

Ejercicios de Git

Ejercicio 1 — Configurar Git y verificar

Pasos

1. Define nombre, email y color en la salida.
2. Lista la configuración final.

```
git config --global user.name "Tu Nombre Completo"  
git config --global user.email "tu-correo@ejemplo.com"  
git config --global color.ui auto  
git config --list
```

Ejercicio 2 — Crear repo libro y listar contenido

Pasos

1. Crea carpeta y entra.
2. Inicializa Git.
3. Muestra archivos (incluye ocultos).

```
mkdir libro  
cd libro  
git init  
ls -la      # (en Windows PowerShell: ls; en CMD: dir /a)
```

Ejercicio 3 — Estado, crear indice.txt, añadir al stage y verificar

Pasos

1. Ver estado inicial.
2. Crear indice.txt con el contenido indicado.
3. Ver estado (aparece como "untracked").
4. Añadir al stage.
5. Ver estado (ahora "staged").

```
git status
```

```
cat > indice.txt <<'TXT'
```

```
Capítulo 1: Introducción a Git
```

```
Capítulo 2: Flujo de trabajo básico
```

```
Capítulo 3: Repositorios remotos
```

```
TXT
```

```
# En Windows CMD: usa notepad indice.txt (guardar y cerrar)
```

```
git status
```

```
git add indice.txt
```

```
git status
```

Ejercicio 4 — Commit y comprobar estado

Pasos

1. Haz commit con mensaje.
2. Comprueba que el árbol está limpio.

```
git commit -m "Añadido índice del libro."
```

```
git status
```

Ejercicio 5 — Modificar indice.txt, ver diff y commitear

Pasos

1. Reemplaza el contenido de indice.txt.
2. Muestra diferencias con respecto a HEAD.
3. Añade al stage y commitea.

```
cat > indice.txt <<'TXT'
```

```
Capítulo 1: Introducción a Git
```

```
Capítulo 2: Flujo de trabajo básico
```

```
Capítulo 3: Gestión de ramas
```

```
Capítulo 4: Repositorios remotos
```

```
TXT
```

```
# Windows CMD: editar con notepad indice.txt
```

```
git diff
```

```
git add indice.txt
```

```
git commit -m "Añadido capítulo 3 sobre gestión de ramas"
```

Ejercicio 6 — Ver último cambio, enmendar mensaje y volver a ver

Pasos

1. Mostrar el último commit (diff + metadatos).
2. Cambiar el mensaje del último commit (sin crear uno nuevo).
3. Volver a mostrar el commit enmendado.

```
git show
```

```
git commit --amend -m "Añadido capítulo 3 sobre gestión de ramas al índice."
```

```
git show
```