



Note que Git especifica cuántos archivos han sido cambiados y cuántas líneas de código han sido insertadas o eliminadas.

El autor de un `commit` debe comentar lo que hizo mediante un breve mensaje. Esto ayuda a otras personas (y a sí mismo) a comprender más adelante el propósito de los cambios.

En caso de equivocarse en el `commit`, puede modificarlo usando la opción `amend`.

- Use el siguiente comando para modificar el mensaje de confirmación.

```
$ git commit --amend -m "nuevo mensaje"
```

//la opción `-m` permite escribir un nuevo mensaje desde la línea de comandos sin tener que abrir un editor.

- Si desea modificar archivos del `commit`, realice los ajustes correspondientes en la copia de trabajo local, agréguelos al área de preparación y ejecute el siguiente comando.

```
$ git commit --amend --no-edit
```

//la opción `--no-edit` permite hacer las correcciones en la confirmación sin cambiar el mensaje.

4.6. Consulte el historial de cambios.

```
$ git log
```

```
user:~/recetario$git log
commit 212ccba6d186d49b979b126613a4e62049da546e (HEAD -> master)
Author: Adriana Vargas <a.vargas20@uniandes.edu.co>
Date: Fri Apr 17 11:30:11 2020 -0500
```

```
Se añade codigo base
```

Note que además de la información básica del `commit`, también es posible identificar la rama donde se realizaron los cambios y el identificador único del mismo. Este último le permitirá buscarlo o devolverse de `commit` más adelante.

Este comando lista en orden cronológico el historial de los `commits` que se han realizado al repositorio.

Git también le permite listar el detalle de los cambios que ocurrieron en cada `commit`. Utilice la bandera `-p` con el comando `git log` para agregar este tipo de información.

En este momento, ya cuenta con el código base del proyecto en el repositorio y está listo para empezar a trabajar en él.