

## Desarrollo de Interfaces

## Documentación Ejercicio 3



## Ana Betsabé Cuende Tomé 2º DAM

# Índice

<b>Creación y actualización de repositorios</b>	<b>2</b>
Inicio	2
Ejercicio 3	2
Ejercicio 4	2
Ejercicio 5	3
Ejercicio 6	3
<b>Manejo del historial de cambios</b>	<b>3</b>
Ejercicio 1	3
Ejercicio 2	3
Ejercicio 3	4
Ejercicio 4	5
<b>Deshacer cambios</b>	<b>6</b>
Ejercicio 1	6
Ejercicio 2	6
Ejercicio 3	7
Ejercicio 4	8
<b>Gestión de ramas</b>	<b>9</b>
Ejercicio 1	9
Ejercicio 2	9
Ejercicio 3	10
Ejercicio 4	10
Ejercicio 5	11

# Creación y actualización de repositorios

## Inicio

Creamos un directorio que tenga por nombre “libro”, dentro de la carpeta correspondiente a nuestro repositorio y nos posicionamos dentro de dicha carpeta.

## Ejercicio 3

Creamos un archivo “.txt” que contenga el índice de nuestro libro.

```
GNU nano 7.2                               indice.txt                               Modified
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
```

Guardamos el archivo y hacemos “git add \*” para añadir los cambios.

## Ejercicio 4

Hacemos un commit para confirmar los cambios realizados.

## Ejercicio 5

Añadimos un capítulo más a nuestro índice.

```
GNU nano 7.2                               indice.txt                               Modified
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
```

Guardamos el archivo, añadimos los cambios y los confirmamos.

## Ejercicio 6

Mediante el comando “--amend” cambiamos el nombre del commit.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git commit --amend -m "Añadido capítulo 3 sobre gestión de ramas al índice"
[main 05d9486] Añadido capítulo 3 sobre gestión de ramas al índice
Date: Tue Sep 19 10:11:06 2023 +0200
1 file changed, 2 insertions(+), 1 deletion(-)

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git show
commit 05d94860ec4498c487025238509bfd7a73ac7aa1 (HEAD -> main)
Author: anacuende <acuendet@fpcoruna.afundacion.org>
Date: Tue Sep 19 10:11:06 2023 +0200

    Añadido capítulo 3 sobre gestión de ramas al índice
```

# Manejo del historial de cambios

## Ejercicio 1

Creamos la carpeta “capítulos” dentro de nuestra carpeta “libro” y dentro de ella creamos el archivo de texto “capitulo1.txt”.

```
GNU nano 7.2                                capitulo1.txt                                Modified
Git es un sistema de control de versiones ideado por Linus Torvalds|
```

Añadimos los cambios y los confirmamos.

## Ejercicio 2

Creamos el archivo “capitulo2.txt” dentro de la carpeta capítulos.

```
GNU nano 7.2                                capitulo2.txt                                Modified
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.
```

Añadimos los cambios y los confirmamos. Con el comando “git diff HEAD~2..HEAD” comparamos las diferencias entre la última versión y dos versiones anteriores.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git diff HEAD~2..HEAD
diff --git a/libro/capitulos/capitulo1.txt b/libro/capitulos/capitulo1.txt
new file mode 100644
index 0000000..cdced15
--- /dev/null
+++ b/libro/capitulos/capitulo1.txt
@@ -0,0 +1 @@
+Git es un sistema de control de versiones ideado por Linus Torvalds
diff --git a/libro/capitulos/capitulo2.txt b/libro/capitulos/capitulo2.txt
new file mode 100644
index 0000000..0070310
--- /dev/null
+++ b/libro/capitulos/capitulo2.txt
@@ -0,0 +1,4 @@
+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.
```

## Ejercicio 3

Añadimos el archivo “capitulo3.txt” en la carpeta “libros”.

```
GNU nano 7.2                                capitulo3.txt                                Modified
Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto
y trabajar de manera simultánea en ellas.
```

Añadimos los cambios y los confirmamos. Mostramos las diferencias entre la primera y la última versión del repositorio con el comando “git diff <codigo hash de la primera version>..HEAD”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git diff 21f6ee..HEAD
diff --git a/libro/capitulos/capitulo1.txt b/libro/capitulos/capitulo1.txt
new file mode 100644
index 0000000..cdced15
--- /dev/null
+++ b/libro/capitulos/capitulo1.txt
@@ -0,0 +1 @@
+Git es un sistema de control de versiones ideado por Linus Torvalds
diff --git a/libro/capitulos/capitulo2.txt b/libro/capitulos/capitulo2.txt
new file mode 100644
index 0000000..0070310
--- /dev/null
+++ b/libro/capitulos/capitulo2.txt
@@ -0,0 +1,4 @@
+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.
diff --git a/libro/capitulos/capitulo3.txt b/libro/capitulos/capitulo3.txt
new file mode 100644
index 0000000..bcbee16
--- /dev/null
+++ b/libro/capitulos/capitulo3.txt
@@ -0,0 +1 @@
+Git permite la creación de ramas lo que permite tener distintas versiones del mismo proyecto y
trabajar de manera simultánea en ellas.
diff --git a/libro/indice.txt b/libro/indice.txt
new file mode 100644
index 0000000..ba17d35
--- /dev/null
+++ b/libro/indice.txt
@@ -0,0 +1,4 @@
+Capítulo 1: Introducción a Git
+Capítulo 2: Flujo de trabajo básico
+Capítulo 3: Gestión de ramas
+Capítulo 4: Repositorios remotos
```

## Ejercicio 4

Nos posicionamos en la carpeta “libro” y allí ejecutamos el comando “echo “Capítulo 5: Conceptos avanzados” >> indice.txt”, así añadiremos esa línea que especificamos entre comillas al archivo “indice.txt” sin necesidad de abrirlo. Aquí vemos que se ejecutó correctamente.

```
GNU nano 7.2      indice.txt
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
Capítulo 5: Conceptos avanzados
```

Añadimos los cambios y los confirmamos. Con el comando “git annotate <nombre archivo>” mostramos quién ha hecho los cambios en dicho archivo.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git annotate indice.txt
9b7d5e96      ( anacuende      2023-09-19 10:06:22 +0200      1)Capítulo 1: Introducción a Gi
t
9b7d5e96      ( anacuende      2023-09-19 10:06:22 +0200      2)Capítulo 2: Flujo de trabajo
básico
05d94860      ( anacuende      2023-09-19 10:11:06 +0200      3)Capítulo 3: Gestión de ramas
05d94860      ( anacuende      2023-09-19 10:11:06 +0200      4)Capítulo 4: Repositorios remo
tos
bbfd932d      ( anacuende      2023-09-20 09:22:20 +0200      5)Capítulo 5: Conceptos avanzad
os
```

## Deshacer cambios

### Ejercicio 1

Abrimos el archivo “índice.txt”, situado en la carpeta “libro” y borramos la última línea.

```
GNU nano 7.2                               índice.txt                               Modified
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
```

Lo guardamos y mediante el comando “git status” vemos que el archivo aparece como “modificado”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   índice.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Después, con el comando “git checkout -- <nombre archivo>” deshacemos los cambios realizados para volver a una versión anterior. Comprobamos que se ha realizado correctamente lanzando “git status”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git checkout -- índice.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

## Ejercicio 2

Volvemos a acceder al archivo “indice.txt”, volvemos a borrar la última línea y guardamos el archivo.

```
GNU nano 7.2                               indice.txt                               Modified
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
```

Añadimos los cambios y con el comando “git reset <nombre archivo>” eliminamos los cambios realizados de la zona temporal. Con un “git status” comprobamos que se realizó correctamente. Luego hacemos “git checkout -- <nombre archivo>” para volver a la versión anterior del fichero y comprobamos con otro “git status” que todo se ejecutó con éxito.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git reset indice.txt
Unstaged changes after reset:
M      libro/indice.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   indice.txt

no changes added to commit (use "git add" and/or "git commit -a")

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git checkout -- indice.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

## Ejercicio 3

Volvemos a eliminar la última línea del archivo “indice.txt”.

```
GNU nano 7.2                               indice.txt                               Modified
Capítulo 1: Introducción a Git
Capítulo 2: Flujo de trabajo básico
Capítulo 3: Gestión de ramas
Capítulo 4: Repositorios remotos
```

Accedemos a la carpeta “capitulos” y desde ahí borramos el archivo correspondiente al capítulo 3 mediante el comando “rm capitulo3.txt” y añadimos el capítulo 4 mediante el comando “touch capitulo4.txt”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ cd capitulos/

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ dir
capitulo1.txt  capitulo2.txt  capitulo3.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ rm capitulo3.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ touch capitulo4.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ dir
capitulo1.txt  capitulo2.txt  capitulo4.txt
```

Añadimos los cambios y hacemos “git reset” para eliminarlos de la zona de intercambio temporal y luego “git checkout –.”

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git checkout -- .

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ../indice.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    capitulo4.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Con “git clean -f” deshacemos los cambios realizados.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git clean -f
Removing capitulo4.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ../indice.txt

no changes added to commit (use "git add" and/or "git commit -a")
```



## Ejercicio 4

Borramos la última línea del archivo “indice.txt”. Después accedemos a la carpeta “capitulos” y borramos el archivo correspondiente al capítulo 3. Confirmamos los cambios con “git commit -a “Borrado accidental””.

Comprobamos con git status que todo se realizó correctamente y eliminamos

el commit sin borrar los cambios con “git reset soft HEAD~1”.

Después de comprobar que todo se realizó correctamente con “git status” hacemos un “git commit -m “Borrado accidental””. Por último, para deshacer el commit y los cambios hacemos “git reset --hard HEAD~1” y comprobamos que todo está correcto con “git status”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git reset --hard HEAD~1
HEAD is now at bbfd932 Añadido capítulo 5 al índice

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

## Gestión de ramas

### Ejercicio 1

Creamos una nueva rama “bibliografia” con el comando “git branch <nombre rama>” y mostramos todas las ramas de repositorio con el comando “git branch -av”.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git branch bibliografia

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git branch -av
 bibliografia          bbfd932 Añadido capítulo 5 al índice
* main                bbfd932 [ahead 6] Añadido capítulo 5 al índice
 remotes/origin/main  21f6ee1 Commit inicial
```

### Ejercicio 2

Nos posicionamos en la carpeta “capitulos” y creamos el archivo “capitulo4.txt”.

```
GNU nano 7.2          capitulo4.TXT          Modified
En este capítulo varemos cómo usar GitHub para alojar repositorios en remoto.:
```

Añadimos y confirmamos los cambios y después con el comando “git log --graph --all --oneline” visualizamos la historia del repositorio, incluyendo las ramas.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro/capitulos
$ git lg --graph --all --oneline
* 8b6c4d8 (HEAD -> main) Añadido capítulo 4
* bbfd932 (bibliografia) Añadido capítulo 5 al índice
* 6a48c94 Añadido capítulo 3
* be4e76a Añadido capítulo 2
* 5957ea8 Añadido capítulo 1
* 05d9486 Añadido capítulo 3 sobre gestión de ramas al índice
* 9b7d5e9 Añadido índice del libro
* 21f6ee1 (origin/main) Commit inicial
```

### Ejercicio 3

Nos posicionamos en la rama “bibliografía” con el comando “git checkout bibliografia” y en la carpeta “libro” creamos el archivo “bibliografia.txt”.

```
GNU nano 7.2 bibliografia.txt Modified
- Chacon, S. and Strub, B, Pro git. Apress.:
```

Añadimos y confirmamos los cambios y por último volvemos a visualizar la historia del repositorio.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git lg --graph --all --oneline
* b2b254a (HEAD -> bibliografia) Añadida primera referencia bibliográfica
| * 8b6c4d8 (main) Añadido capítulo 4
|/
* bbfd932 Añadido capítulo 5 al índice
* 6a48c94 Añadido capítulo 3
* be4e76a Añadido capítulo 2
* 5957ea8 Añadido capítulo 1
* 05d9486 Añadido capítulo 3 sobre gestión de ramas al índice
* 9b7d5e9 Añadido índice del libro
* 21f6ee1 (origin/main) Commit inicial
```

### Ejercicio 4

Cambiamos a la rama principal “main”, mergeamos la rama “bibliografía” a main con “git checkout main”. Visualizamos el historial del repositorio y borramos la rama “bibliografia” con “git branch -d bibliografia”. Por último volvemos a visualizar el historial del repositorio.

```
Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 7 commits.
(use "git push" to publish your local commits)

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git merge bibliografia
Merge made by the 'ort' strategy.
 libro/bibliografia.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 libro/bibliografia.txt

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git lg --graph --all --oneline
* 0eb134e (HEAD -> main) Merge branch 'bibliografia'
|
| * b2b254a (bibliografia) Añadida primera referencia bibliográfica
| * 8b6c4d8 Añadido capítulo 4
|/
* bbfd932 Añadido capítulo 5 al índice
* 6a48c94 Añadido capítulo 3
* be4e76a Añadido capítulo 2
* 5957ea8 Añadido capítulo 1
* 05d9486 Añadido capítulo 3 sobre gestión de ramas al índice
* 9b7d5e9 Añadido índice del libro
* 21f6ee1 (origin/main) Commit inicial

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git branch -d bibliografia
Deleted branch bibliografia (was b2b254a).

Alumno@DESKTOP-N6F2RNG MINGW64 ~/DI/libro
$ git lg --graph --all --oneline
* 0eb134e (HEAD -> main) Merge branch 'bibliografia'
|
| * b2b254a Añadida primera referencia bibliográfica
| * 8b6c4d8 Añadido capítulo 4
|/
* bbfd932 Añadido capítulo 5 al índice
* 6a48c94 Añadido capítulo 3
* be4e76a Añadido capítulo 2
* 5957ea8 Añadido capítulo 1
* 05d9486 Añadido capítulo 3 sobre gestión de ramas al índice
* 9b7d5e9 Añadido índice del libro
* 21f6ee1 (origin/main) Commit inicial
```

## Ejercicio 5

Volvemos a crear la rama “bibliografia” y nos posicionamos en ella como vimos anteriormente. En el archivo “bibliografía.txt” escribimos una nueva línea.

```
GNU nano 7.2 bibliografia.txt Modified
- Chacon, S. and Strub, B, Pro git. Apress.
- Ryan Hodson. Ry's Git Tutorial. Smashwords (2014)
```

Confirmamos los cambios y nos posicionamos en la rama “main” y abrimos el archivo “bibliografía.txt” en el que añadiremos una nueva línea.

```
GNU nano 7.2 bibliografia.txt Modified
- Chacon, S. and Strub, B, Pro git. Apress.
- Loeliger, J. and McCullough, M. Version control with Git. O'Reilly
```

Confirmamos los cambios y mergeamos la rama “bibliografia” a “main” como hicimos anteriormente. Se genera un conflicto que nos permite realizar el “merge”. Editamos el archivo “bibliografia.txt” para resolver el conflicto.

```
GNU nano 7.2 bibliografia.txt Modified
- 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)
```

Nos posicionamos en la rama “main” y mergeamos la rama “bibliografia”, ahora ya no hay conflictos. Confirmamos los cambios y visualizamos por última vez el historial. Por último hacemos un “git push” a nuestro repositorio.