

Guía para borrar un commit en git desde consola

Caso 1: Borrar el último commit sin haber hecho push

Este es el caso más sencillo, ya que el commit solo está en tu repositorio local y no en el remoto. Aquí puedes utilizar el comando `git reset`.

1. Verifica el estado del commit

Usa el siguiente comando para ver el historial de tus commits y confirmar cuál es el último:

```
git log --oneline
```

Esto mostrará una lista de commits recientes con sus hashes. El commit que deseas eliminar estará al principio de la lista.

2. Borrar el último commit con `git reset`

Si decides que quieres eliminar el último commit y también quieres borrar sus cambios en el directorio de trabajo, usa el siguiente comando:

```
git reset --hard HEAD~1
```

- `--hard`: Este parámetro elimina tanto el commit como los cambios hechos en ese commit del área de trabajo.
- `HEAD~1`: Indica que quieres regresar un commit atrás.

Si solo quieres eliminar el commit pero mantener los cambios en el área de trabajo para revisarlos o hacer un commit diferente, puedes usar:

```
git reset --soft HEAD~1
```

3. Verifica que el commit fue eliminado

Revisa nuevamente tu historial de commits para confirmar que el commit ha sido eliminado:

```
git log --oneline
```

Caso 2: Borrar el último commit después de haber hecho push

En este caso, el commit ya está en el repositorio remoto, lo que puede afectar a otros colaboradores si ya lo han descargado. Aquí tienes dos opciones:

deshacer y forzar el push (método más directo, pero potencialmente riesgoso) o **revertir el commit** (más seguro y recomendado).

Opción 1: Usar `reset` y `push --force`

1. Elimina el último commit

Al igual que en el caso anterior, usa `git reset`:

```
git reset --hard HEAD~1
```

2. Forzar el push al repositorio remoto

Para actualizar el repositorio remoto y borrar el commit, usa `push` con el parámetro `--force`:

```
git push origin main --force
```

⚠ Advertencia: Forzar un push puede crear problemas para otros colaboradores si ya han descargado el commit. Utiliza este método solo si eres el único que trabaja en el proyecto o si estás seguro de que nadie ha descargado los últimos cambios.

Opción 2: Revertir el commit (recomendado)

Este método es menos riesgoso y evita la necesidad de usar `--force`. Aquí, en lugar de borrar el commit, creas uno nuevo que revierta los cambios.

1. Revertir el último commit

Usa el comando `git revert` para crear un nuevo commit que revierta los cambios del último commit:

```
git revert HEAD
```

Esto abrirá un editor para escribir un mensaje de commit de reversión (normalmente algo como "Revertir cambios del commit "). Guarda y cierra el editor para completar el commit de reversión.

2. Sube los cambios al repositorio remoto

Luego, usa `git push` normalmente para que el commit de reversión se agregue al repositorio remoto:

```
git push origin main
```

Resumen rápido de comandos

Acción	Comando
Borrar último commit localmente (sin push)	<code>git reset --hard HEAD~1</code>
Borrar último commit localmente y conservar cambios	<code>git reset --soft HEAD~1</code>
Borrar último commit y forzar push (riesgoso)	<code>git reset --hard HEAD~1</code> <code>git push origin main --force</code>
Revertir último commit y push (seguro)	<code>git revert HEAD</code> <code>git push origin main</code>

Buenas prácticas

- Usa `reset --hard` solo cuando estés seguro de que no necesitas los cambios.
- Prefiere `revert` sobre `reset` para commits ya subidos al remoto; es más seguro y claro para otros colaboradores.
- Considera hablar con tus colaboradores antes de forzar un push, ya que esto puede causar conflictos si ellos ya han descargado el commit eliminado.