

1. Crear una carpeta de trabajo local.  
**Capturar imagen**
2. Crear un repositorio público en GITHUB, recomendable que tenga el mismo nombre que la carpeta local.  
**Capturar imagen**
3. Una vez creado el repositorio, en GitHub se dan instrucciones para inicializarlo desde una carpeta de trabajo, **SEGUIR LAS INSTRUCCIONES...or create a new repository on the command line**  
**Capturar imagen**
4. Crear rama develop desde master y subirla al repositorio remoto.  
**Capturar imagen**
5. Crear un fichero en develop y subirlo al repositorio remoto. Comprobar que el fichero se ha creado en la rama develop y no en master.  
**Capturar imagen**
6. Cambiar a la rama master, hacer merge desde develop y subir cambios al repositorio. Comprobar que se ha subido el fichero también a master.  
**Capturar imagen**
7. Crear un tag con la versión actual en master.  
**Capturar imagen**
8. Crear **desde develop** dos ramas, **dev1 y dev2** y subirlas al repositorio. Comprobar la configuración del fichero config ha cambiado.  
**Capturar imagen**
9. En la rama dev1 modificar una línea del fichero subido anteriormente y subirlo al repositorio remoto.  
**Capturar imagen**
10. Hacer merge de dev1 en develop. Comprobar en Github que dev1 y develop coinciden.  
**Capturar imagen**
11. Hacer checkout de la rama dev2 y comprobar que el fichero es el original y no contiene la modificación de dev1.  
**Capturar imagen**
12. De la rama dev2, modificar la misma línea del fichero que se cambió anteriormente en dev1, pero con un cambio diferente y subir el cambio al repositorio. Comprobar en Github el cambio.  
**Capturar imagen**
13. Hacer merge de dev2 en develop. Observar que se obtiene un mensaje de que existen conflictos.  
**Capturar imagen**

14. Editar el fichero modificado y observar cómo se han marcado las líneas en conflicto.

***Capturar imagen***

15. Rama develop, modificar el fichero eliminando los marcadores de conflicto y dejando la línea en la forma definitiva y subirlo al repositorio. Comprobar que en la rama develop se ha subido el fichero con el conflicto resuelto.

***Capturar imagen***

16. Cambiar a la rama dev1 y modificar nuevamente la misma línea del fichero

***Capturar imagen***

17. Intentar cambiar a dev2. Observar el error y comprobar que se sigue en dev1.

***Capturar imagen***

18. Hacer stash para almacenar los cambios pendientes.

***Capturar imagen***

19. Cambiar a dev2 y comprobar que ahora sí se cambia de rama.

***Capturar imagen***

20. Recuperar el fichero de stash. Comprobar que se genera un conflicto.

***Capturar imagen***

21. Editar el fichero y resolver el conflicto. Subir el fichero al repositorio remoto dev2.

***Capturar imagen***

22. Volver a la rama develop y hacer merge de dev2. Comprobar que se genera un nuevo conflicto.

***Capturar imagen***

23. Resolver conflicto y subir develop al repositorio remoto.

***Capturar imagen***

24. Volver a dev1 y comprobar que el stash sigue en la lista. Recuperar el stash en dev1.

***Capturar imagen***

25. Obtener diferencias entre master y develop, guardarlas en un fichero .diff y subirlas a develop.

***Capturar imagen***

26. Hacer merge de develop a master y subir al repositorio.

***Capturar imagen***

27. Crear una carpeta Log y un fichero de texto dentro de la carpeta.

***Capturar imagen***

28. Crear un fichero .gitignore en la carpeta de trabajo e incluir dentro una línea con /Log/ → nombre de la nueva carpeta creada.

***Capturar imagen***

29. Comprobar con git status que el contenido del Log no se tiene en cuenta a la hora de hacer el commit.

***Capturar imagen***