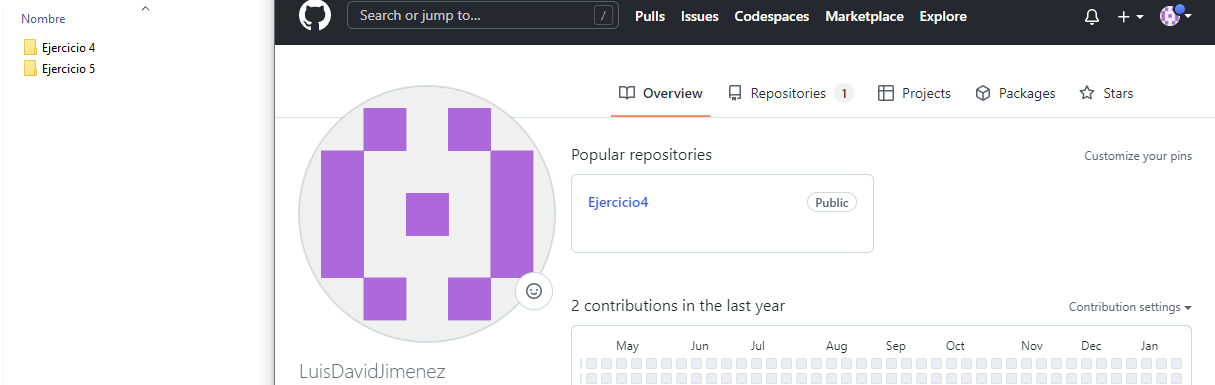
1. Crear una carpeta de trabajo local.

# Capturar imagen

****

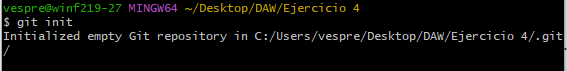
1. Crear un repositorio **público en GITHUB**, recomendable que tenga el mismo nombre que la carpeta local.

# Capturar imagen

****

1. 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

******

1. Crear rama develop desde master y subirla al repositorio remoto.

## Capturar imagen

1. 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

1. 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

1. Crear un tag con la versión actual en master.

## Capturar imagen

1. Crear **desde develop** dos ramas, ***dev1 y dev2*** y subirlas al repositorio. Comprobar la configuración del fichero config ha cambiado.

## Capturar imagen

1. En la rama dev1 modificar una línea del fichero subido anteriormente y subirlo al repositorio remoto.

## Capturar imagen

1. Hacer merge de dev1 en develop. Comprobar en Github que dev1 y develop coinciden.

## Capturar imagen

1. Hacer checkout de la rama dev2 y comprobar que el fichero es el original y no contiene la modificación de dev1.

## Capturar imagen

1. 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

1. Hacer merge de dev2 en develop. Observar que se obtiene un mensaje de que existen conﬂictos.

## Capturar imagen

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

## Capturar imagen

1. 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

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

## Capturar imagen

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

## Capturar imagen

1. Hacer stash para almacenar los cambios pendientes.

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

1. Resolver conﬂicto y subir develop al repositorio remoto.

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen

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

## Capturar imagen