

Deshaciendo cambios

GIT

❑ Deshacer cambios

❑ Modificando la última confirmación

- ❑ Uno de los casos más comunes en el que se quiere deshacer cambios es cuando se confirma demasiado pronto y olvidamos añadir algún archivo, o nos equivocamos al introducir el mensaje de confirmación.
- ❑ Para volver a repetir la confirmación, se puede ejecutar un commit con la opción --amend:

```
$ git commit --amend
```

GIT

☐ **Deshacer cambios**

- ☐ Este comando utiliza lo que haya en el área de preparación para la confirmación.
- ☐ Si no se ha hecho ningún cambio desde la última confirmación, el snapshot será exactamente igual, y lo único que cambia será el mensaje de confirmación.
- ☐ Se lanzará el editor de texto para introducir el mensaje, pero ya contendrá el mensaje de la confirmación anterior.
- ☐ Se puede editar el mensaje, igual que siempre, pero se sobrescribirá la confirmación anterior.

GIT

❑ **Deshacer cambios**

- ❑ Por ejemplo, si se confirma algo y luego nos damos cuenta de que olvidamos preparar los cambios en uno de los archivos:

```
$ git commit -m 'initial commit'  
$ git add forgotten_file  
$ git commit --amend
```

GIT

☐ **Deshaciendo la preparación de un archivo**

- ☐ El comando que se usa para determinar el estado de ambas áreas recuerda como deshacer los modificaciones.
- ☐ Si por ejemplo, modificamos dos archivos, y queremos confirmarlos como cambios separados, pero ponemos sin querer `git add *` y prepaamos ambos, el comando `git status` recuerda como arreglarlo

GIT

□ Deshaciendo la preparación de un archivo

```
$ git add *.txt
```

```
alberto@DESKTOP-P4DN3CA MINGW64 ~/practicas (master)
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
renamed:      ejemplo2.txt -> ejemplo3.t  
new file:     ejemplo5.txt  
new file:     g1.tx1  
new file:     g2.txt  
new file:     simplegit-progit
```

GIT

□ Deshaciendo la preparación de un archivo

```
$ git reset HEAD g2.txt
```

```
alberto@DESKTOP-P4DN3CA MINGW64 ~/practicas (master)
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
renamed:      ejemplo2.txt -> ejemplo3.t
```

```
new file:     ejemplo5.txt
```

```
new file:     g1.tx1
```

```
new file:     simplegit-progit
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
g2.txt
```

GIT

☐ **Deshaciendo la modificación de un archivo**

- ☐ ¿Qué pasa si no queremos mantener las modificaciones que se han hecho sobre un archivo .
- ☐ Si queremos revertir el archivo al mismo estado en el que estaba cuando hicimos nuestro último commit, git status también nos indica como hacerlo.

GIT

❑ Deshaciendo la modificación de un archivo

❑ Por ejemplo:

```
$ echo adios > g2.txt
```

```
alberto@DESKTOP-P4DN3CA MINGW64 ~/practicas (master)
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    modified:   g2.txt
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   g2.txt
```

GIT

❑ Deshaciendo la modificación de un archivo

❑ Por ejemplo:

```
$ git checkout -- g2.txt

alberto@DESKTOP-P4DN3CA MINGW64 ~/practicas (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   g2.txt
```

GIT

☐ **Deshaciendo la modificación de un archivo**

☐ **Práctica**

- ☐ Añadir un fichero al repositorio
- ☐ Validarlo
- ☐ Modificarlo con algún contenido adicional
- ☐ Recuperar el valor que tenemos en el repositorio
- ☐ Comprobar que funciona

GIT

☐ **Deshaciendo la modificación de un archivo**

☐ **Práctica**

- ☐ Añadir un fichero al repositorio
- ☐ Validarlo
- ☐ Modificarlo con algún contenido adicional
- ☐ Recuperar el valor que tenemos en el repositorio
- ☐ Comprobar que funciona