

## Actividad: Control de versiones utilizando Git

**Objetivo:** utilizar Git y Github para realizar control de versiones de un software.

A continuación encontrará algunos comandos básicos de Git y una serie de actividades que deberán realizar. Para consultar más información sobre Git revisar [-1-](#).



## ACTIVIDADES:

1. En grupos de 4 van a realizar un proyecto colaborativo, utilizar cualquier proyecto que tengan de APO.
2. Crear o utilizar un repositorio montado en Github.
3. Crear una rama dev y cada uno clonar el repositorio (revisar [-2-](#)), para tenerlo de forma local.
4. El nuevo reléase tiene 4 funcionalidades nuevas, cada uno va a simular que tiene un nuevo feature que realizar, para ello crear una rama `feature#n` [1,2,3,4] y en el código cada uno realizar un nuevo método, simulando una nueva funcionalidad (para ayuda de cómo crear ramificaciones, y fusionarlas revisar [-3-](#)).
5. El que estaba realizando la funcionalidad 1 (feature#1) no alcanzó a terminarla y le solicitaron corregir un bug, entonces debe crear una rama hotfix para corregir un bug e integrarlo en master. (La rama hotfix debería salir de master o desde feature#1?)
6. Ahora es necesario fusionar todo en dev, y luego de organizar los conflictos montarlo en master.
7. Analizar con qué número de versión debería quedar cada versión del software, en el caso de la corrección del error y en el caso de los nuevos features que corresponden a un nuevo reléase.

Links de interés:

- 1 Fundamentos de Git <https://git-scm.com/book/es/v2/Fundamentos-de-Git-Obteniendo-un-repositorio-Git>
- 2 Clonar repositorio <https://docs.github.com/es/repositories/creating-and-managing-repositories/cloning-a-repository>
- 3 Ramificar y fusionar <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-Procedimientos-B%C3%A1sicos-para-Ramificar-y-Fusionar>