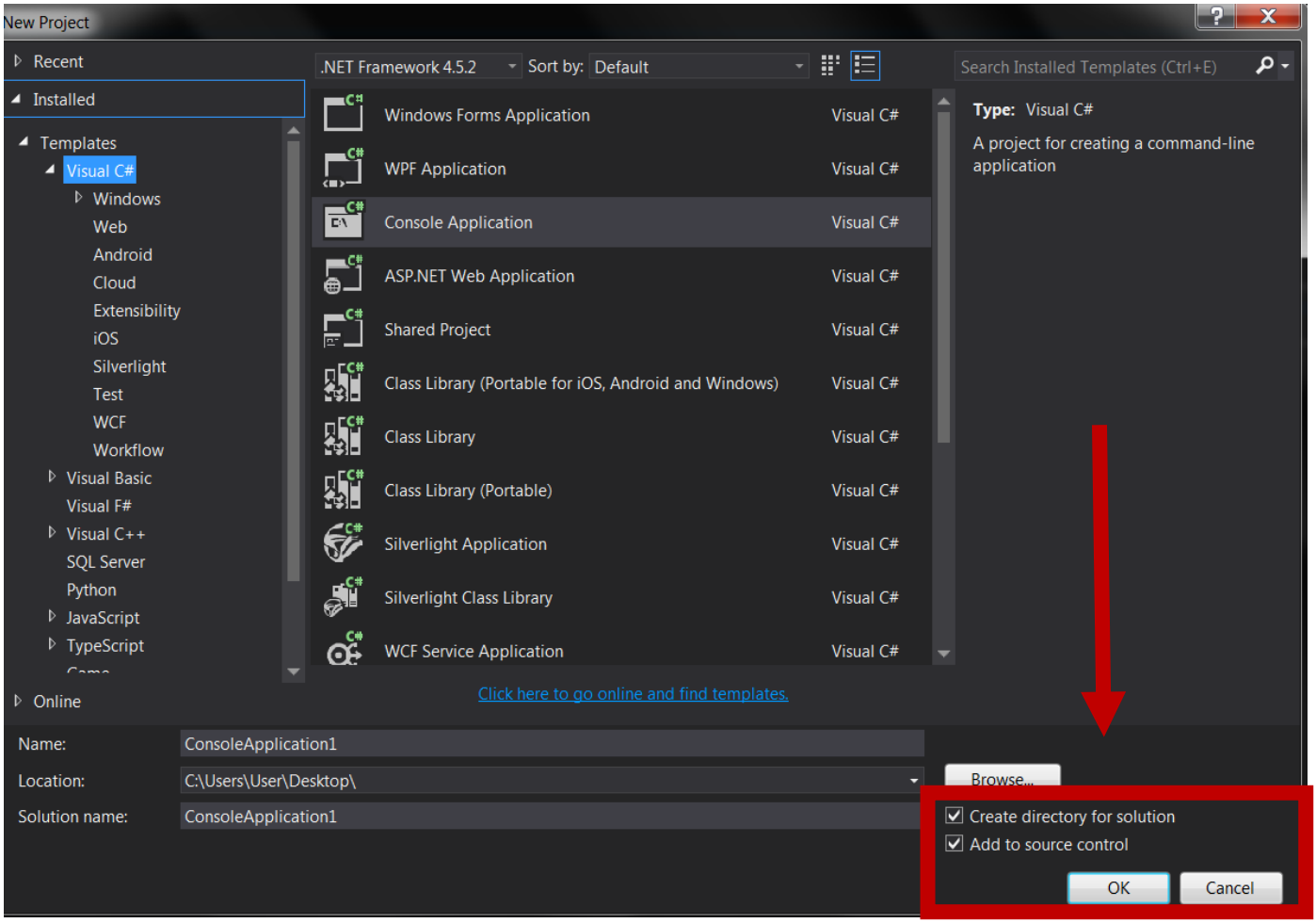
Guía para utilizar Git integrado con Visual Studio

¿Qué se necesita?

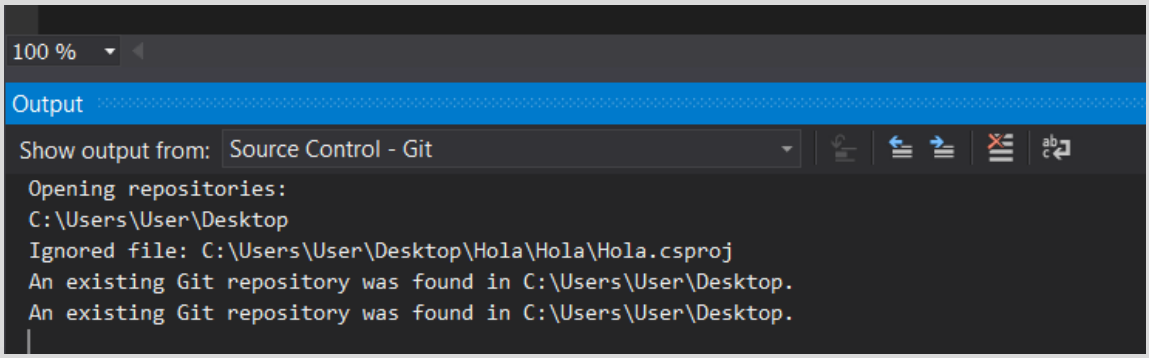
- Si decidimos usar Visual Studio como IDE + Git, debemos usar versiones recientes de Visual Studio, existen versiones gratuitas que se pueden encontrar en la página https://www.visualstudio.com/ o bien, para algunas carreras (como informática desde luego) los estudiantes de la UNED, tienen la posibilidad de utilizar una cuenta de DreamSpark® Premium. Para obtener esta cuenta hay que contactar al compañero que se entiende con el Partnership de Microsoft® - UNED, al tener esta cuenta se realiza el pedido y se descarga el programa (hay decenas de herramientas), pero para esto es necesario lo antes mencionado y también usar nuestra cuenta de correo de la UNED.
- Además las herramientas de Git se encuentran en https://git-scm.com/

¿Cómo usarlo?

En Visual Studio, al crear un nuevo proyecto, el asistente nos permite escoger si deseamos agregar un control de código:

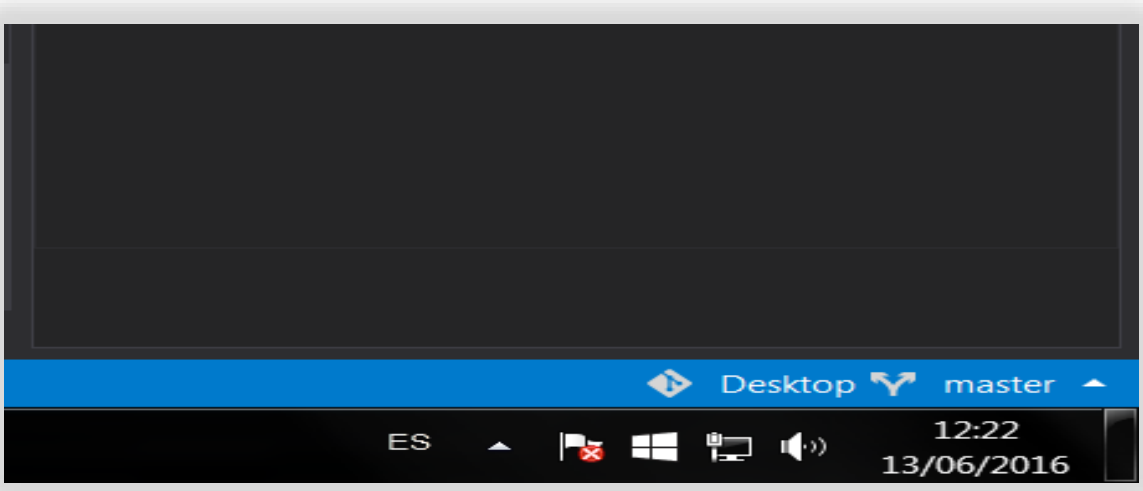


Si seleccionamos usar un control de código, una vez creado el proyecto, si no tenemos algún control de versiones configurado en VS, entonces VS enviará un mensaje para que seleccionemos el control de versiones que deseamos usar, de lo contrario creará el proyecto e integrará el controlador de código (control de versiones) que esté como predeterminado, en este caso Git.



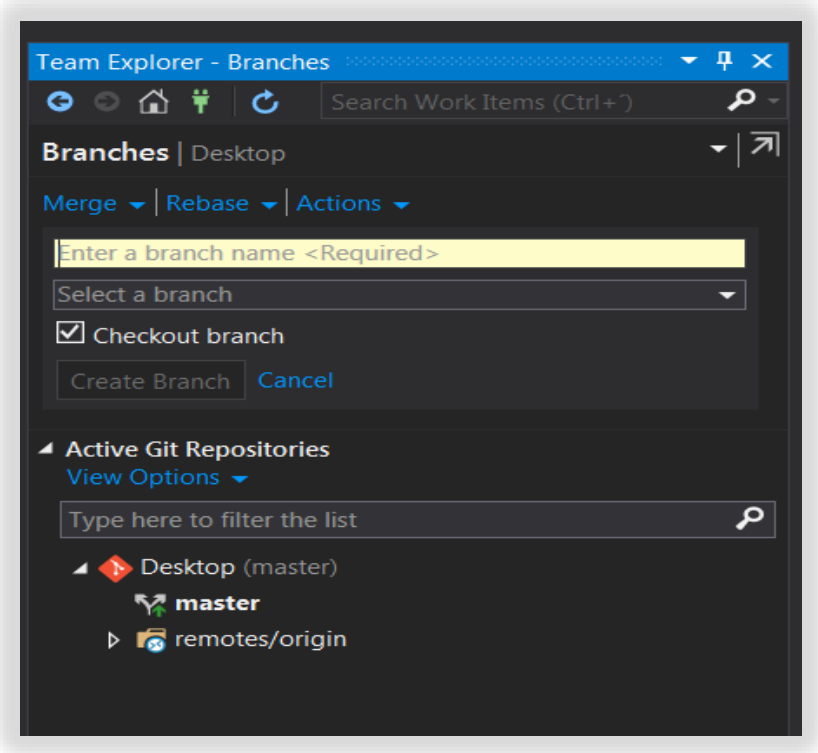
Como se ve, la información aparece en la consola integrada del VS, en la parte inferior del entorno, ya se observa cuál es el controlador de versiones usado, además de los repositos existentes y las rutas pertinentes al presente proyecto.

Si nos vamos a la esquina inferior derecha del IDE, veremos estos:



Si damos clic en la flecha de “master” se desplegarán varias opciones, donde se puede ver el manejo de branches (branch es la palabra que usan Git y VS para decir que es una derivación, rama o camino del proyecto), por defecto, desde luego un proyecto debe tener al menos una rama, la cual es la principal y llama master, por eso está creada.

Si seleccionamos lo referente a “new branch” o “manage branches”, se aprecia un panel de control al lado derecho del entorno.

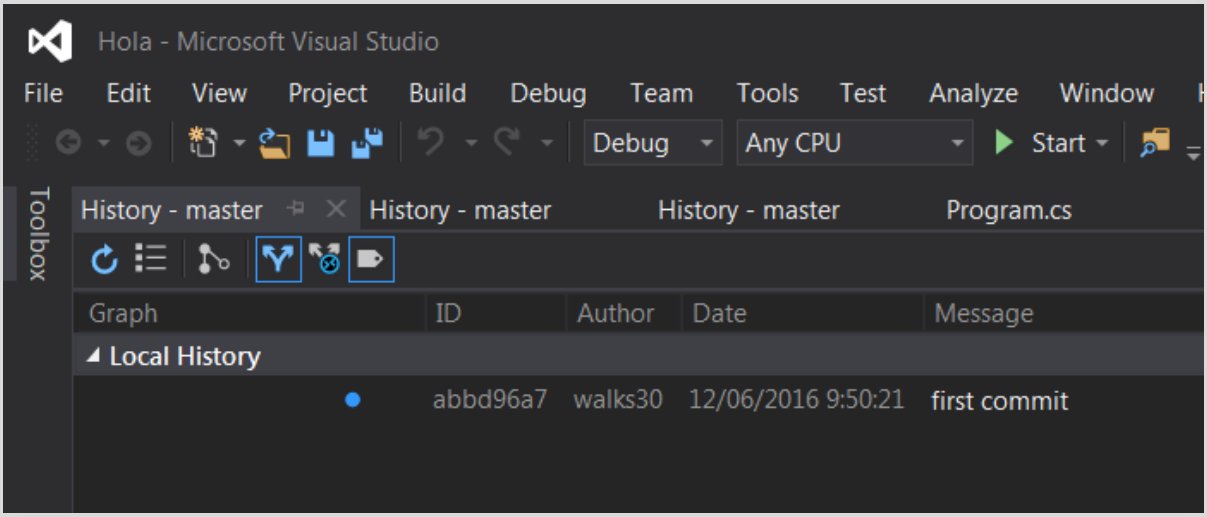


Como se aprecia, el panel permite crear una nueva rama del proyecto, referenciarse a ella, combinar (merge) ramas, navegar entre repositorios, etc. También se puede acceder a la consola (CMD).

Lo bueno es que acá ahorramos trabajo con el Bash, ya que se está integrando al VS de una forma más gráfica.

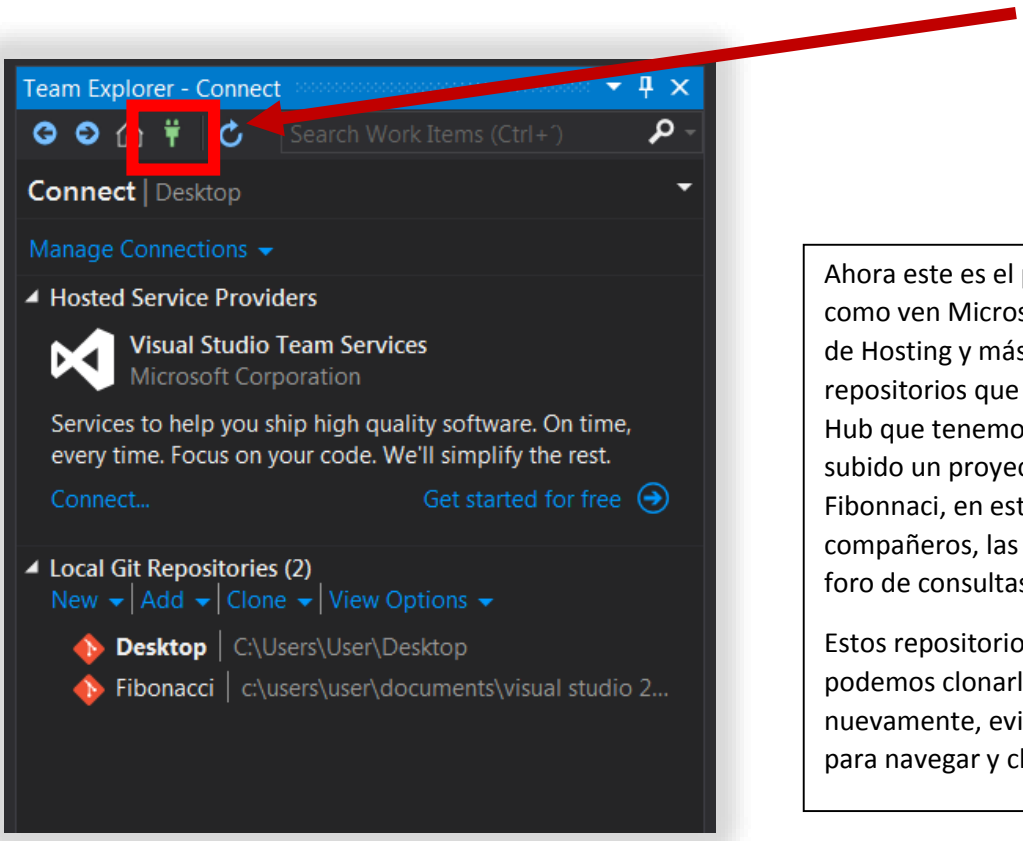
Si seleccionamos “Actions”, seguido de clic en la opción de ver historial, en el panel principal veremos el historial detallado de Git. Hay otras formas de ver el historial.

El historial detalla un ID y nótese el Autor, si no tenemos un autor configurado, Git y VS nos lo solicitarán, para ello es necesario tener una cuenta en Git Hub, es preferible instalar el VS, luego el Git y luego hacer la cuenta en Git Hub, pero todo esto antes de realizar la integración que estamos realizando. La fecha en formato y el mensaje. Como vemos acá no tuvimos que abrir el Bash y poner comandos.



Hasta ahora hemos cumplido con el crear un proyecto en VS, integrarlo a Git, historizarlo al estilo Git, pero aún el proyecto sigue localmente, para subirlo a la plataforma Git Hub es necesario sincronizar para hacer un push a través del VS.

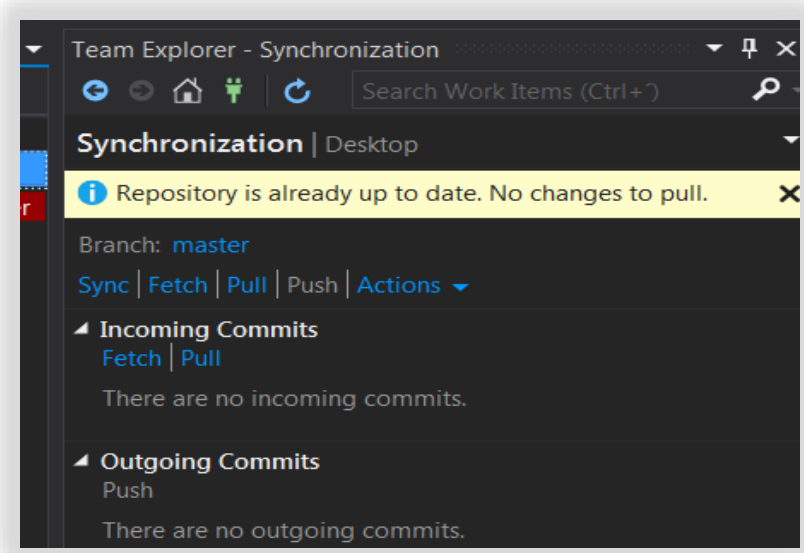
En el panel de control mencionado anteriormente, aparece un ícono de conexión...



Ahora este es el panel de conexiones, como ven Microsoft nos ofrece servicios de Hosting y más abajo tenemos los repositorios que están en la cuenta en Git Hub que tenemos conectada. Yo había subido un proyecto de VS en C# sobre Fibonnaci, en este aportamos dos compañeros, las interacciones están el foro de consultas del presente curso.

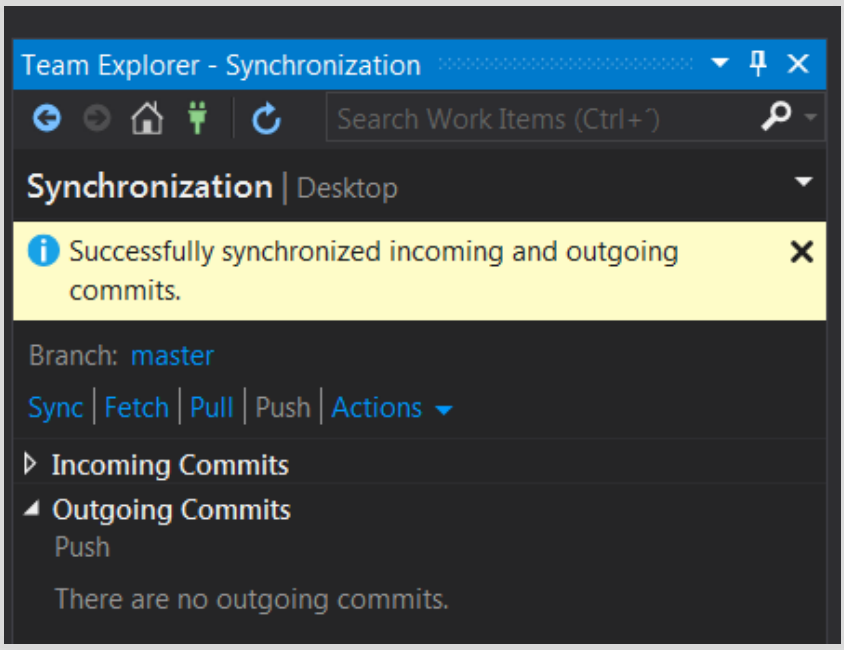
Estos repositorios ya alojados en Git Hub podemos clonarlos desde este panel, nuevamente, evitando comandos en Bash para navegar y clonar.

Por ejemplo, si quisiéramos que nuestro repositorio de Git, Fibonacci, sea enviado (o actualizado) a nuestro perfil en Git Hub, lo que debemos hacer es agregarlo, con la ruta, en la imagen arriba ya está agregado. Y luego dar doble clic en el Repositorio local y luego clic en Sync o sincronizar, VS si VS no ha realizado y guardado nuestra cuenta para sincronizar, entonces nos pedirá las credenciales (usuario y contraseña de nuestro usuario en Git Hub). Posteriormente podremos hacer lo que queramos con nuestro Git Hub: pull, push…



Ya acá hemos logrado hacer todas las integraciones con Git, Git Hub y VS. Al llegar a este panel, luego de darle las credencias de nuestra cuenta, VS ya estará trabajando con Git Hub.

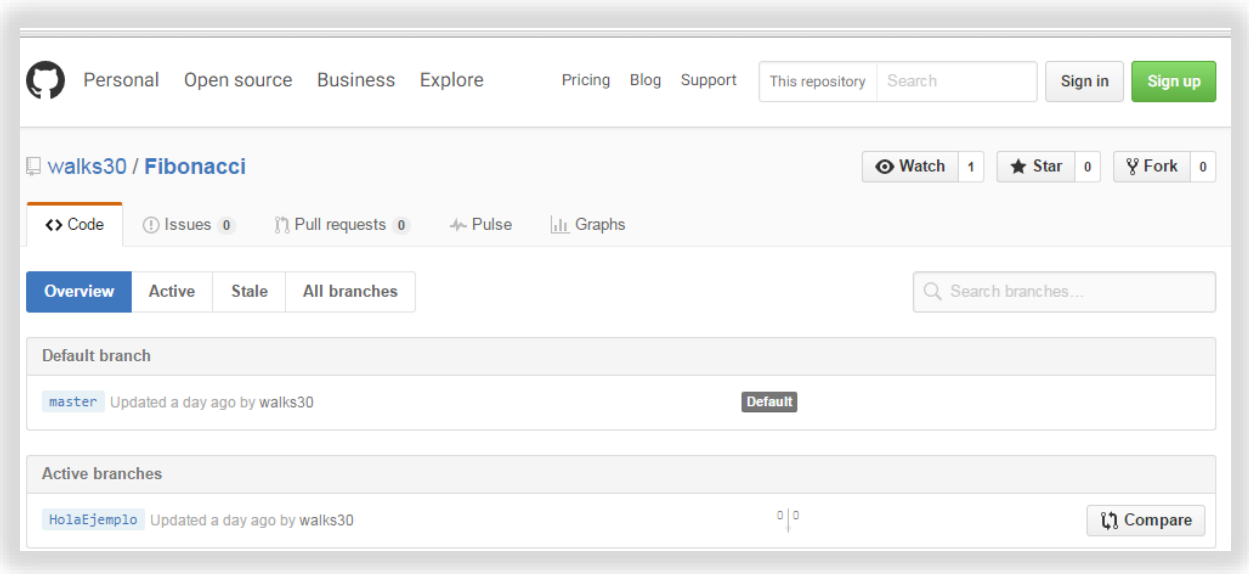
Para comprobar que ha hecho el trabajo, voy a sincronizar Fibonnaci… y luego ver qué pasa.



Como se aprecia, el VS ha sincronizado con Git Hub.

Voy a crear otra rama, o branch de la siguiente forma: En la esquina inferior derecha del VS, clic en “master”, luego “New Branch”, luego seguir lo que dice el asitente… asignarle un nombre (la llamaré HolaEjemplo) y anidarlo en otro Branch (en el master). Luego “Crear” y finalmente seleccionar en la esquina inferior derecha al Branch y sincronizarlo.

Si me fijo en el perfil de Git Hub, ya el proyecto tendrá dos ramas.



Con esto se hace una integración básica de Git y Git Hub con Visual Studio. Si no se tiene un IDE que soporte esta integración, se puede usar el Bash, aunque es más complicado, pero importante saber que se puede usar el Bash y si no, el Git Hub Desktop, para evitar los comandos del Bash y hacerlo por una interfaz, sin embargo, el Git Hub Desktop pesa más de 100 Mb.

Cuando se crea una cuenta en Git Hub hay que hacerlo con un correo válido para recibir el mensaje de Git Hub, ya que a diferencia de FB y otros servicios, la activación a veces se pasa por alto, en el caso de Git Hub no es así, si queremos subir un proyecto, se debe haber verificado la cuenta de cuenta.

Para las sincronizaciones, a veces el antivirus o un firewall de un tercero puede estar bloqueando las sincronizaciones, por lo que, si hay problemas para sincronizar y aparentemente todo en VS, Git y Git Hub está bien configurado, debemos verificar el antivirus o firewall.