

VISUAL STUDIO CON GIT

Antes de comenzar, crearemos si no tenemos una cuenta, una cuenta en GitHub, esto lo realizaremos desde <https://github.com/>.

Luego comenzaremos con la siguiente documentación:

Actualmente se hace prácticamente esencial en todo desarrollo de software trabajar con un gestor de código que nos brinde ayuda en tareas como las operaciones de versionado, el almacenamiento de código, la modificación continua de código para corregir errores, la mejora del rendimiento o el ajuste de nuestro desarrollo a nuevos requerimientos, entre otras.

A día de hoy, el repositorio de código más utilizado por la industria es **Git**. Dentro de Git, **GitHub** es sin duda la plataforma Git más potente y utilizada en la actualidad. Siguiendo la filosofía de servicios que está transformando Microsoft, **Visual Studio**, desde su versión **2013**, incluye integración con repositorios Git sin necesidad de instalar extensiones.

GitHub en Visual Studio

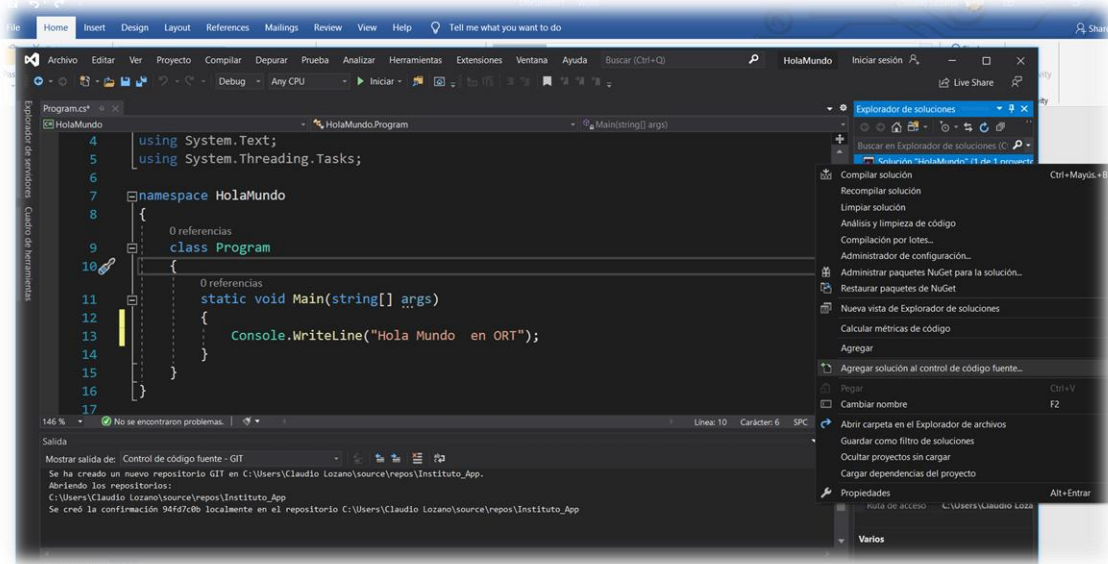
Actualmente, la integración del **IDE de Visual Studio con Git** es total. Esta integración se ha mejorado mucho en las últimas versiones del IDE de Visual Studio (2015 y 2017 como también en la actual 2019). A continuación, veremos una serie de operaciones no tan comunes que se incluyen al introducir **Git**:

- **Rebase.** Permite integrar el trabajo que se tiene en una rama puntual. Podemos situarnos sobre ella y lanzar el comando **Rebase**; de esta forma recompondremos los cambios encima de nuestra actual rama 'master'. Si funciona, se realizará un avance rápido (**fast-forward**) en nuestra rama master, y se acabará teniendo un historial lineal en nuestro proyecto.
- **Cherry-pick.** El otro camino para introducir cambios de una rama a otra es entresacando. Entresacar (cherry-pick) en Git es como reorganizar (rebase) una sola confirmación de cambios (**commit**). Se trata de coger el cambio introducido por una determinada confirmación de cambios y reaplicarlo sobre la rama donde nos encontremos en ese momento. Puede ser útil si tenemos varias confirmaciones de cambios en una rama puntual y tan sólo deseamos integrar una de ellas, o si se tiene una única confirmación de cambios en una rama puntual y preferimos entresacarla en lugar de reorganizarla.
- **Pull Request.** Permite a los colaboradores comentar los cambios propuestos en las solicitudes de "**merge**", aprobar los cambios o solicitar cambios adicionales antes de fusionar la solicitud de "**merge**". Los administradores del repositorio pueden exigir que todas las solicitudes de "**merge**" sean aprobadas antes de fusionarse con el código master del servidor.
-

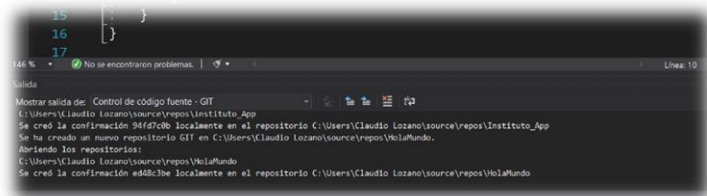
Configurar un repositorio Git en Visual Studio Online

A continuación, se detallará cómo añadir un proyecto existente a Visual Studio Online haciendo uso de un repositorio Git mediante el **IDE de Visual Studio 2017/19**.

Trabajaremos sobre un proyecto existente. En primera instancia en la solución de nuestro proyecto, hacemos clic derecho. En el menú que nos aparece, le damos a **“Add Solution to Source Control”**.



Esta acción sobre la versión **2019** nos creará el control de Código en **Git**, repositorio por defecto que hoy día aplicamos en **Visual Studio**.

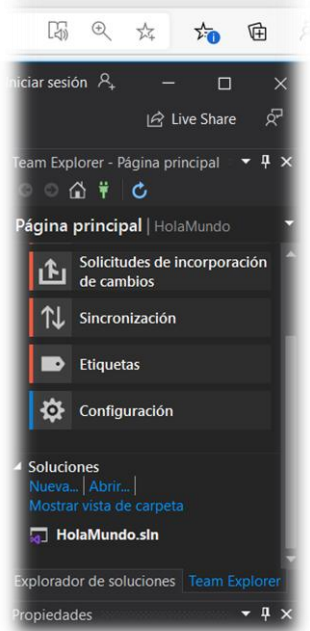
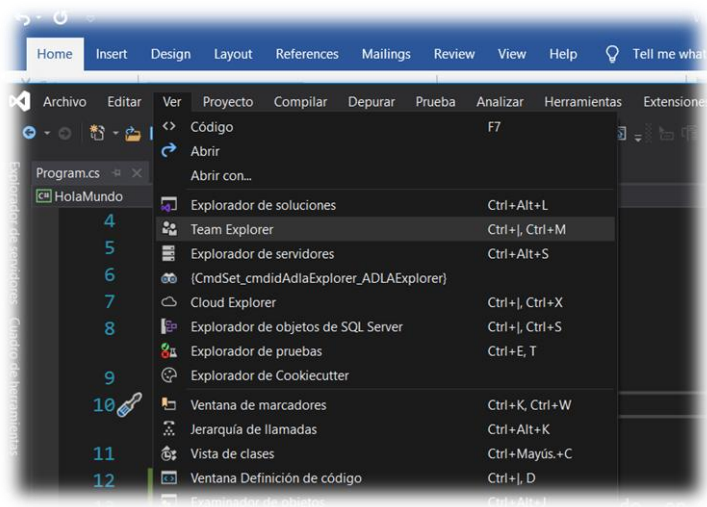


En el caso de usar versiones como **2015** o **2019**:

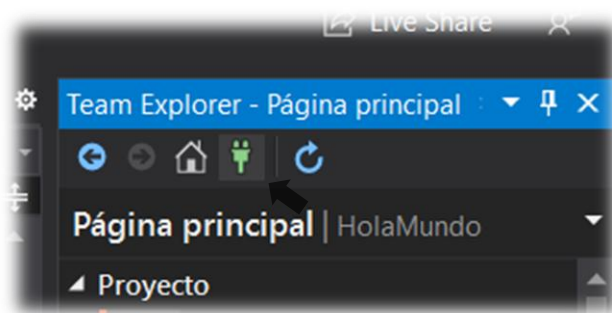
Esta acción desencadenará otra acción que nos dará a elegir el control de código que queremos utilizar. Por defecto, se puede elegir entre **Team Foundation Version Control** y **Git**. Si marcamos el **checkbox** de debajo, se usará siempre la opción que seleccionemos en ese momento. Seleccionamos la opción **“Git”** y presionamos el botón **“OK”**.



Ahora vamos a la pestaña **“Team Explorer”** y después a **“Changes”**.

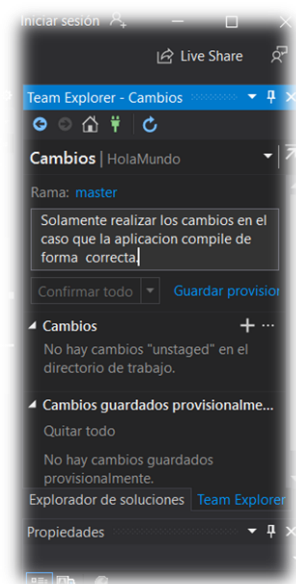


Deberemos conectar a un repositorio, en este caso local para poder comenzar a administrar los cambios que hay en el proyecto y seleccionaremos el repositorio que hemos creado recién.

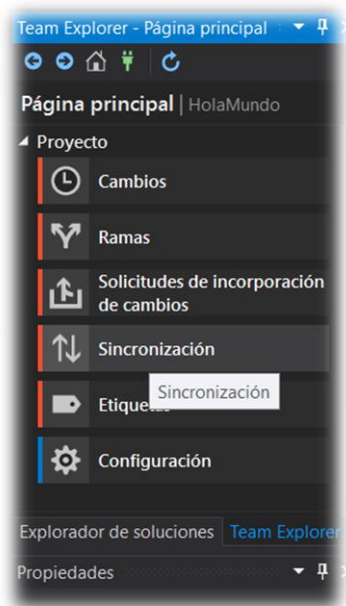




Una vez ahí, ya podremos hacer **commit**, que no es más que un salvaguardado de los cambios en el repositorio local en el caso de Git. Siempre que realicemos un **commit**, deberemos incluir un comentario explicativo con una notación consensuada con el equipo de trabajo.

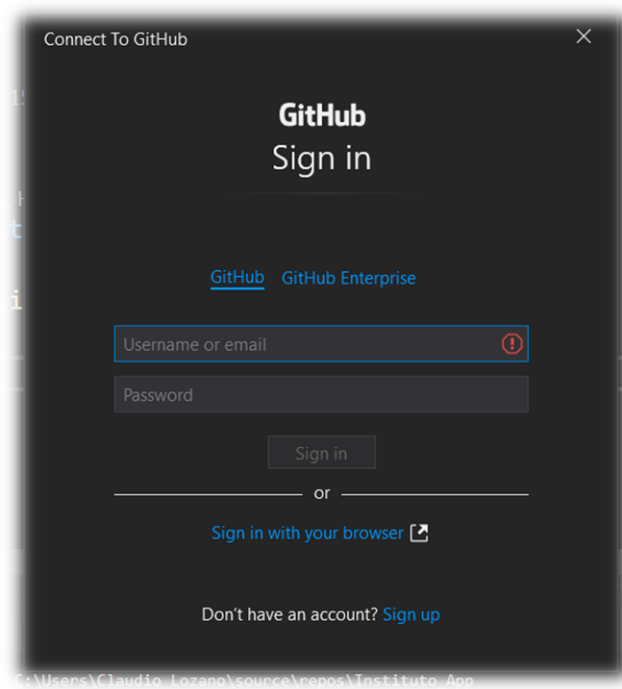


Volvemos a la pantalla general (**Home**) de **Team Explorer**. En esta pantalla, podremos sincronizar nuestro repositorio local con el remoto haciendo clic en la pestaña **"Sync"**. Esto también se puede hacer desde la pantalla de **"Changes"** tras hacer **commit**, en el link que nos aparecerá tras la acción.



Una vez subidos los cambios a nuestro servidor local de **Git** mediante el comando **commit**, tendremos que subir estos cambios a nuestro servidor de **Git** mediante el comando **push**. Para poder hacer **Publish** y así subir nuestros cambios de local al servidor, deberemos introducir la dirección del servidor si es la primera vez que realizamos esta acción. En este caso se ha empleado **Team Services** con **Git** para tenerlo publicado en nuestro Visual Studio Online haciendo uso del repositorio **Git**.

Si no se unió previamente al repositorio Git, le pedirá las credenciales para poder acceder al repositorio:



Como último, realizaremos la publicación presionando **Publish**.

