

RAMAS EN GITHUB

¿Qué son las ramas en GitHub?

Las ramas (*branches*) en GitHub nos permiten trabajar en versiones paralelas de un proyecto sin afectar el código principal. Son muy útiles para desarrollar nuevas funcionalidades, corregir errores o probar bloques de código y experimentar sin modificar el proyecto principal (este suele estar en la rama **main** preparado para producción).

Ejemplo de ramas en un proyecto

En un proyecto de desarrollo de aplicaciones web, se suelen utilizar las siguientes ramas:

- **main o master:** Contiene la versión final del código que se sube a producción.
- **develop:** Rama de desarrollo donde se integran los cambios antes de pasar a producción.
- **feature/nombre-funcionalidad:** Ramas creadas para desarrollar nuevas funcionalidades. Puede haber más de una.
- **bugfix/arreglo+error:** Se usa para corregir errores detectados en el código y que quede registrada su solución.

Comandos básicos para ramas (*branches*)

Crear una rama

```
Unset  
git branch nombre-rama
```

Cambiar de rama

```
Unset  
git checkout nombre-rama
```

Crear y cambiar de rama (ambos comandos combinados). Esto crea y cambia a la nueva rama en un solo paso

```
Unset  
git checkout -b nombre-rama
```

Mostrar todas las ramas

Unset

```
git branch -d nombre-rama
```

Fusionar ramas (*merge*)

Una vez completado el trabajo en una rama concreta hay que fusionarla con otra. Por ejemplo, si es un arreglo de un error, se fusionará con *develop*, si es un cambio en *develop* y tiene que pasar a producción, se fusionará con *main*.

Ejemplo de fusionar una rama con ***main***:

Ir a la rama donde queremos hacer la fusión. En este caso, ir a la rama ***main***

Unset

```
git checkout main
```

Actualizar con los cambios más recientes con *git pull*

Unset

```
git pull origin main
```

Fusionar la rama de trabajo que queremos añadir a ***main***

Unset

```
git merge nombre-rama
```

Si todo está correcto, subir los cambios

Unset

```
git push origin main
```

Solucionar errores al fusionar ramas

A veces, pueden ocurrir conflictos al fusionar ramas. Para resolverlos hay que leer bien toda la información que nos da Git.

1. Git indicará los archivos con conflictos para poder solucionarlos.
2. Se pueden editar manualmente los archivos desde VSC y hay que decidir qué cambios conservar y cuáles desechar de esos que están en conflicto.
3. Una vez resueltos los conflictos, se marcan los archivos como solucionados o resueltos y se añaden al repositorio

Unset

```
git add archivo-con-conflicto
```

4. Comitear la fusión

Unset

```
git commit -m "mensaje"
```

5. Por último, subir los cambios al repositorio remoto indicando la rama principal

Unset

```
git push origin main
```