

Rebase and Merge

rebase and merge sirve para reorganizar los commits de una rama en una secuencia lineal antes de hacerle merge con otra rama. En lugar de crear un commit para fusionarlas, rebase hace que cada commit de la rama actual se aplique sobre la rama de destino.

Cuando se usa:

- Cuando se desea mantener un historial de commits limpio y lineal.
- Cuando se trabaja en un entorno colaborativo y se quiere mantener el orden de commits para mejor comprensión del historial.

Squash and Merge

El comando squash and Merge a diferencia del rebase este combina todos los commits de una rama en un solo commit antes de hacer la fusion con otra rama. Esto reduce el número de commits en el historial.

Cuando se usa:

- Cuando se desea evitar una gran cantidad de commits pequeños en la rama principal.
- Para mantener un historial de commits más ordenado y compacto.
- Ideal cuando se fusiona una funcionalidad completa y no se necesita conservar los detalles intermedios de desarrollo.

Reset

Explicación: El comando git reset se usa para mover el HEAD (**HEAD** es un puntero que indica el commit actual en el que se está trabajando) de la rama actual a un commit anterior, descartando los cambios posteriores en la historia del repositorio.

Tipos de git reset:

- --soft: Mueve HEAD al commit especificado pero mantiene los cambios en el área de staging
- --mixed: Mueve HEAD al commit especificado y deshace los cambios del área de staging, pero los deja en el directorio de trabajo
- --hard: Mueve HEAD al commit especificado y borra completamente los cambios en el área de staging y el directorio de trabajo

Cuando se usa

- Para deshacer commits locales antes de hacer push.
- Para modificar el historial de commits antes de compartir cambios

- Cuando se quiere eliminar cambios en el área de staging o en el working directory.

Revert

Explicación: El comando git revert crea un nuevo commit que invierte los cambios de un commit anterior sin modificar la historia del repositorio

Cuando se usa:

- Cuando se quiere revertir un cambio sin modificar el historial de commits
- Para corregir errores en commits previos sin afectar a otros colaboradores
- Es seguro para usar en ramas que se comparten

Diferencias entre Revert y Reset:

Reset modifica el historial de commits, Revert no.

Revert es seguro para repositorios compartidos, Reset no.

Reset afecta el área de staging, Revert no ya que crea y añade un commit al historial.