

Condiciones de entrega:

- Informe de la práctica, que al menos incluya: pantallazos, comentarios.
- El taller es en grupo y su entrega se realiza por medio del intu.
- En el informe deben incluir los nombres de los participantes, evidencias de cada uno y el enlace al fork del repositorio.

Enunciado

Git reset, rebase y cherry-pick son distintas formas que nos ofrece git para borrar nuestros commits. Cada una con su peculiaridad. Aunque no está bien visto ya que estamos «borrando la historia», puede ocurrir que queramos borrar varios commits que hemos hecho en algún momento y hemos subido al repositorio.

Usando el comando “*git log*” se pueden obtener los detalles de cada commit (hash, autor, fecha y comentario). Sin embargo, en nuestra practica solo nos importara el hash y el comentario de los commits. Por esto, tendrán que usar “*git log --pretty=oneline*” para obtener esta información.

```
C:\Users\cuatr\IdeaProjects\PI1\PI1-Git-Commits(main)
λ git log --pretty=oneline
1d95476b5d933467206f59ecb2c37b323e80cb4b (HEAD -> main, origin/main) add author
f599329e6c881a05e43474d9e324d574134f2307 commit to remove2
f84a0e846b44f1646ee62eae132f291b644c11d6 commit to remove1
a489a2d3975e50fd05ae5a8802d9408530775b78 add project description
e92731b9bec28cf3a335bddfd191a97d1438da5 add cars toString and print
6e75cd7b1ff4b6b04cab0cf0fb5c0b12190fc435 add list on main and add cars on for loop
75bc803cc2df3aa8913486afe61e500e03937900 add setters on car class
1d742be062bc4429eb6643ad38bbd80f47ddde03 add getters on car
c2c96c27352dde022b250378a7bc7a7e62029ed4 add car class
55af227b55619e3a4c2502b856f79a92e574f73f delete description of project on readme
a9853548200230aae5efefc0db34a9b827593e0e add for loop
97aeb992f79b74840a94d6b7e3d1baa546488bb7 add README.md
c23fb337d0280fe4a8b8f33c0341213310ddbe2e first commit
```

Para esta práctica, un miembro de cada grupo deberá realizar un Fork del repositorio <https://github.com/cuatrosr/PI1-Git-Commits> invitando a los demás miembros a ese Fork. Luego, seguirán los pasos del taller:

1. Cada uno deberá clonar el Fork creando un branch con su nombre.

Git Rebase

1. Deberán borrar los dos commits consecutivos “commit to remove2” y “commit to remove1”.

Git cherry-pick y reset

1. Encontrar el commit "add car class" y moverse usando *git checkout* al hash del commit.
2. Desde ese commit crean una nueva rama auxiliar (llámenla como quieran) y se cambian a ella.
3. Deberán borrar todos los commits excepto "add list on main and add cars on for loop", "add cars toString and print" y "add author". Osea, deberan usar el commando *git cherry pick* con los hash de estos commits.
4. Se deben mover a la rama de su nombre y devolverse al commit "add car class" usando *git reset --hard*.
5. Realizan un merge con la branch auxiliar que crearon.
6. Realizan un git push de manera forzada de la branch al remoto.

Recomendaciones

Usen *git log* para entender que se esta haciendo con cada comando (de igual manera suben las evidencias tomándole foto al log). De esta forma pueden entender de una manera visual el cambio en el historial de commits. Tengan mucho cuidado que estos comandos pueden ser peligrosos porque borran directamente el historial del proyecto.

Se pueden guiar si no entendieron de estas fuentes:

<https://vergaracarmona.es/reset-rebase-cherry-pick/>

<https://jonsegador.com/2015/06/borrar-commits-de-git-reset-rebase-cherry-pick/>