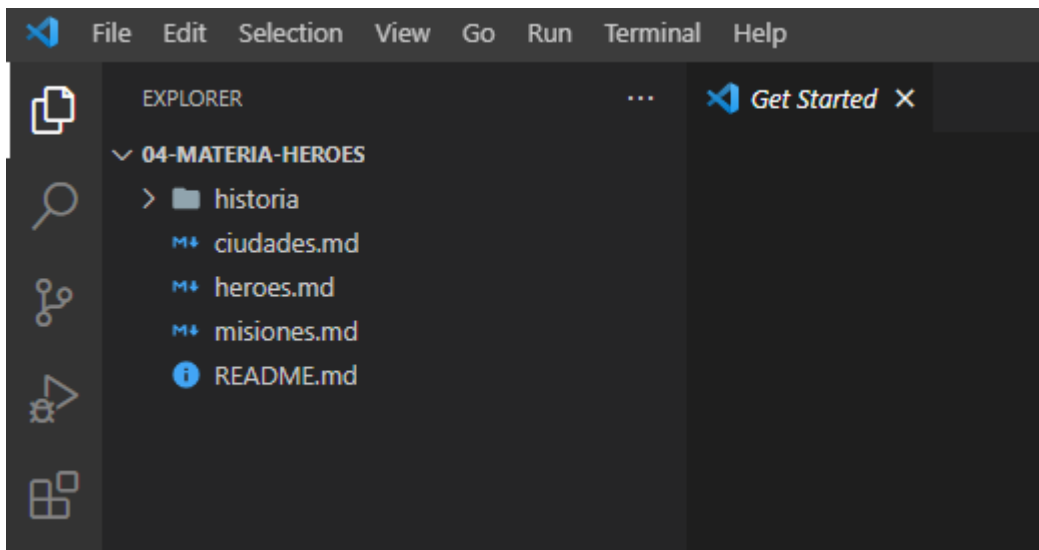
commit bf2c8a5005dc972b93dba9a3424d73b4a62456f4
Author: Nieves <sevein_cd@hotmail.com>
Date: Thu Jan 12 13:09:44 2023 +0100

    nuevo texto para mi commit
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git reset --hard HEAD
HEAD is now at 2d15ee6 commit para borrar
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> git reset --hard HEAD^
HEAD is now at bf2c8a5 nuevo texto para mi commit
PS C:\NIEVES\DAW\ENDES\Práctica GIT\03-instalaciones> 
```

4 Preparando un repositorio para viajes en el tiempo

Descomprimos el archivo Materia-Heroes y colocamos la carpeta Materia-Heroes en la carpeta donde estemos creando nuestros repositorios.

Renombramos la carpeta como: 04-Materia-Heroes y arrastramos la carpeta a visual estudio code. Deberíamos tener algo así:



Vamos a ir ejecutando los siguientes comandos:

Inicializamos el repositorio:

```
git init
```

Añadimos el archivo README.md

```
git add README.md
```

Hacemos el commit de este archivo:

```
git commit -m "readme.md agregado"
```

Añadimos el archivo misiones.md

```
git add misiones.md
```

Hacemos el commit de este archivo:

```
git commit -m "misiones.md agregado"
```

Añadimos el archivo heroes.md

```
git add heroes.md
```

Hacemos el commit de este archivo:

```
git commit -m "heroes.md agregado"
```

Añadimos el archivo ciudades.md

```
git add ciudades.md
```



```
git commit -m "ciudades.md agregado"
```

```
git add historia\
```

```
git commit -m "historia agregado"
```

se abre un editor de texto – pulso la i para insertar – ESC para salir - :wq (guardar y salir)

```
# historia agregado

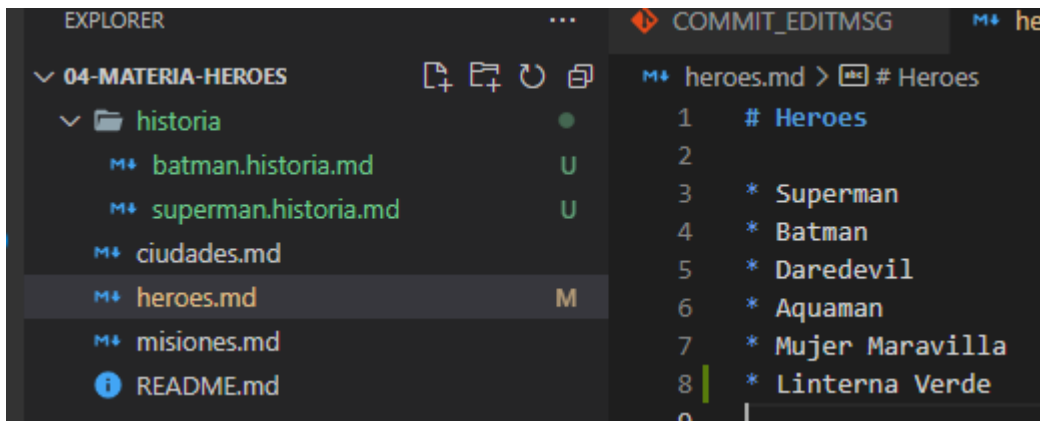
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Thu Jan 12 13:36:40 2023 +0100
#
# On branch main
# Changes to be committed:
#       new file:   ciudades.md
#       new file:   historia/batman.historia.md
#       new file:   historia/superman.historia.md
#
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

.git/COMMIT_EDITMSG [unix] (13:37 12/01/2023) 1,1 All
cateria-Heroes/.git/COMMIT_EDITMSG" [unix] 13L 360B

Al ejecutar el comando sin poner al final `-m` “nuevo texto para el commit” se abrirá el editor y podremos cambiar en él el texto del commit:

```
.git > COMMIT_EDITMSG
1  Agregada la hitoria de batman y superman
2
3  # Please enter the commit message for your changes. Lines starting
4  # with '#' will be ignored, and an empty message aborts the commit.
5  #
6  # Date:      Thu Dec 16 12:06:24 2021 +0100
7  #
```

Editamos el archivo héroes.md y agregamos al héroe Linterna Verde:



```
EXPLORER
04-MATERIA-HEROES
  historia
    batman.historia.md
    superman.historia.md
  ciudades.md
  heroes.md (M)
  misiones.md
  README.md

COMMIT_EDITMSG
1  # Heroes
2
3  * Superman
4  * Batman
5  * Daredevil
6  * Aquaman
7  * Mujer Maravilla
8  * Linterna Verde
9
```

Hacemos un nuevo commit:

```
git commit -am "agregamos el heroe Linterna Verde"
```

Ejecutamos el comando `git lg` para ver un listado de los commits que hemos realizado:

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git commit -am "agregamos el heroe Linterna Verde"
warning: in the working copy of 'heroes.md', LF will be replaced by CRLF the next time Git touches it
[main 9e0baef] agregamos el heroe Linterna Verde
1 file changed, 6 insertions(+), 5 deletions(-)
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 9e0baef - (21 seconds ago) agregamos el heroe Linterna Verde - Nieves (HEAD -> main)
* 27b9dc3 - (5 minutes ago) agregada la historia de batman y superman - Nieves
* 07470c5 - (7 minutes ago) heroes.agregado - Nieves
* 05914eb - (7 minutes ago) misiones.md agregado - Nieves
* 2010295 - (10 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> █
```

5 Viajes en el tiempo, resets y reflogs

Vamos a añadir el héroe Robin en el archivo heroes.md:

En vez de hacer un nuevo commit para añadir este cambio queremos que haya un solo commit con la agregación de Robin y Linterna Verde.

Hacemos un `git lg`: Copia una captura e indica el código hash del commit al que queremos volver

```
* 9e0baef - (21 seconds ago) agregamos el heroe Linterna Verde - Nieves (HEAD -> main)
```

Vamos a regresar a un commit anterior usando como referencia el código hash del commit al que queremos volver:

```
git reset --soft 27b9dc3
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git commit -am "agregamos el her
oe Linterna Verde"
warning: in the working copy of 'heroes.md', LF will be replaced by CRLF the next ti
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git reset --soft 27b9dc3
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 27b9dc3 - (16 minutes ago) agregada la historia de batman y superman - Nieves (HEA
D -> main)
* 07470c5 - (17 minutes ago) heroes.agregado - Nieves
* 05914eb - (18 minutes ago) misiones.md agregado - Nieves
* 2010295 - (21 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git commit -am "agregamos los he
roes Linterna Verde y Robin"
warning: in the working copy of 'heroes.md', LF will be replaced by CRLF the next ti
me Git touches it
[main 7170f67] agregamos los heroes Linterna Verde y Robin
 1 file changed, 7 insertions(+), 5 deletions(-)
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 7170f67 - (4 seconds ago) agregamos los heroes Linterna Verde y Robin - Nieves (HE
AD -> main)
* 27b9dc3 - (17 minutes ago) agregada la historia de batman y superman - Nieves
* 07470c5 - (18 minutes ago) heroes.agregado - Nieves
* 05914eb - (18 minutes ago) misiones.md agregado - Nieves
* 2010295 - (21 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> █
```

Hacemos un nuevo commit:

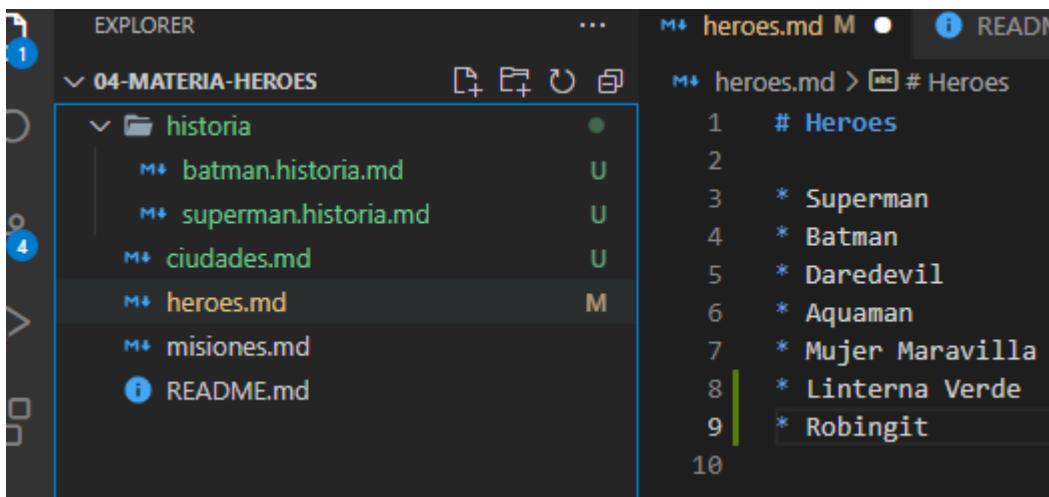
```
git commit -am "agregamos los heroes Linterna Verde y Robin"
```

Ahora decidimos que los últimos cambios realizados no están bien y que queremos volver al punto en el que agregamos el archivo de heroes. Y además queremos que los archivos de los que se hubiera hecho commit se queden fuera del stage si es que los habíamos subido. Para ello hacemos un git reset --mixed:

```
git reset --mixed 07470c5
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 7170f67 - (4 seconds ago) agregamos los heroes Linterna Verde y Robin - Nieves (HEAD -> main)
* 27b9dc3 - (17 minutes ago) agregada la historia de batman y superman - Nieves
* 07470c5 - (18 minutes ago) heroes.agregado - Nieves
* 05914eb - (18 minutes ago) misiones.md agregado - Nieves
* 2010295 - (21 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git reset --mixed 07470c5
Unstaged changes after reset:
M      heroes.md
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> 
```

Si no ponemos una opción del modo de reset (--soft --mixed etc) el tipo de reset que se hace es el --mixed ya que es el modo por defecto.

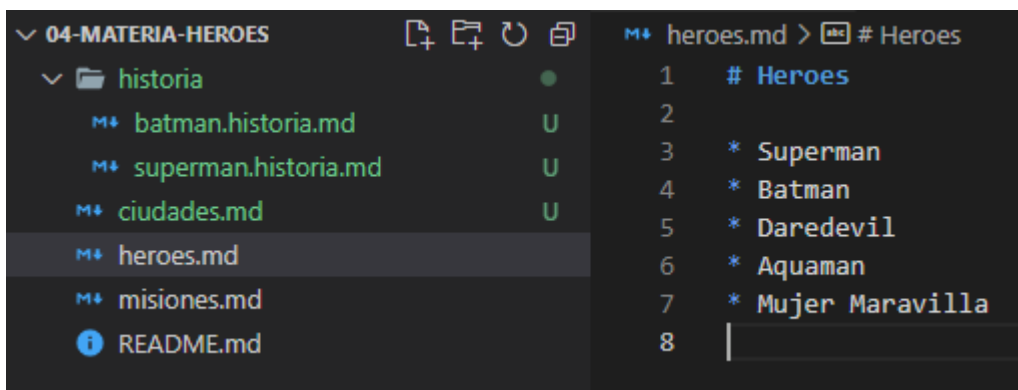


Vemos que los archivos han pasado a estar con la U de untracked es decir sin seguimiento pero que los últimos cambios realizados en heroes.md se han mantenido.

Si queremos hacer un reset que borre los cambios usaremos la opción --hard:

```
git reset --hard
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git
reset --hard
HEAD is now at 07470c5 heroes.agregado
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git
lg
* 07470c5 - (24 minutes ago) heroes.agregado - Nieves (HEAD -> main)
* 05914eb - (24 minutes ago) misiones.md agregado - Nieves
* 2010295 - (27 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> 
```



Vemos que el archivo heroes.md ha vuelto al estado del commit al que hemos regresado.

Hacemos un reset hard al commit donde agregamos las misiones:

```
git reset --hard 07470c5
```

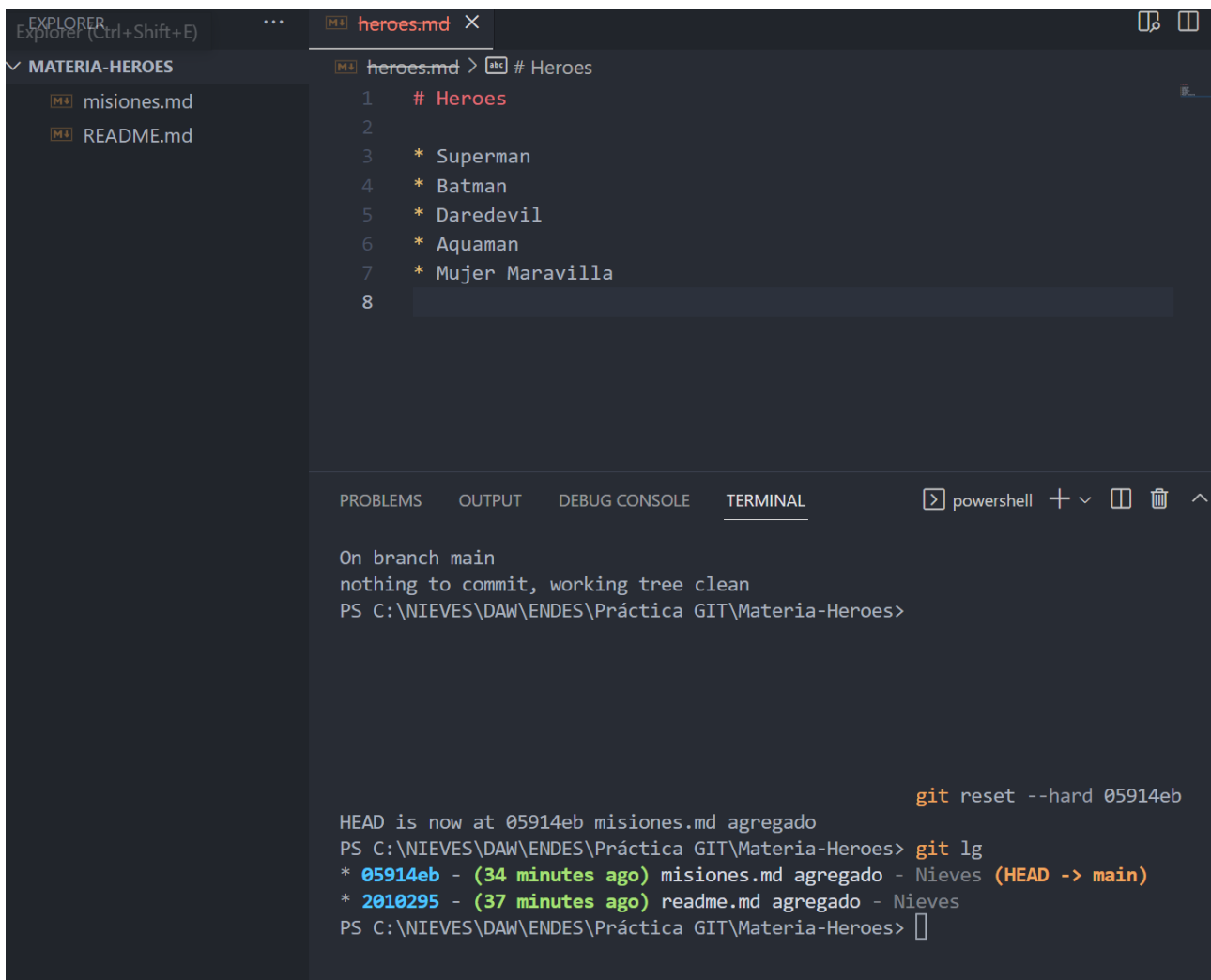
```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 1a38d87 - (3 minutos ago) historia agregada - Nieves (HEAD -> main)
* 07470c5 - (30 minutos ago) heroes.agregado - Nieves
* 05914eb - (31 minutos ago) misiones.md agregado - Nieves
* 2010295 - (34 minutos ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git reset --hard 07470c5
HEAD is now at 07470c5 heroes.agregado
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> 
```

Si hacemos un nuevo `git lg`:

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 07470c5 - (32 minutos ago) heroes.agregado - Nieves (HEAD -> main)
* 05914eb - (32 minutos ago) misiones.md agregado - Nieves
* 2010295 - (35 minutos ago) readme.md agregado - Nieves
```

Vemos que sólo nos quedan dos commits. Si ahora nos damos cuenta de que estaba todo bien y que queremos volver a un commit de los que borramos, todavía es posible ya que git guarda un historial:

```
git reset --hard 05914eb
```



```
EXPLORER
Explorer (Ctrl+Shift+E)
MATERIA-HEROES
  misiones.md
  README.md
  heroes.md > # Heroes
    1 # Heroes
    2
    3 * Superman
    4 * Batman
    5 * Daredevil
    6 * Aquaman
    7 * Mujer Maravilla
    8

TERMINAL
powershell
On branch main
nothing to commit, working tree clean
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes>

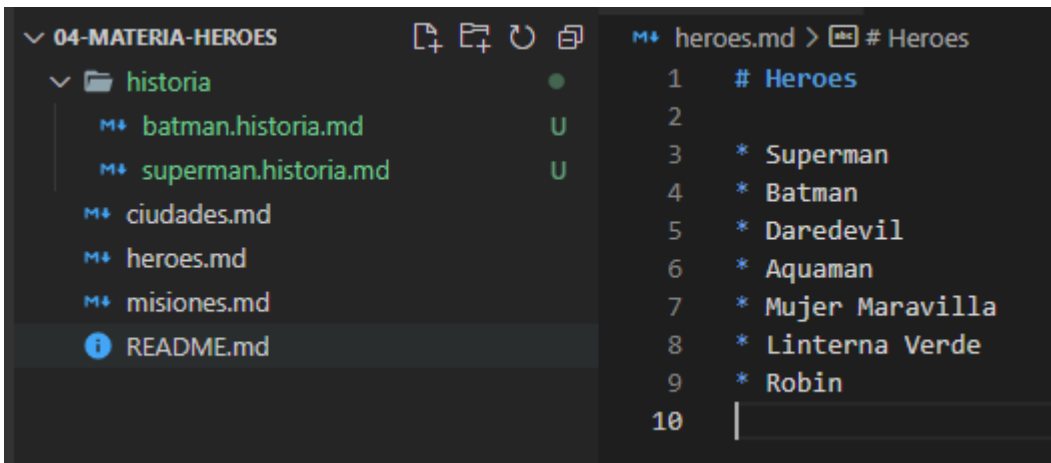
git reset --hard 05914eb
HEAD is now at 05914eb misiones.md agregado
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 05914eb - (34 minutes ago) misiones.md agregado - Nieves (HEAD -> main)
* 2010295 - (37 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes>
```

Como podemos ver en este historial se guardan tanto los commit como los reset que se fueron realizando. Esto nos permite ver los hash de los commit borrados y si por ejemplo queremos volver al commit en el que agregamos a los héroesLinterna Verde y Robin no tenemos más que ejecutar el siguiente comando:

```
git reflog
```

```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git reflog
05914eb (HEAD -> main) HEAD@{0}: reset: moving to 05914eb
07470c5 HEAD@{1}: reset: moving to 07470c5
1a38d87 HEAD@{2}: commit: historia agregada
07470c5 HEAD@{3}: reset: moving to HEAD
07470c5 HEAD@{4}: reset: moving to 07470c5
7170f67 HEAD@{5}: commit: agregamos los heroes Linterna Verde y Robin
27b9dc3 HEAD@{6}: reset: moving to 27b9dc3
9e0baef HEAD@{7}: commit: agregamos el heroe Linterna Verde
27b9dc3 HEAD@{8}: commit (amend): agregada la historia de batman y superman
5188561 HEAD@{9}: commit: historia agregado
07470c5 HEAD@{10}: commit: heroes.agregado
05914eb (HEAD -> main) HEAD@{11}: commit: misiones.md agregado
2010295 HEAD@{12}: commit (initial): readme.md agregado
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes>
```

Vemos que todo vuelve a ese punto:



```
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git reset --hard 7170f67
HEAD is now at 7170f67 agregamos los heroes Linterna Verde y Robin
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git lg
* 7170f67 - (20 minutes ago) agregamos los heroes Linterna Verde y Robin - Nieves
S (HEAD -> main)
* 27b9dc3 - (37 minutes ago) agregada la historia de batman y superman - Nieves
* 07470c5 - (38 minutes ago) heroes.agregado - Nieves
* 05914eb - (38 minutes ago) misiones.md agregado - Nieves
* 2010295 - (42 minutes ago) readme.md agregado - Nieves
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes> git status
On branch main
nothing to commit, working tree clean
PS C:\NIEVES\DAW\ENDES\Práctica GIT\Materia-Heroes>
```