

git revert

Crea un nuevo commit que revierte los cambios de los commits previos:

```
git revert <commit-hash>
```

Crea un nuevo commit que deshace los cambios de un commit anterior (no debe usarse con commits de fusión)

```
git revert -m <parent-number> <commit-hash>
```

Recupera cambios desde el más reciente stash en la pila de stash

git stash

Guarda los cambios en el árbol de trabajo, pero no los confirma:

```
git stash
git stash save <message>
```

Almacena todos los cambios desde el commit anterior (si no se especifica el mensaje del stash, se usará el mensaje del commit más reciente)

```
git stash pop
```

Recupera los cambios desde el stash más reciente en la pila de stash

```
git stash apply stash@{n}
```

Recupera los cambios del stash en el índice *n* en la pila de stash

```
git stash list
```

Enumera todas las entradas en la pila de stash

```
git stash clear
```

Borra la pila de stash

Conflictos de fusión

- Los conflictos ocurren cuando la misma línea en un archivo se ha cambiado en dos ramas distintas, o cuando un archivo ha sido eliminado en una rama, pero editado en otra.
- Los equipos que trabajan asíncronamente suelen tener conflictos de fusión.
- Los conflictos se pueden resolver conservando los cambios que deseas, y después agregando y confirmando los cambios.
- VS Code proporciona herramientas de gran utilidad para los conflictos de fusión.

```
<body>
<<<<<< HEAD <!-- HEAD indica la rama actual -->
  <h1>La rama actual cambió el título</h1>
=====
  <h1>La otra rama tiene un título distinto</h1>
</body>
>>>>>> other-branch-name
```

