

# Ejercicios de manejo del historial de cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios de creación y actualización de repositorios o bien hacer un clon del repositorio remoto <https://github.com/asalber/libro-git> mediante la siguiente secuencia de comandos:

```
git clone https://github.com/asalber/libro-git.git  
cd libro-git
```

Mueve la rama actual (normalmente master/main) **exactamente** al commit cuyo hash empieza por 8c808, y además sincroniza el **índice** y el **working tree** con ese commit.

Deja tu copia **tal y como estaba** en ese commit, **borrando** cualquier cambio local y commits más recientes en tu rama (seguirán en el reflog, pero ya no están en la rama).

El repositorio quedará “por detrás” del remoto, porque tú te has ido a un punto anterior de la historia.

```
git reset --hard 8c808
```

Elimina el remoto origin. Con eso tu repo queda **desconectado** de GitHub (no podrás git pull/git push accidentalmente). Útil para hacer prácticas sin tocar el repositorio original.

```
git remote remove origin
```

**Nota:** --hard **descarta** cambios no confirmados; úsalo solo si estás seguro. Si luego quisieras volver a enlazar un remoto, podrías hacerlo con git remote add origin <URL>.

## Ejercicio 1

Mostrar el historial de cambios del repositorio. Crear la carpeta capítulos y crear dentro de ella el fichero capítulo1.txt con el siguiente texto:

Git es un sistema de control de versiones ideado por Linus Torvalds.

Añadir los cambios a la zona de intercambio temporal. Hacer un commit de los cambios con el mensaje "Añadido capítulo 1." Volver a mostrar el historial de cambios del repositorio.

```
git log  
mkdir capítulos  
cat > capítulos/capítulo1.txt << 'EOF'  
Git es un sistema de control de versiones ideado por Linus Torvalds.  
EOF
```

**CMD**

(

**echo Git es un sistema de control de versiones ideado por Linus Torvalds.**  
 ) > capítulos \ capítulo1.txt

```
git add .  
git commit -m "Añadido capítulo 1."  
git log
```

## Ejercicio 2

Crear el fichero capítulo2.txt en la carpeta capítulos con el siguiente texto.

El flujo de trabajo básico con Git consiste en:

- 1- Hacer cambios en el repositorio.
- 2- Añadir los cambios a la zona de intercambio temporal.
- 3- Hacer un commit de los cambios.

Añadir los cambios a la zona de intercambio temporal. Hacer un commit de los cambios con el mensaje “Añadido capítulo 2.” Mostrar las diferencias entre la última versión y dos versiones anteriores.

```
cat > capitulo/capitulo2.txt << 'EOF'
```

El flujo de trabajo básico con Git consiste en:

- 1- Hacer cambios en el repositorio.
- 2- Añadir los cambios a la zona de intercambio temporal.
- 3- Hacer un commit de los cambios.

EOF

```
git add .
```

```
git commit -m "Añadido capítulo 2."
```

```
git diff HEAD~2..HEAD
```

## Ejercicio 3

Crear el fichero capitulo3.txt en la carpeta capitulo con el siguiente texto.

Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultanea en ellas.

Añadir los cambios a la zona de intercambio temporal. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3.” Mostrar las diferencias entre la primera y la última versión del repositorio.

```
cat > capitulo/capitulo3.txt << 'EOF'
```

Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y trabajar de manera simultanea en ellas.

EOF

```
git add .
```

```
git commit -m "Añadido capítulo 3."
```

## Ejercicio 4

Añadir al final del fichero indice.txt la siguiente línea:

Capítulo 5: Conceptos avanzados

Añadir los cambios a la zona de intercambio temporal. Hacer un commit de los cambios con el mensaje “Añadido capítulo 5 al índice.”.

Mostrar quién ha hecho cambios sobre el fichero indice.txt.

```
echo "Capítulo 5: Conceptos avanzados" >> indice.txt  
git add .  
git commit -m "Añadido capítulo 5 al índice."  
git annotate indice.txt
```

## Ejercicios de deshacer cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios o bien hacer un clon del repositorio remoto <https://github.com/asalber/libro-git> mediante la siguiente secuencia de comandos:

```
git clone https://github.com/asalber/libro-git.git  
cd libro-git  
git reset --hard 48ed8  
git remote remove origin
```

### Ejercicio 1

Eliminar la última línea del fichero indice.txt y guardarla. Comprobar el estado del repositorio. Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero. Volver a comprobar el estado del repositorio.

```
nano indice.txt # Eliminar la última línea y guardar el fichero.  
git status  
git checkout -- indice.txt  
git status
```

## Ejercicio 2

Eliminar la última línea del fichero indice.txt y guardarla. Añadir los cambios a la zona de intercambio temporal.

Comprobar de nuevo el estado del repositorio.

Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.

Comprobar de nuevo el estado del repositorio.

Deshacer los cambios realizados en el fichero indice.txt para volver a la versión anterior del fichero.

Volver a comprobar el estado del repositorio.

`nano indice.txt` *# Eliminar la última línea y guardar el fichero.*

`git add .`

`git status`

`git reset indice.txt`

`git status`

`git checkout -- indice.txt`

`git status`

## Ejercicio 3

Eliminar la última línea del fichero indice.txt y guardarla.

Eliminar el fichero capítulos/capítulo3.txt.

Añadir un fichero nuevo capítulos/capítulo4.txt vacío.

Añadir los cambios a la zona de intercambio temporal.

Comprobar de nuevo el estado del repositorio.

Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.

Comprobar de nuevo el estado del repositorio.

Deshacer los cambios realizados para volver a la versión del repositorio.  
Volver a comprobar el estado del repositorio.

```
nano indice.txt # Eliminar la última línea y guardar el fichero.  
rm capítulos/capítulo3.txt  
touch capítulos/capítulo4.txt  
git add .  
git status  
git reset  
git status  
git checkout -- .  
git status  
git clean -f  
git status
```

## Ejercicio 4

Eliminar la última línea del fichero indice.txt y guardarlo.

Eliminar el fichero capítulos/capítulo3.txt.

Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Borrado accidental.”

Comprobar el historial del repositorio.

Deshacer el último commit pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.

Comprobar el historial y el estado del repositorio. Volver a hacer el commit con el mismo mensaje de antes.

Deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.

Comprobar de nuevo el historial y el estado del repositorio.

```
nano indice.txt # Eliminar la última línea y guardar el fichero.  
rm capítulos/capítulo3.txt  
git commit -am "Borrado accidental."  
git status  
git log
```

```
git reset --soft HEAD~1
git status
git commit -m "Borrado accidental."
git status
git log
git reset --hard HEAD~1
git log
git status
```

## Ejercicios de gestión de ramas

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios o bien hacer un clon del repositorio remoto <https://github.com/asalber/libro-git> mediante la siguiente secuencia de comandos:

```
git clone https://github.com/asalber/libro-git.git
cd libro-git
git reset --hard 48ed8
git remote remove origin
```

### Ejercicio 1

Crear una nueva rama bibliografia y mostrar las ramas del repositorio.

```
git branch bibliografia
git branch -av
```

### Ejercicio 2

Crear el fichero capitulo/capitulo4.txt y añadir el texto siguiente:

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

Añadir los cambios a la zona de intercambio temporal. Hacer un commit con el mensaje “Añadido capítulo 4.”

Mostrar la historia del repositorio incluyendo todas las ramas.

```
cat > capítulos/capítulo4.txt << 'EOF'  
En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.  
EOF  
git add .  
git commit -m "Añadido capítulo 4."  
git log --graph --all --oneline
```

## Ejercicio 3

Cambiar a la rama bibliografia.

Crear el fichero bibliografia.txt y añadir la siguiente referencia:

Chacon, S. and Straub, B. Pro Git. Apress.

Añadir los cambios a la zona de intercambio temporal. Hacer un commit con el mensaje “Añadida primera referencia bibliográfica.”

Mostrar la historia del repositorio incluyendo todas las ramas.

```
git checkout bibliografia  
cat > bibliografia.txt << 'EOF'  
- Chacon, S. and Straub, B. Pro Git. Apress.  
EOF  
git add .  
git commit -m "Añadida primera referencia bibliográfica."  
git log --graph --all --oneline
```

## Ejercicio 4

Fusionar la rama bibliografia con la rama master.

Mostrar la historia del repositorio incluyendo todas las ramas.

Eliminar la rama bibliografia.

Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.

```
git checkout master  
git merge bibliografia  
git log --graph --all --oneline
```

```
git branch -d bibliografia  
git log --graph --all --oneline
```

## Ejercicio 5

Crear la rama bibliografia. Cambiar a la rama bibliografia. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:

Scott Chacon and Ben Straub. Pro Git. Apress.  
Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)

Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Añadida nueva referencia bibliográfica." Cambiar a la rama master. Cambiar el fichero bibliografia.txt para que contenga las siguientes referencias:

Chacon, S. and Straub, B. Pro Git. Apress.  
Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.

Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mismo mensaje.

Fusionar la rama bibliografia con la rama master. Resolver el conflicto dejando el fichero bibliografia.txt con las referencias:

Chacon, S. and Straub, B. Pro Git. Apress.  
Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.  
Hodson, R. Ry's Git Tutorial. Smashwords (2014)

Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Resuelto conflicto de bibliografía."

Mostrar la historia del repositorio incluyendo todas las ramas.

```
git branch bibliografia  
git checkout bibliografia  
cat > bibliografia.txt << 'EOF'  
- Scott Chacon and Ben Straub. Pro Git. Apress.  
- Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)  
EOF  
git commit -a -m "Añadida nueva referencia bibliográfica."  
git checkout master
```

```
cat > bibliografia.txt << 'EOF'  
- Chacon, S. and Straub, B. Pro Git. Apress.  
- Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.  
EOF  
git commit -a -m "Añadida nueva referencia bibliográfica."  
git merge bibliografia  
nano bibliografia.txt # Hacer los cambios indicados en el fichero  
git commit -a -m "Resuelto conflicto bibliografía."  
git log --graph --all --oneline
```

## Ejercicios de repositorios remotos

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre ramas o bien hacer un clon del repositorio remoto <https://github.com/asalber/libro-git> mediante la siguiente secuencia de comandos:

```
git clone https://github.com/asalber/libro-git.git  
cd libro-git  
git reset --hard cb1e4  
git remote remove origin
```

### Ejercicio 1

Crear un nuevo repositorio público en GitHub con el nombre libro-git. Añadirlo al repositorio local del libro. Mostrar todos los repositorios remotos configurados.

```
# Crear el repositorio en GitHub y copiar su URL con protocolo https.  
git remote add github url  
git remote -v
```

### Ejercicio 2

Añadir los cambios del repositorio local al repositorio remoto de GitHub. Acceder a GitHub y comprobar que se han subido los cambios mostrando el historial de versiones.

```
git push github master
```

## Ejercicio 3

Colaborar en el repositorio remoto libro-git de otro usuario. Clonar su repositorio libro-git. Añadir el fichero autores.txt que contenga el nombre del usuario y su correo electrónico. Añadir los cambios a la zona de intercambio temporal. Hacer un commit con el mensaje “Añadido autor.” Subir los cambios al repositorio remoto.

```
# Entrar en GitHub en el proyecto libro-git del que seamos colaboradores y copiar la URL.  
git clone url  
cat > autores.txt << 'EOF'  
# Escribir el nombre del autor y su correo.  
EOF  
git add .  
git commit -m "Añadido autor."  
git push origin master
```

## Ejercicio 4

Hacer una bifurcación del repositorio remoto asalber/libro-git en GitHub. Clonar el repositorio creado en la cuenta de GitHub del usuario. Crear una nueva rama autoria y activarla. Añadir el nombre del usuario y su correo al fichero autores.txt. Añadir los cambios a la zona de intercambio temporal. Hacer un commit con el mensaje “Añadido nuevo autor.” Subir los cambios de la rama autoria al repositorio remoto en GitHub. Hacer un Pull Request de los cambios en la rama autoria.

```
# Hacer el fork del repositorio asalber/libro-git en GitHub y copiar la URL del repositorio creado en tu cuenta.  
git clone url  
git checkout -b autoria  
# Editar con nano el fichero autores.txt y añadir el nombre y el correo electrónico del usuario en una nueva línea.  
git commit -am "Añadido nuevo autor."  
git push origin autoria  
# En GitHub: Compare & Pull Request → Create Pull Request
```

**En CMD solo cambia la forma de crear el archivo con varias líneas (el resto de comandos git son iguales).**

**El here-doc << 'EOF' no existe en CMD. Usa uno de estos métodos:**

**Opción 1 – Con paréntesis y echo (recomendada)**

(

echo El flujo de trabajo básico con Git consiste en:

echo 1- Hacer cambios en el repositorio.

echo 2- Añadir los cambios a la zona de intercambio temporal.

echo 3- Hacer un commit de los cambios.

) > capítulos\capítulo2.txt

git add .

git commit -m "Añadido capítulo 2."

git diff HEAD~2..HEAD

**Opción 2 – Con copy con (pegas el texto y cierras con Ctrl+Z)**

copy con capítulos\capítulo2.txt

El flujo de trabajo básico con Git consiste en:

1- Hacer cambios en el repositorio.

2- Añadir los cambios a la zona de intercambio temporal.

3- Hacer un commit de los cambios.

`^Z`

(luego ejecutas los mismos git add ., git commit ..., git diff ...)

**En CMD usa barras invertidas \ en rutas: capítulos \ capítulo2.txt.**