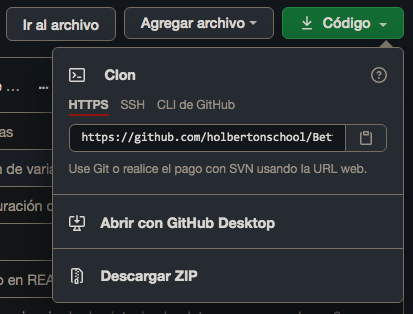
***Introducción a repositorios remotos (GITHUB)***

Resulta que si, por ejemplo, usted no se encuentra solo en un proyecto y necesita que varias personas, no solo usted, estén manipulando constantemente el proyecto en cuestión (haciendo avances y modificaciones sobre él, creando nuevas versiones del mismo, etc) y desea, al igual, contar con un sistema de versiones que los permite controlar cada copia de seguridad que se haga sobre el proyecto: es aquí donde se ve necesario el uso de ***“repositorios remotos”.*** Es así porque es el repositorio remoto el que nos da la facultad, a mí y a mi equipo de trabajo, de poder contar con un servidor unificado; es decir, un servidor donde estarían las versiones unificadas del trabajo de multiples desarrolladores (incluyendome).

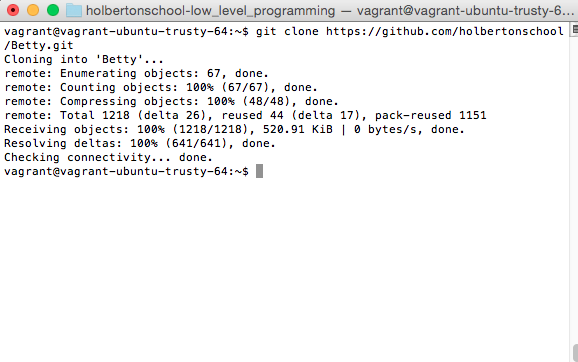
Con un repositorio local, solamente, no se podría trabajar en equipo sobre el mismo proyecto estando cada desarrollador en su propio entorno de trabajo o *setup.* He aquí donde surge la importancia de un servidor remoto (o siendo más precisos, un repositorio remoto): es, potencialmente, el punto de encuentro de todos los repositorios locales (de cada desarrollador) que quieran participar en un repositorio que los unifique sobre un mismo proyecto. Entonces, cada quién trabaja en local y, cuando terminan, lo mandan todo a esa nube en común que se llama repositorio remoto, donde todos pueden revisar el trabajo de cada uno. El repositorio de mayor aceptación laboral y el más popular a la vez es: **github.**

Ahora, si por ejemplo, hemos decidido primero crear un repositorio remoto (antes que un local). Si quisieramos importar ese repositorio remoto a nuestro potencial repositorio local, se hace através de un comando: **“git clone”.**

**git clone:** tiene la función de importar un repositorio remoto de *github* (u otro servidor externo), que aún no está vinculado a repositorio local (por lo general esto se hace cuando queremos vincular el repositorio remoto -o en nube- de un tercero... *para poder trabajar conjuntamente en un proyecto, tal cual*). Si el repositorio no nos pertenece, para importarlo debe hacerlo de esta forma (tomaremos un ejemplo):



Nos dirigimos a esta parte del repositorio remoto que queramos importar (este ejemplo se hace con GitHub) y copiamos link. Luego, para vincularlo a nuestro repositorio local, sería así:



Al llamar este *link* lo que hace “git clone”, en su maquina, es llamar completamente a la dirección del repositorio remoto en nuestro repositorio local; lo que quiere decir que, se trae sus archivos y lo hace de tal manera que: se trae una copia del “master” (del repositorio remoto) a nuestro directorio de trabajo (la carpeta que se le inicializó con git y que está ejecutando “git clone”) y, adicionalmente, crea una base de datos de todos los cambios historicos (copias de seguridad o “commits”) del repositorio remoto en nuestro repositorio local.

Del resto, seguiría trabajando igual cada vez que quiera exportar copias de seguridad, de su repositorio local, al repositorio remoto clonado. Cómo se exportan nuestros “commits” del repositorio local a un repositorio remoto vinculado? Con el comando: **“git push”.** *Git push* se encargaría de exportar nuestro “head” o última versión del proyecto guardada en copia de seguridad.

Ahora, si deseamos *importar* copias de seguridad del repositorio remoto a nuestro repositorio local (luego de ya haber clonado, ojo, al repositorio en sí. Importante esto), lo podemos hacer con una variedad de comandos de git, entre esos: **“git fetch”**, **“git merge”** y, el más poderoso, **“git pull”**. Vamos uno a uno.