

## Stash

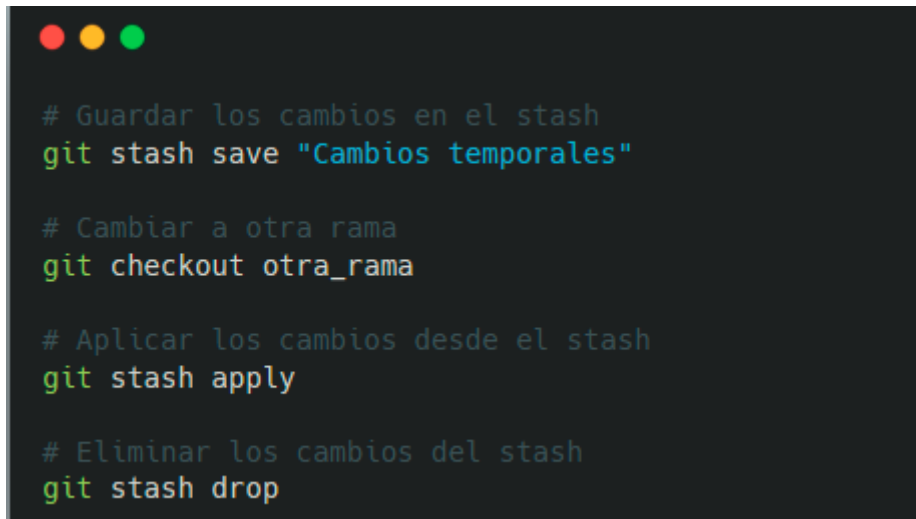
El comando *Stash* se usa para guardar temporalmente los cambios no confirmados en una rama sin confirmarlos ni desecharlos.

**-Guardado de cambios temporales:** Es útil cuando necesitas cambiar de rama o realizar alguna acción que requiera un directorio de trabajo limpio sin perder tus cambios en progreso.

**-Aplicación y eliminación de Stash:** Puedes aplicar o eliminar los cambios almacenados en el Stash cuando sea conveniente.

**-Múltiples Stashes:** Puedes tener múltiples Stashes y nombrarlos para identificarlos más fácilmente.

Supongamos que estás trabajando en una rama con cambios no confirmados y necesitas cambiar a otra rama sin perder esos cambios. Puedes usar **stash**:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It displays four lines of Git commands with their respective comments in a light blue monospace font. The commands are: 'git stash save "Cambios temporales"', 'git checkout otra\_rama', 'git stash apply', and 'git stash drop'.

```
# Guardar los cambios en el stash
git stash save "Cambios temporales"

# Cambiar a otra rama
git checkout otra_rama

# Aplicar los cambios desde el stash
git stash apply

# Eliminar los cambios del stash
git stash drop
```

Ejemplo stash

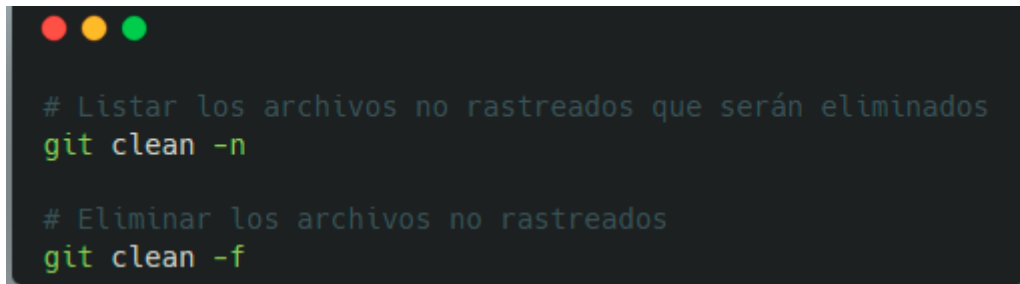
## Clean

El comando *Clean* se utiliza para eliminar archivos y directorios no rastreados en el repositorio.

**-Limpieza del espacio de trabajo:** Ayuda a mantener el espacio de trabajo limpio eliminando archivos generados automáticamente o no deseados.

**-Precaución:** Debe usarse con precaución, ya que eliminará permanentemente los archivos no rastreados. Se recomienda revisar los cambios antes de ejecutar el comando.

Por ejemplo, para eliminar todos los archivos no rastreados:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains two lines of text: a comment in light blue and a command in green.

```
# Listar los archivos no rastreados que serán eliminados  
git clean -n  
  
# Eliminar los archivos no rastreados  
git clean -f
```

Ejemplo clean

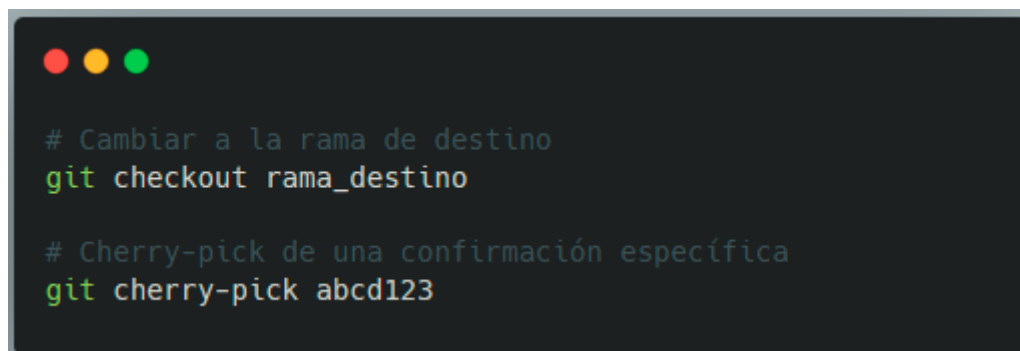
## Cherry-pick

El comando *Cherry-pick* se utiliza para aplicar una confirmación específica de una rama a otra.

**-Selección de confirmaciones:** Permite seleccionar confirmaciones individuales y aplicarlas a otra rama sin la necesidad de fusionar ramas completas.

**-Resolución de conflictos:** Puede ser necesario resolver conflictos si las confirmaciones que intentas aplicar entran en conflicto con los cambios existentes en la rama de destino.

Por ejemplo, para aplicar la confirmación con el identificador `abcd123` de la rama `rama_fuente` a la rama actual:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains two lines of text: a comment in light blue and a command in green.

```
# Cambiar a la rama de destino  
git checkout rama_destino  
  
# Cherry-pick de una confirmación específica  
git cherry-pick abcd123
```

Ejemplo cherry-pick

Estos comandos avanzados de Git te permiten realizar operaciones más específicas y personalizadas en tu flujo de trabajo de control de versiones, facilitando la colaboración, la gestión del historial de confirmaciones y la limpieza de tu espacio de trabajo.