

Ejercicios de Git: Creación y actualización de repositorios

Ejercicio 1

Mostrar la configuración de usuario y email (git config y algo más)

```
git config --list
```

Ejercicio 2

Crear un repositorio nuevo con el nombre `libro`.

```
mkdir libro  
cd libro  
git init
```

Ejercicio 3

1. Comprobar el estado del repositorio.

```
git status
```

2. Crear un fichero `indice.txt` con el siguiente contenido:

```
Capítulo 1: Introducción a Git  
Capítulo 2: Flujo de trabajo básico
```

```
vim indice.txt
```

3. Añadelo a la staging área

```
git add .
```

4. Nos hemos olvidado de añadir una línea, sácalo de la staging área

```
git rm --cached indice.txt
```

5. Añade la siguiente línea al archivo

```
Capítulo 3: Repositorios remotos
```

```
vim indice.txt
```

6. Comprobar de nuevo el estado del repositorio.

```
git status
```

7. Añadir el fichero a la zona de intercambio temporal (staging área)

```
git add .
```

8. Volver a comprobar una vez más el estado del repositorio.

```
git status
```

Ejercicio 4

Realizar un commit de los últimos cambios con el mensaje “Añadido índice del libro.” y ver el estado del repositorio.

```
git commit -m «Añadido índice del libro»  
git status
```

Ejercicio 5

1. Cambiar el fichero `indice.txt` para que contenga lo siguiente:

```
Capítulo 1: Introducción a Git  
Capítulo 2: Flujo de trabajo básico  
Capítulo 3: Gestión de ramas  
Capítulo 4: Repositorios remotos
```

```
vim indice.txt
```

2. Mostrar los cambios con respecto a la última versión guardada en el repositorio.

```
git diff HEAD
```

3. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3 sobre gestión de ramas”.

```
git add .  
git commit -m «Añadido capítulo 3 sobre gestión de ramas»
```

Ejercicio 6

1. Mostrar los cambios de la última versión del repositorio con respecto a la anterior.

```
git diff HEAD~1
```

2. Cambiar el mensaje del último commit por "Añadido capítulo 3 sobre gestión de ramas al índice."

`git commit --amend -m «Añadido capítulo 3 sobre gestion de ramas al indice»`

3. Volver a mostrar los últimos cambios del repositorio.

`git diff HEAD~1`