

Deshacer Cosas

En cualquier momento puede que quieras deshacer algo. Aquí repasaremos algunas herramientas básicas usadas para deshacer cambios que hayas hecho. Ten cuidado, a veces no es posible recuperar algo luego que lo has deshecho. Esta es una de las pocas áreas en las que Git puede perder parte de tu trabajo si cometes un error.

Uno de las acciones más comunes a deshacer es cuando confirmas un cambio antes de tiempo y olvidas agregar algún archivo, o te equivocas en el mensaje de confirmación. Si quieres rehacer la confirmación, puedes reconfirmar con la opción `--amend`:

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

Este comando utiliza tu área de preparación para la confirmación. Si no has hecho cambios desde tu última confirmación (por ejemplo, ejecutas este comando justo después de tu confirmación anterior), entonces la instantánea lucirá exactamente igual y lo único que cambiarás será el mensaje de confirmación.

Se lanzará el mismo editor de confirmación, pero verás que ya incluye el mensaje de tu confirmación anterior. Puedes editar el mensaje como siempre y se sobrescribirá tu confirmación anterior.

Por ejemplo, si confirmas y luego te das cuenta que olvidaste preparar los cambios de un archivo que querías incluir en esta confirmación, puedes hacer lo siguiente:

```
$ git commit -m 'initial commit'

$ git add forgotten_file

$ git commit --amend
```

Al final terminarás con una sola confirmación - la segunda confirmación reemplaza el resultado de la primera.

Deshacer un Archivo Preparado

Las siguientes dos secciones demuestran cómo lidiar con los cambios de tu área de preparación y tú directorio de trabajo. Afortunadamente, el comando que usas para determinar el estado de esas dos áreas también te recuerda cómo deshacer los cambios en ellas. Por ejemplo, supongamos que has cambiado dos archivos y que quieres confirmarlos como dos cambios separados, pero accidentalmente has escrito `git add *` y has preparado ambos. ¿Cómo puedes sacar del área de preparación uno de ellos? El comando `git status` te recuerda cómo:

```
$ git add .

$ git status

On branch master
```

```
Changes to be committed:
```

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

```
renamed:    README.md -> README
```

```
modified:   CONTRIBUTING.md
```

Justo debajo del texto “Changes to be committed” (“Cambios a ser confirmados”, en inglés), verás que dice que uses `git reset HEAD <file>...` para deshacer la preparación. Por lo tanto, usemos el consejo para deshacer la preparación del archivo `CONTRIBUTING.md`:

```
$ git reset HEAD CONTRIBUTING.md
```

```
Unstaged changes after reset:
```

```
M       CONTRIBUTING.md
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

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

```
renamed:    README.md -> README
```

```
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:   CONTRIBUTING.md
```

El comando es un poco raro, pero funciona. El archivo `CONTRIBUTING.md` esta modificado y, nuevamente, no preparado.

Por ahora lo único que necesitas saber sobre el comando `git reset` es esta invocación mágica. Entraremos en mucho más detalle sobre qué hace `reset`

Deshacer un Archivo Modificado

¿Qué tal si te das cuenta que no quieres mantener los cambios del archivo `CONTRIBUTING.md`? ¿Cómo puedes restaurarlo fácilmente - volver al estado en el que estaba en la última confirmación (o cuando estaba recién clonado, o como sea que haya llegado a tu directorio de trabajo)? Afortunadamente, `git status` también te dice cómo hacerlo. En la salida anterior, el área no preparada lucía así:

```
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:   CONTRIBUTING.md
```

Allí se te indica explícitamente como descartar los cambios que has hecho. Hagamos lo que nos dice:

```
$ git checkout -- CONTRIBUTING.md

$ git status

On branch master

Changes to be committed:

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

        renamed:    README.md -> README
```

Ahora puedes ver que los cambios se han revertido.

Recuerda, todo lo que esté **confirmado** en Git puede recuperarse. Incluso **commits** que estuvieron en ramas que han sido eliminadas o **commits** que fueron sobrescritos con `--amend` pueden recuperarse Sin embargo, es posible que no vuelvas a ver jamás cualquier cosa que pierdas y que nunca haya sido confirmada.