

# Git y Github

Github en Visual Studio



---

# Índice

Introducción	3
Abrir una nueva terminal	4
Clonar un repositorio con Visual Studio	7

---

# Introducción

*Visual Studio Code* es un editor de código multiplataforma y soporta la sintaxis de una gran cantidad de lenguajes de programación.

La herramienta proporciona soporte y asistencia a lenguajes de diverso ámbito: HTML, CSS, JavaScript, diversas variantes de C, Java, SQL, PHP, Ruby, Visual Basic o JSON entre muchos otros, soportando resaltado, sangrado automático, snippets y autocompletado.

Es uno de los editores de código más usado a nivel mundial, tiene detrás una amplia comunidad de apoyo, multitud de foros de ayuda y una ingente cantidad de *plug-ins* disponibles para facilitarnos la tarea de programas sea cual sea el lenguaje en el que trabajemos.

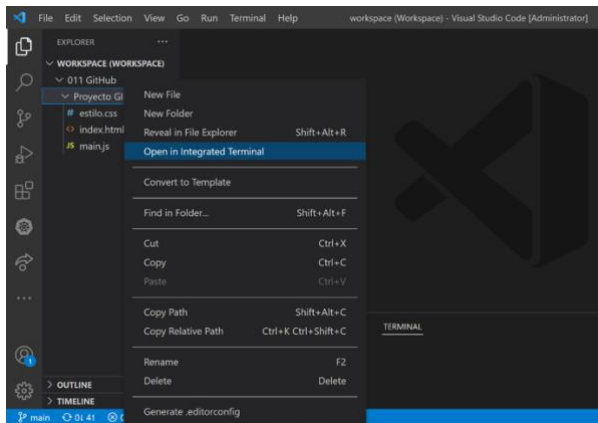
Además, Visual Studio tiene integración nativa con *Git*. Es decir, viene ya preparado para poder trabajar con *Git* sin necesidad de instalar Software adicional.

Para empezar a trabajar con *Git* en *Visual Studio*, lo único que tendremos que hacer es tener un proyecto creado, es decir, uno o varios archivos con los que empezar a trabajar, abrir la terminal en nuestro directorio de trabajo y teclear:

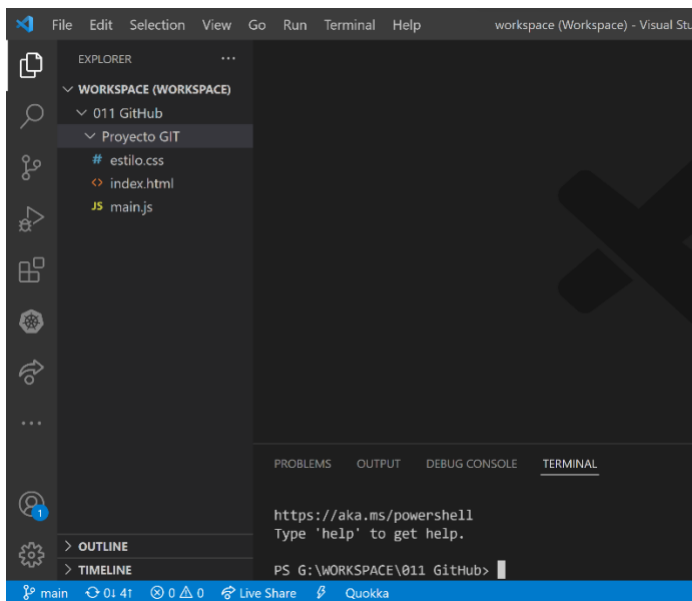
```
git init
```

## Abrir una nueva terminal

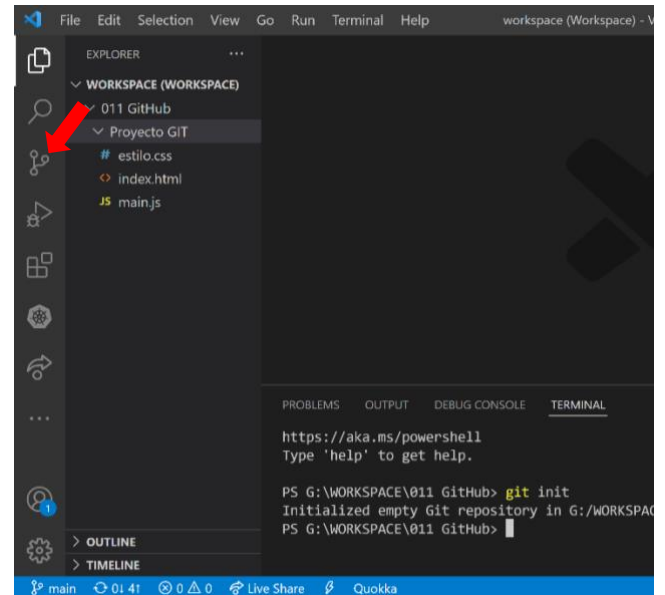
Para abrir una nueva terminal, deberemos situarnos sobre nuestro directorio de trabajo, pulsar el botón derecho del ratón y darle a la opción “*Open in integrated terminal*”:



Esto nos abrirá la terminal integrada de *Visual Studio*. Esta terminal funciona de manera análoga a como lo hace la terminal *Git Bash* que hemos estado usando hasta ahora.



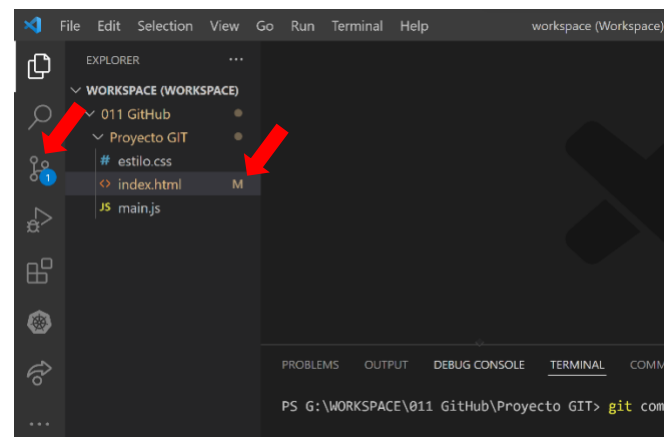
Una vez abierta la terminal sólo tendremos que teclear `git init` para que *Git* comience a monitorear nuestro proyecto.



El icono señalado en rojo es el que despliega el menú de opciones de *Git*. En él se nos informará de todos los cambios que haya pendientes en nuestros archivos para hacer *commit*.

Ahora deberemos añadir nuestros archivos al *staging area*.

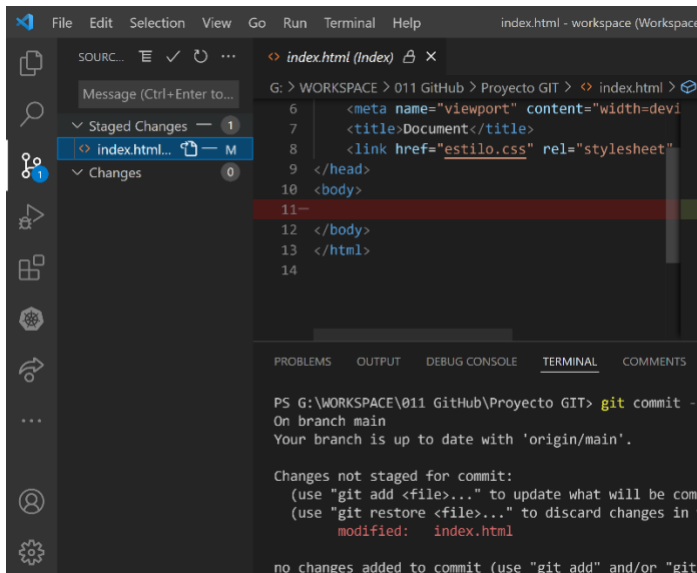
Recordemos que eso se hace con `git commit -m “descripción”`



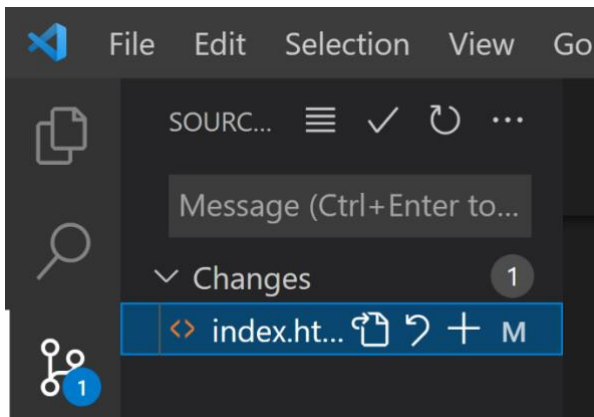
La letra **M** nos está indicando que en ese archivo hay cambios pendientes de sincronizar. Y si vemos, el icono de las ramas de *Git* nos indica también dicho cambio.

Todos los comandos de *Git* que hemos usado hasta ahora, `add`, `branch`, `commit`, `merge`...etc, son perfectamente aplicables en la terminal de Visual Studio.

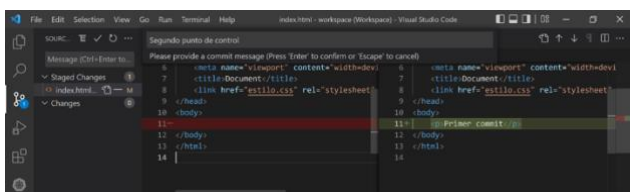
Si pulsamos sobre nuestro archivo, veremos como *Visual Studio* nos abre una doble pantalla en la que nos muestra tanto la versión de nuestro archivo antiguo como la nueva con los cambios que hayamos realizado:



Pero además tendremos la opción de hacer *commits* sin tener que pasar por el *staging area* simplemente pulsando el icono señalado con la flecha roja:

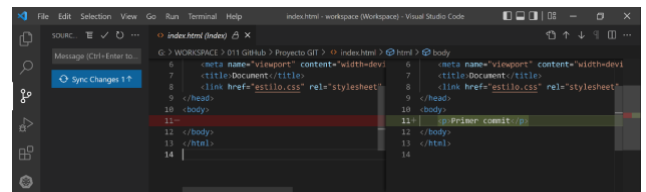


Al pulsar sobre el icono que señala la flecha roja se nos abrirá una pequeña ventana donde tendremos que poner la descripción de nuestro *commit*. En este caso le llamaremos “segundo punto de control”:

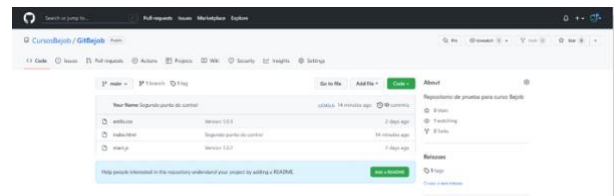


Al escribir la descripción y darle a ENTER se creará nuestro *commit*.

De momento tenemos nuestros archivos actualizados, pero en local. Para subir las nuevas versiones de los archivos al repositorio *GitHub* tendremos que darle al botón “Sync Changes”:



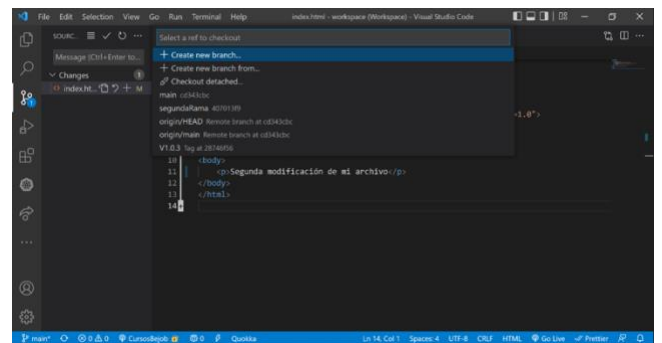
Y ya tendremos los archivos actualizados y subidos a la nube de *GitHub*:



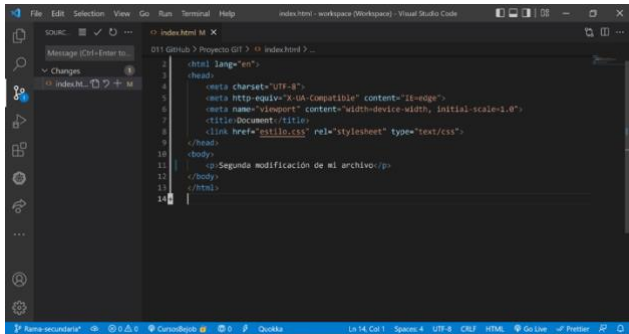
**Nota:** Para que dicha sincronización se lleve a cabo deberemos estar dados de alta en *GitHub* y tener el repositorio de la nube creado.

Como vemos, *Visual Studio* nos permite hacer el paso del *Staging area* y el *commit* simplemente pulsando un icono.

Para crear nuevas ramas podemos hacerlo mediante los comandos que ya conocemos o pulsando en la opción resaltada en la siguiente captura:

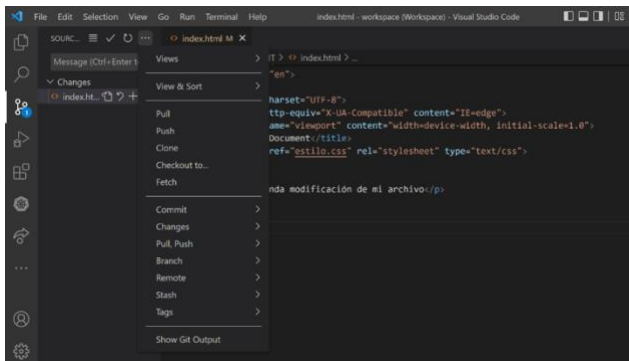


Pulsamos y se nos abre un menú con diferentes opciones, entre ellas la de “Create new Branch”, le ponemos un nombre a nuestra nueva rama, por ejemplo “Rama secundaria” y ya podríamos trabajar desde dicha rama:



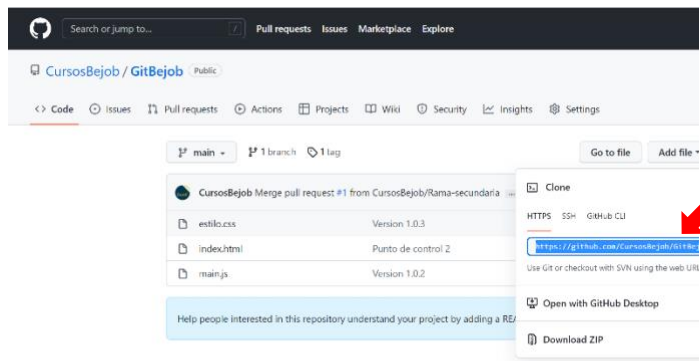
Además, desde esta opción podremos cambiar entre las diferentes ramas que tengamos.

Esta operación y todas las que tengan que ver con *GitHub* también las podemos hacer desde el menú desplegable que se abre al pulsar sobre el icono de los tres puntos:



# Clonar un repositorio con Visual Studio

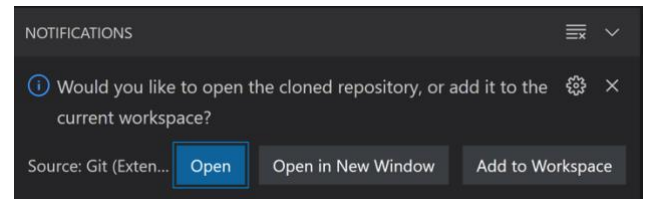
Si ya disponemos de un repositorio en GitHub y queremos que nuestro Visual Studio sincronice con él, deberemos copiar la URL de nuestro repositorio:



Y nos pedirá que le digamos en qué directorio de nuestro disco duro queremos clonar el repositorio.

Una vez lo hagamos, se copiarán todos los archivos del repositorio al directorio que hayamos elegido.

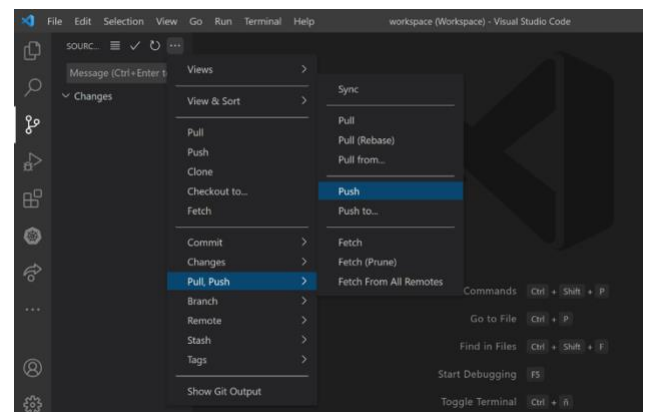
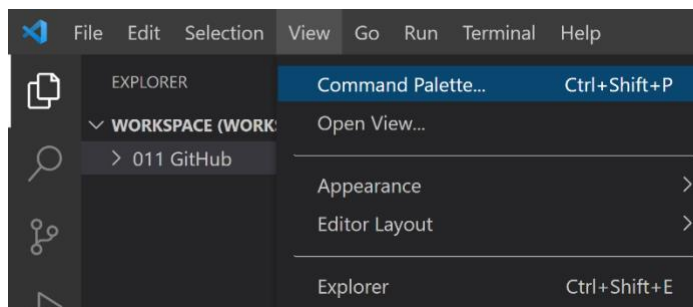
Nos saldrá un aviso en el que se nos pregunta si queremos añadir dicho repositorio a nuestro entorno de trabajo:



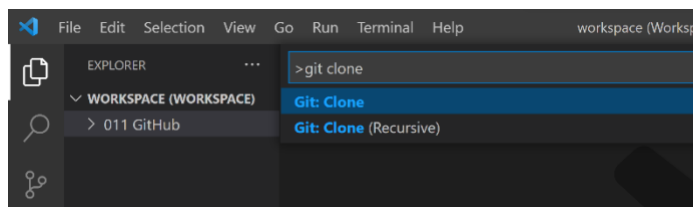
Y si le damos a “open” ya podremos seguir trabajando en el repositorio desde local.

Si hacemos cambios en alguno de los archivos desde local y queremos subirlos a la nube, tendremos que ejecutar el comando **push**:

A continuación nos vamos a Visual Studio y abrimos el menú “view/Command Palette” :



Escribimos el comando “git clone”:



Nos pedirá la URL de nuestro repositorio:

