

EJERCICIOS INTERACTIVOS – GIT RAMAS, MERGE Y CONFLICTOS

1. Crear rama feature-x, editar perfil.py, volver a main

Analogía: Crear **una** copia del cuaderno para probar **una** nueva idea **sin** estropear **el** original.

```
git checkout -b feature-x          # Creamos y entramos en la nueva rama
nano perfil.py                     # Editamos el archivo (puedes usar cualquier editor)
git add perfil.py
git commit -m "Cambios en rama feature-x"
git checkout main                  # Volvemos a la rama principal
```

2. Fusionar feature-x en main sin conflicto

Analogía: Combinar dos versiones **de** un documento sin **que** nadie haya escrito **en la** misma línea.

```
git merge feature-x               # Fusionamos la rama al main
```

Si no hay conflicto, Git fusiona automáticamente.

3. Crear conflicto modificando misma línea en ambas ramas

Analogía: Dos personas editan **el** mismo **párrafo** de un archivo, **y** Git **no** sabe **cuál** elegir.

En rama main

```
git checkout main
nano perfil.py                     # Cambia una línea (por ejemplo, el print)
git commit -am "Cambio en main"
```

En rama feature-x

```
git checkout feature-x
nano perfil.py                     # Cambia la misma línea que antes
git commit -am "Cambio en feature-x"
```

Volvemos a main e intentamos fusionar

```
git checkout main
git merge feature-x               # Aquí aparece el conflicto
```

4. Resolver conflicto y hacer commit de fusión

Analogía: Git **te** pide **que** decidas entre dos versiones **que se** pisan. Tú eres el juez.

```
<<<<<< HEAD
print("Versión en main")
=====
print("Versión en feature-x")
>>>>>> feature-x
```

✂ Borra las marcas <<<<<<, =====, >>>>>> y deja solo la versión correcta.

```
git add perfil.py
git commit -m "Conflicto resuelto entre main y feature-x"
```

5. Crear 2 ramas hijas y fusionarlas antes de volver a main

Analogía: Dos ramas salen del tronco del árbol. Las unimos entre ellas antes de volver al tronco.

```
git checkout -b ramaA
```

Hacer cambios y commit en ramaA

```
git add .
git commit -m "Cambios en ramaA"
git checkout main
git checkout -b ramaB
```

Hacer cambios y commit en ramaB

```
git add .
git commit -m "Cambios en ramaB"
```

Fusionar ramas hijas entre sí

```
git checkout ramaA
git merge ramaB # Fusionamos ramaB dentro de ramaA
```

Volver a main y fusionar resultado final

```
git checkout main
git merge ramaA
```

FRASE PARA RECORDAR

“En Git, cada rama es **una** idea paralela. Saber **cuándo y cómo** unir las es **lo** que convierte caos **en** colaboración.”