**Crear un repositorio:**

Se hace desde la página de GitHub y se le pone un nombre.

**Unirnos a un repositorio ajeno:**

**git clone**

**Enlazar el proyecto con el repositorio:**

Vamos a la carpeta del proyecto y click derecho > (GIT Bash Here), y así ya te aparece la carpeta de proyecto localizada. Entonces hacemos el comando: **git init** que sirve para iniciar un repositorio git en local de un proyecto.

Con git add . añadimos los cambios realizados en nuestro proyecto al commit que realizaremos después.

Y para confirmar los cambios, haremos **git commit -m “Primer commit”**

**\*PARA GUARDAR LOS CAMBIOS REALIZADOS EN UN COMMIT CON UNA DESCRIPCION**

Aún así aún no le hemos dicho a que repositorio se va a enlazar, eso lo hacemos con el comando **git remote add origin** [**https://enlace**](https://enlace)**........**

Una vez enlazado, debemos SUBIR todos los archivos que tenemos a dicho repositorio usando el comando **git push -u origin master** (diciendole que suba todo lo que hay desde origen a la rama master del repositorio).

\_\_

Si quisiéramos subir los cambios al repositorio, debemos hacer otra vez:

**git add .**

**git commit -m “Comentario”**

**git push -u origin master**

**git fetch <repositorio>:** permite recuperar todos los ficheros de un repositorio remoto y los ubica en una rama oculta de nuestro repositorio local (sin fusionarlos). Si quisiéramos fusionarlo habría que usar git merge después.

PARA VER SI HA HABIDO CAMBIOS EN EL REPOSITORIO REMOTO QUE NO TENEMOS EN NUESTRO REPOSITORIO LOCAL

**git pull** es como escribir git fetch y git merge en un solo comando. La ventaja es que no tenemos que escribir un segundo comando, y la desventaja es que puede ocurrirte algún disgusto por accidente.

GIT PULL ES PARA BAJAR DEL REPOSITORIO REMOTO LOS CAMBIOS QUE NO TENEMOS EN LOCAL.

MERGE ES PARA COMPATIBILIZAR POR EJEMPLO DOS RAMAS.