

Actividad 2.5 - Deshaciendo cambios en Git

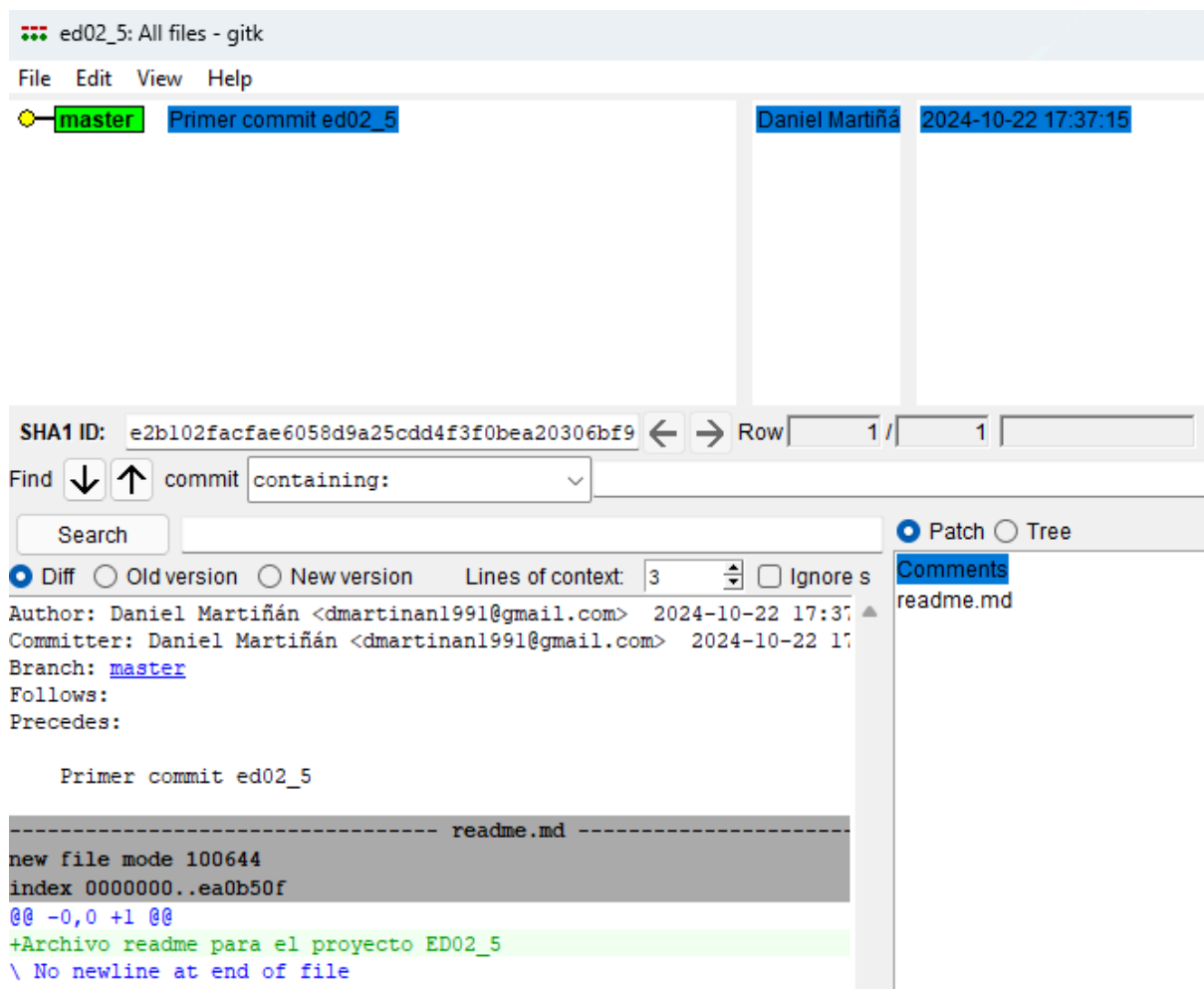
Introducción.....	2
Tareas:.....	2
Preguntas de reflexión:.....	4
Instrucciones de entrega.....	4

Introducción

En esta actividad nos centramos en las diferentes formas de deshacer cambios en Git. Se plantearán diferentes situaciones a manejar a la hora de intentar deshacer ciertos cambios dentro de un repositorio de git.

Tareas:

- A. Crea un nuevo repositorio: crea un repositorio nuevo llamado ed02_5, que incluya un archivo README.md. Puedes crearlo desde github y, posteriormente, clonarlo, o puedes crearlo en local y luego vincularlo con un repositorio nuevo de github. Lo importante es que ese repositorio tenga una rama principal (llamada main o master) y al menos un commit, en el que se incluya el archivo readme.md con contenido (una línea de texto, con el contenido que quieras). El estado del repositorio en este punto debe ser algo así:

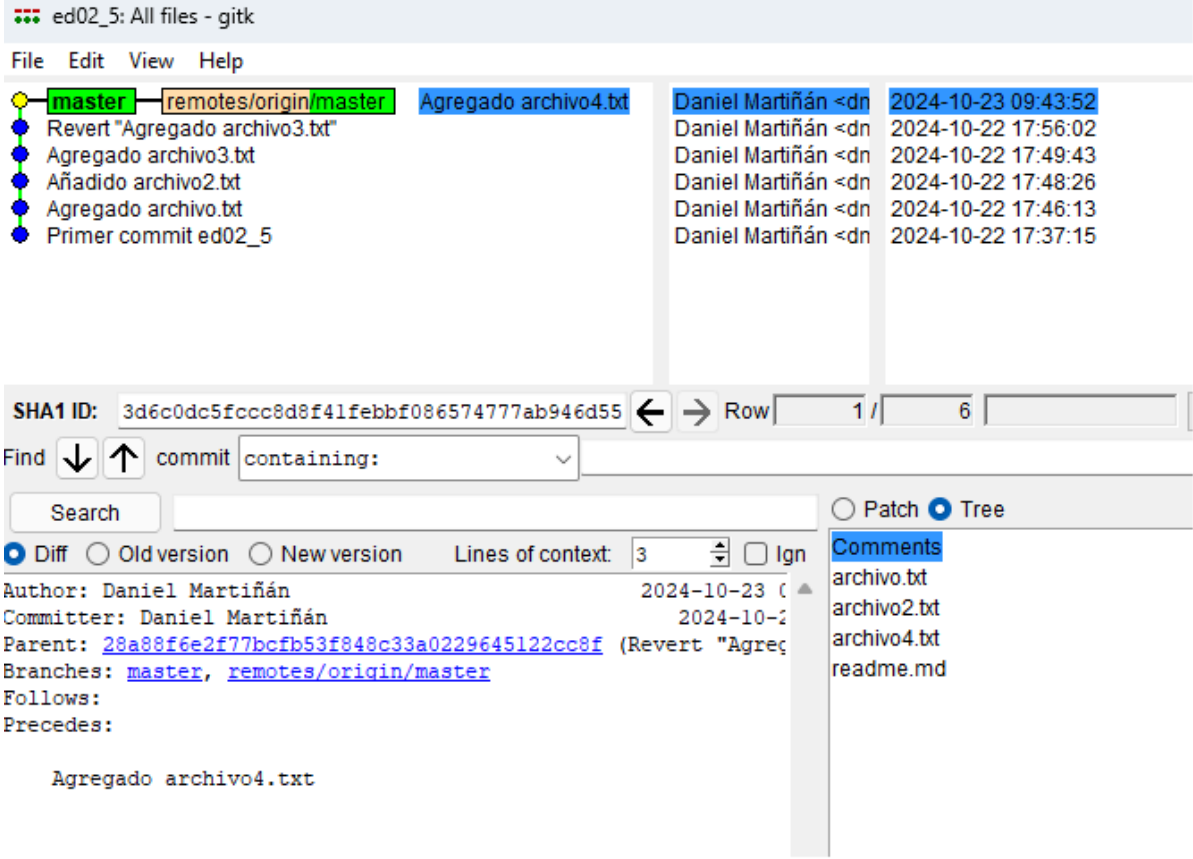


- B. Quitar un archivo del staging area: Crea un nuevo archivo llamado archivo.txt en tu repositorio. Agrega el archivo al *staging area* utilizando `git add archivo.txt`. Utiliza `git restore --staged archivo.txt` para quitar el archivo del *staging*

area. Verifica que el archivo ya no esté en el *staging area* utilizando `git status`. Posteriormente, vuelve a añadir el archivo a la *staging area* y crea un commit con el mensaje "Agregado archivo.txt".

- C. Deshacer modificaciones de un archivo: modifica el archivo `archivo.txt` (añádele una línea de texto). Comprueba el estado del repositorio (deberías ver que el archivo está en la *working area*). Elimina las modificaciones, dejándolo en el estado en el que estaba en el último commit. Muestra el comando empleado y el estado del repositorio tras su ejecución.
- D. Deshacer un commit: Crea un nuevo archivo llamado `archivo2.txt` en tu repositorio. Agrega el archivo al *staging area* y realiza un commit utilizando `git commit -m "Agregado archivo2.txt"`. Posteriormente, deshaz el commit de manera que el archivo `archivo2.txt` se quede en la *staging area*. Muestra el comando empleado y el estado del repositorio tras su ejecución. Vuelve a crear el commit con el mismo mensaje (Agregado archivo2.txt).
- E. Deshacer cambios ya pusheados a remoto: Crea un nuevo archivo llamado `archivo3.txt` en tu repositorio. Agrega el archivo al *staging area* y realiza un commit con el mensaje "Agregado archivo3.txt". Envía los cambios a remoto utilizando `git push`. Utiliza `git revert HEAD` para deshacer los cambios. Verifica que los cambios hayan sido deshechos utilizando `git log` y `git status`. ¿Que ha sucedido en el directorio tras este paso? Sube los cambios realizados por `git revert` al servidor, utilizando de nuevo `git push`.
- F. Deshacer cambios en un commit anterior: Crea un nuevo archivo llamado `archivo4.txt` en tu repositorio. Agrega el archivo al *staging area* y realiza un commit con el mensaje "Agregado archivo4.txt". A continuación, edita el archivo (añádele una línea de texto) y vuelve a comitear, con el mensaje "modificado archivo4.txt". Utiliza `git reset --hard HEAD~1` para deshacer los cambios en el commit anterior. Verifica que los cambios hayan sido deshechos utilizando `git log` y `git status`. ¿En qué estado se encuentra el directorio de trabajo tras esta operación? Haz push de los cambios para que se sincronicen con el repositorio remoto.

Tras la ejecución de todos los pasos, el estado de tu repositorio debería verse de la siguiente manera:



Preguntas de reflexión:

- ¿Cuál es la diferencia entre git restore y git rm?
- ¿Qué sucede cuando se utiliza git revert en un commit que ya ha sido pusheado a remoto?
- ¿Cómo se puede deshacer un cambio en un commit anterior sin afectar los commits posteriores?
- ¿Se te ocurren más situaciones en las que tener que revertir cambios que no se hayan planteado en esta práctica?

Instrucciones de entrega.

Debes entregar una memoria en formato pdf con el nombre

apellidos_nombre_ED02_5.pdf

que incluya todas las capturas así como indicaciones que consideres relevantes en la realización del ejercicio. Incluye, al final del documento, un enlace al repositorio creado en github, para poder ser revisado.