



## **PROCESO DE DESARROLLO**

### **Objetivos**

- Trabajar con repositorios remotos en Git.
- Utilizar flujos de trabajo que involucren repositorios remotos y locales.

### **Recursos**

#### **Vídeos:**

[Repositorios remotos](#)

[Fetch y push](#)

[Ciclo de trabajo local y remoto](#)

[Ciclo de trabajo: fetch, merge y push](#)

[Pull](#)

[Clone](#)

#### **Referencias:**

<https://git-scm.com/doc>

### **Pasos a seguir**

1. Realiza los siguientes pasos:
  - a. Crea un repositorio remoto en Github.
  - b. Sube tu proyecto al repositorio remoto.
  - c. Desarrolla una funcionalidad que implique al menos un par de cambios en local y súbelos al remoto.
  - d. Realiza otra funcionalidad y sube los cambios.
  - e. Comprueba que en el repositorio remoto esta la versión correcta.
2. Repite los apartados c y d del ejercicio anterior, pero entre la primera y la segunda funcionalidad, modifica algunos ficheros desde Github (simulando que otro compañero los ha cambiado).
3. Ahora trabajas en una funcionalidad que implica 10 commits. Cada dos commits, modifica algunos ficheros desde Github (simulando que otro compañero los ha cambiado) y ves incorporando estos cambios en tu rama de funcionalidad. Cuando termines, intégralo todo en master.



# Florida

Ciclo

2ºDAW

Módulo DAW

Bloque 0

Actividad

Tarea 3

Nombre

4. Por grupos de 3 o 4 personas, realizad los siguientes pasos:
  - a. Cread un repositorio en Github y empezad a trabajar cada uno en una funcionalidad que implique el fichero index.html. Cada uno pondrá un enlace justo después del título.
  - b. Al final, deberá quedar todo integrado en la rama master del repositorio remoto.
  - c. Reflexionar sobre el proceso realizado y las decisiones tomadas como equipo.
5. Por grupos de 3 o 4 personas, realizad:
  - a. Cread un repositorio en Github con permisos de escritura para todos y empezad a trabajar cada uno en una funcionalidad que implique ficheros diferentes y varios commits.
  - b. En el servidor habrá dos ramas: master y develop. La rama master contendrá la última versión del código listo para subir a producción, mientras que la rama develop contendrá la última versión en desarrollo con las funcionalidades que se incluirán en la siguiente versión.
  - c. Las integraciones de las funcionalidades se realizarán en develop, etiquetando cada funcionalidad como una versión. El etiquetado seguirá el formato dev-0.0 para la versión inicial, y dev-0.X para el resto de funcionalidades.
  - d. Cuando esté todo integrado en develop, se integrará en master, asumiendo que se han realizado las pruebas y todo está correcto. Se debe etiquetar la versión estable como v-1.0.
  - e. Reflexionad sobre el proceso realizado y las decisiones tomadas como equipo.