Primero, creamos un nuevo repositorio en GitHub y copiamos la URL del repositorio, vamos a la consola de Git y usamos git remote add origin URL. Luego, si ponemos git remote y con git remote --v, vemos que se ha guardado correctamente la URL.

Con git push origin master enviamos todo lo que tenemos en la rama master a origin. ANTES DE HACER ESTO TENEMOS QUE TRAERNOS TODO LO QUE HAY EN EL REPOSITORIO REMOTO CON git pull origin master. Si hay historias no relacionadas no será exitoso, se soluciona con: git pull origin master --allow-unrelated-histories (Resuelve el conflicto de historias no relacionadas entre ambos repositorios)

Las llaves SSH no se hacen por repositorio o proyecto, sino por persona.

En todo momento que haga cambios, tengo que hacer un pull para traerme los cambios más recientes del repositorio:

git pull origin master → Fusiona los cambios del origin con master

git push origin master → Fusiona los cambios del master con origin

Podemos crear alias de comandos muy largos como, por ejemplo: alias arbol="git log --all --graph --decorate --oneline”. Cada vez que pongamos arbol, se ejecutará ese comando.

Podemos agregar un tag de manera local a una versión de los commits con el comando git tag –a *nombre* –m “Mensaje” c*ódigoDeReferencia.*

Si queremos enviar estos tags al repositorio remoto, podemos hacer lo con un push origin –tags.

Se puede borrar un tag con el comando git tag –d ***nombre***. Después, usamos git push origin :refs/tags/*nombreDelTag* lo borraremos también del servidor remoto.

Se puede ver la historia de una rama con git show-branch y git show-branch --all

Con gitk se nos abre una interfaz gráfica que nos muestra lo mismo del comando anterior.

Para unirme al proyecto de otro, tengo que copiar el enlace de clonación y usar el comando git clone *URL*. No pide usuario ni contraseña ya que es un repositorio público. Además, si queiro hacer un push hacia ese repositorio, necesito los permisos necesarios