de Javier Carrasco

Entornos de desarrollo

Bloque 1

Unidad 8: Entornos de propósito específico

1.8.1. Entornos de desarrollo e IDE

Como ya debería saber, un (*software)* **entorno de desarrollo** es un conjunto de herramientas que nos permiten automatizar algunas tareas, y nos dan soporte para análisis, diseño, codificación, pruebas, validación, etc. Debe haber un alto nivel de integración dentro de estos entornos, para que estas tareas puedan interoperar entre ellos.

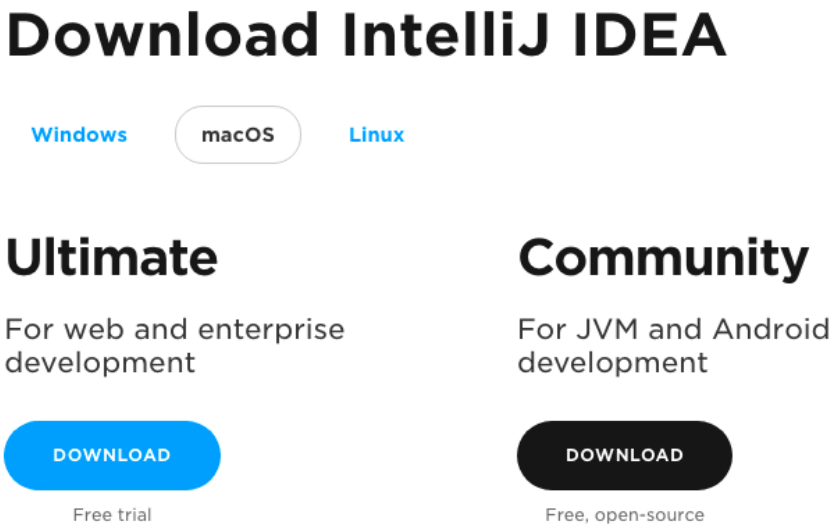
Entre estos entornos de desarrollo, podemos encontrar algunos entornos de propósito general, como Geany o Visual Studio Code, que pueden hacer frente a muchos lenguajes diferentes. Además, existen otros entornos específicos que están especialmente diseñados para un idioma (o una lista reducida de ellos). En esta unidad nos vamos a centrar en este segundo grupo, ya que el primero se ha explicado en unidades anteriores.

Para ser más precisos, nos centraremos en **IntelliJ IDEA,** ya que es el IDE que usaremos para la mayoría de los ejercicios de este módulo. Además, veremos una descripción general rápida de algunos otros IDE específicos, como Visual Studio.

1.8.2. IntelliJ IDEA

IntelliJ IDEA es un IDE multiplataforma desarrollado por JetBrains . Fue lanzado por primera vez en 2001 como *IntelliJ,* y fue uno de los primeros IDE con navegación avanzada y refactorización de código.

Está disponible en dos versiones: *Edición comunitaria (* libre y *Ultima edición (* comercial). La principal diferencia entre ellos se puede encontrar en los idiomas y sistemas de control de versiones compatibles. Por ejemplo,
Comunidad La versión no admite PHP o Javascript. En nuestro caso, como vamos a trabajar con Java Virtual Machine, necesitamos descargar el *Comunidad* edición, que es gratuita y de código abierto. Este es el página de descarga .



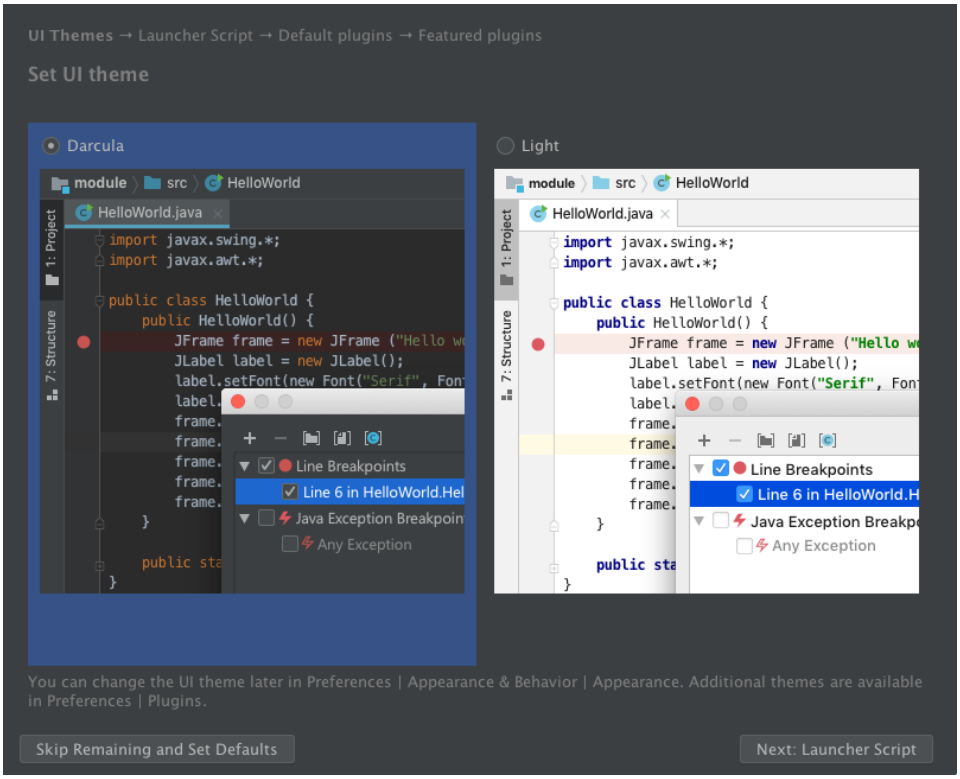
1.8.2.1. Instalación y configuración

Respecto al proceso de instalación:

- En **Ventanas** tenemos un asistente paso a paso que nos guía a través de todo el proceso de instalación. Podemos elegir la carpeta de instalación (o simplemente dejar la configurada por defecto), y si queremos crear un acceso directo en el escritorio.

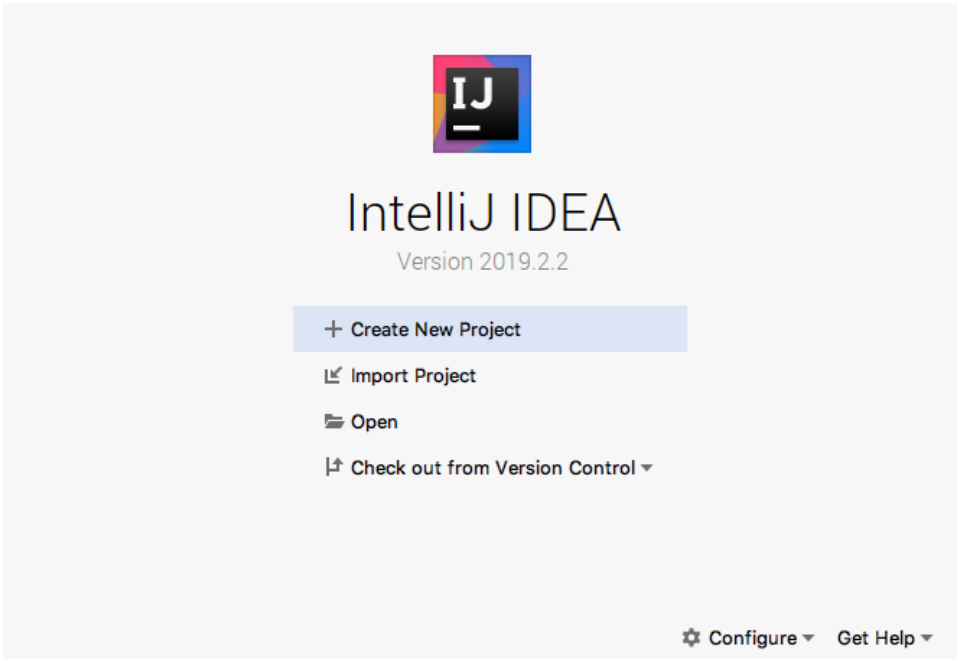
- En **Mac OS X,** tenemos un instalador que nos pide que arrastremos la aplicación al *Aplicaciones* carpeta. En **Linux** descargamos un *tar.gz* archivo que debemos
- descomprimir. Dentro de la carpeta principal hay una *compartimiento* subcarpeta. Debemos entrar en esta carpeta desde un terminal y ejecutar el comando `./ idea.sh` para iniciar el IDE. La primera vez que lo ejecutemos, creará un acceso directo en algún lugar del menú de aplicaciones, por lo que podemos iniciar IntelliJ usando este acceso directo a partir de ese momento.

La primera vez que corremos *IntelliJ,* nos permite importar configuraciones anteriores, si teníamos alguna versión anterior instalada. Si no, podemos simplemente elegir " *No importar configuraciones ".* A continuación, podemos elegir el tema de la interfaz de usuario ...



A continuación, nos permite definir un script para lanzar programas desde la línea de comandos, pero podemos omitir este paso y pasar al siguiente, en el que podemos optar por instalar algunos complementos adicionales. También podemos dejar este paso con su configuración predeterminada y empezar a usar *IntelliJ*.

Esta es la pantalla de bienvenida:

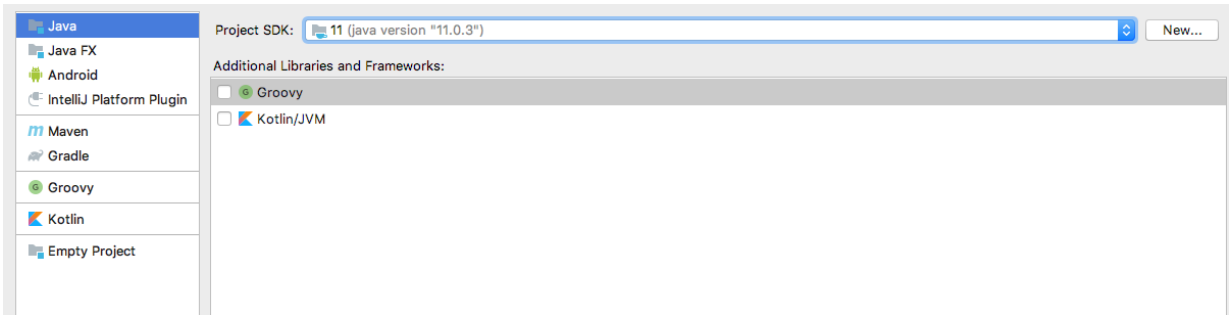


1.8.2.2. Creando proyectos Java

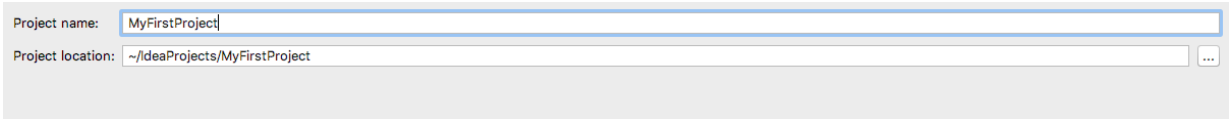
Desde la pantalla de bienvenida, podemos elegir entre:

- Creando un nuevo proyecto
- Importar un proyecto existente (si lo creamos en otra computadora y queremos usarlo en nuestra actual). Abra un proyecto existente desde
- nuestra computadora
- Consultar un proyecto desde un repositorio remoto

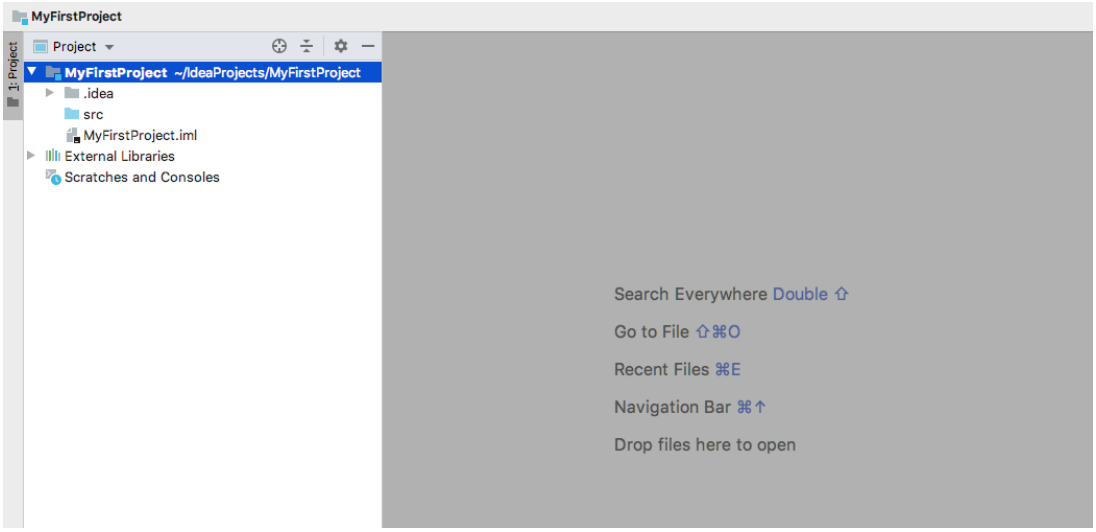
Entonces, si queremos crear un nuevo proyecto, elegimos la primera opción *Crear nuevo proyecto,* y luego especificar que queremos crear un proyecto Java, desde el panel izquierdo:



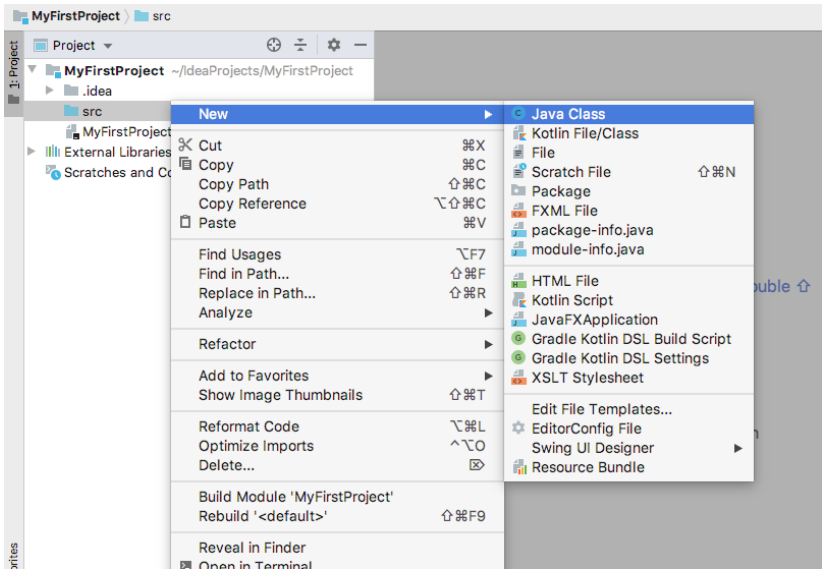
A continuación, podemos elegir una plantilla para nuestro proyecto, pero podemos omitir este paso y pasar al siguiente, en el que debemos especificar un nombre y una ubicación del proyecto (podemos dejar la ubicación predeterminada si queremos):



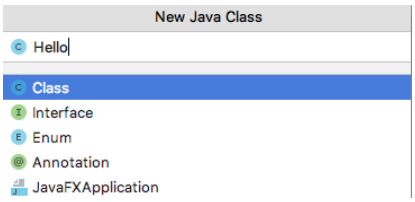
Luego, hacemos clic en *Terminar* y veremos nuestro proyecto. Si hacemos clic en la pestaña del proyecto a la izquierda, podemos ver la estructura de la carpeta del proyecto y crear elementos (archivos fuente) en ella.



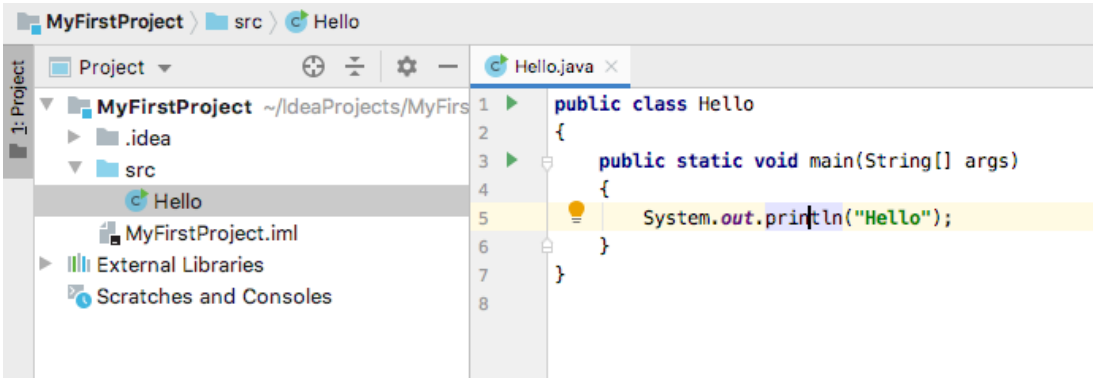
Ahora, creemos nuestro primer archivo fuente. Haga clic derecho en el *src* carpeta y elija *Nuevo> Clase Java.*



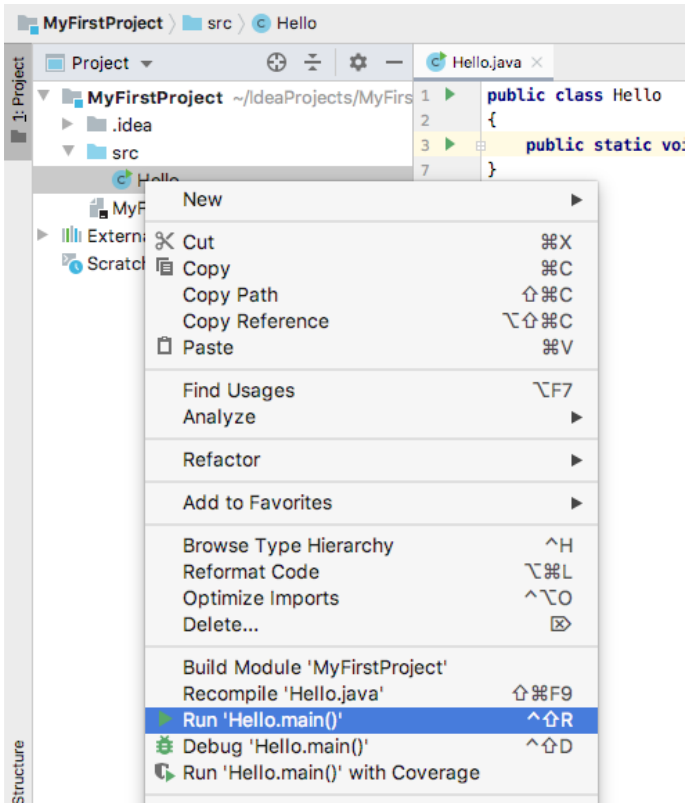
Luego, debemos especificar el nombre de la clase. Por ejemplo, Hola .



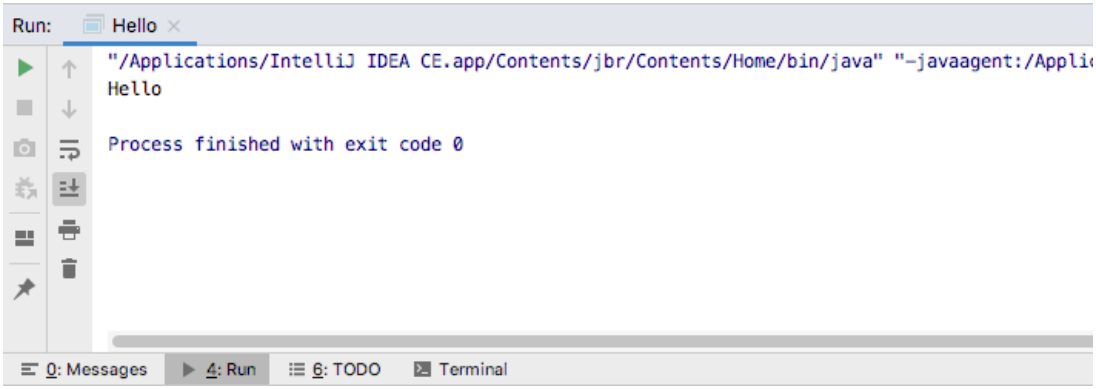
Se creará un nuevo archivo y podremos editarlo en el área principal. Podemos dejar un código como este:



Podemos ejecutar el archivo haciendo clic derecho sobre él y eligiendo " *Ejecute Hello.main ()* " opción:



Luego, podemos verificar los resultados en el terminal integrado en la parte inferior de la ventana:



1.8.2.3. Atajos

En la siguiente tabla puede ver algunos de los accesos directos más comunes disponibles para *IntelliJ IDEA* debajo Ventanas sistemas.

Atajo	Acción
Ctrl + Cambio + N Ctrl + norte	Abra un archivo nuevo.
	Abra cualquier clase.
Ctrl + Barra espaciadora	Código completo.
Ctrl + Cambio + Ctrl de la barra espaciadora + S	Finalización de código inteligente.
	Guardar el archivo.
Ctrl + O	Sobrescribir métodos.
Ctrl + yo	Implementar todo.
Ctrl + /	Línea de comentario / descomentar. Línea
Ctrl + re	duplicada.
Ctrl + Z	Deshace la última acción.
Ctrl + Cambio + Ctrl Z + F	Rehacer la última acción deshecha.
	Mostrar diálogo de búsqueda.
Ctrl + R	Mostrar diálogo de reemplazo.
Ctrl + F9	Compilar proyecto.
Cambio + F10	Ejecutar proyecto.
Cambio + F9	Depurar.
F7	Paso a la función (en *depurar* modo). Siguiente línea (en *depurar*
F8	modo). Detener la depuración.
F9	
Ctrl + F8	Crea un punto de interrupción.
Ctrl + Cambio + F12	Maximizar el panel del editor.

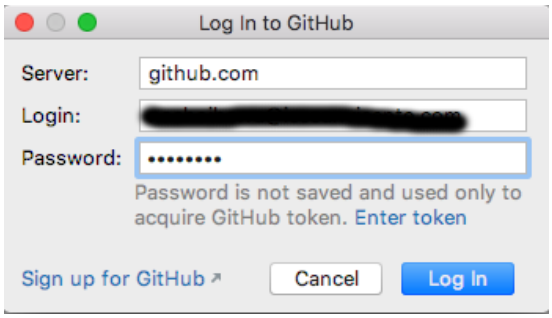
La mayoría de estos atajos también están disponibles en Linux. Con respecto a los sistemas MacOSX, debe reemplazar Ctrl clave con Cmd llave. Puedes encontrar más atajos aquí .

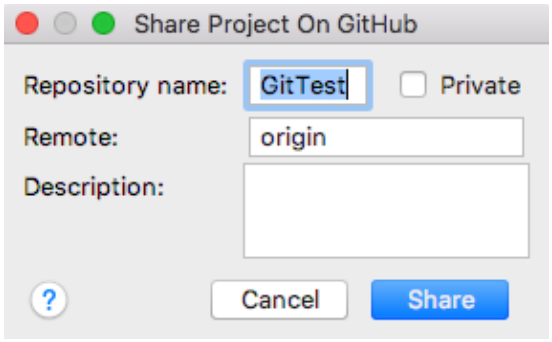
1.8.2.4. Integración con Git

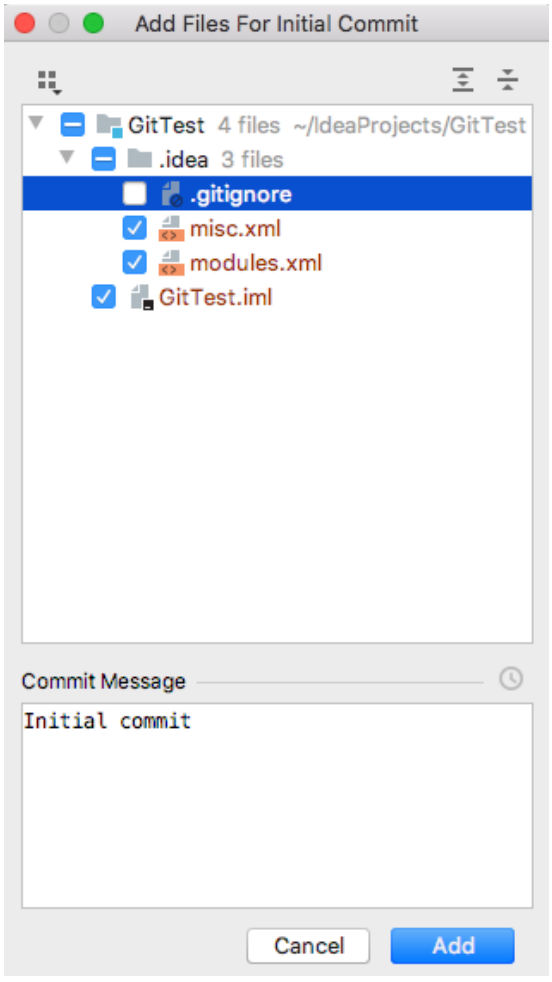
La integración de *IntelliJ* con repositorios de Git es realmente fácil:

- Si queremos **clonar o descargar un repositorio remoto** con un proyecto IntelliJ existente para comenzar a trabajar en él, simplemente vamos a *VCS> Pago desde Control de versiones> Git* menú. Luego, especificamos la URL del repositorio y se descargará a nuestra ubicación especificada (en nuestra máquina local).

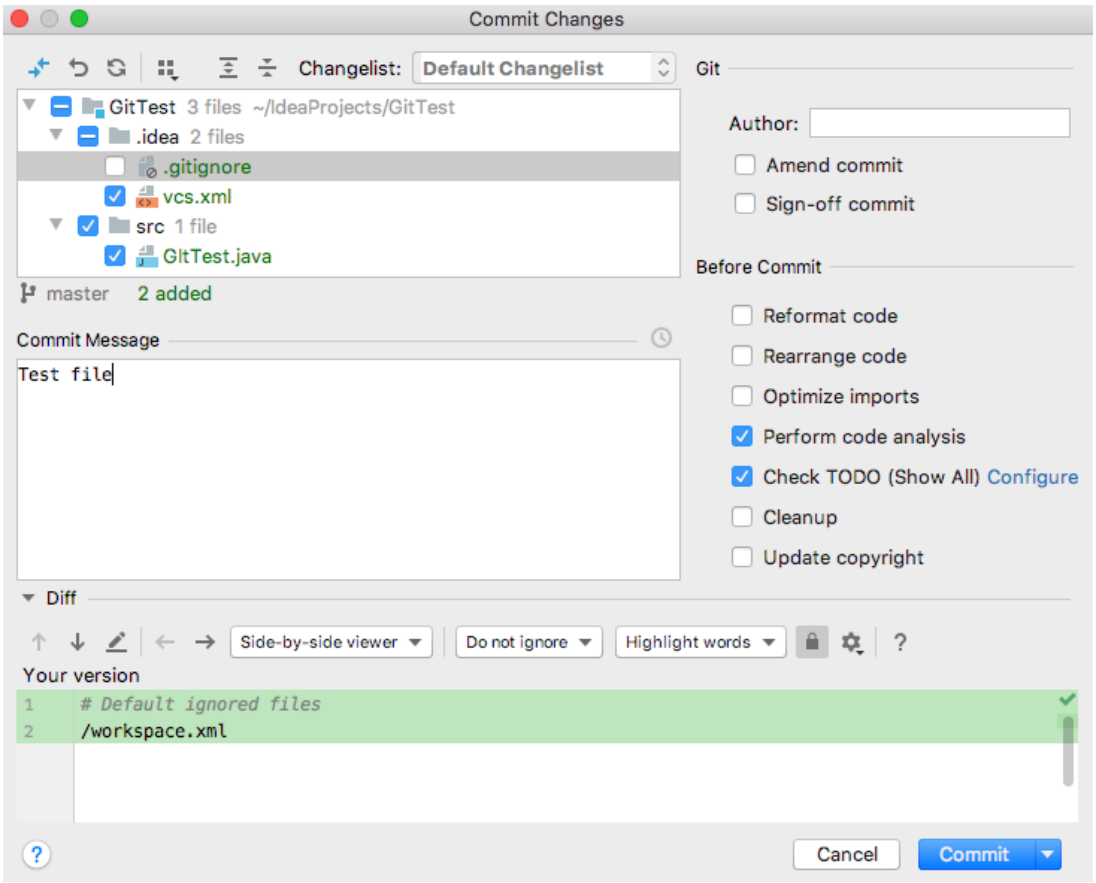
- Si queremos **crear un nuevo repositorio remoto,** debemos crear el *IntelliJ* proyecto primero, y luego elija *VCS> Importar a control de versiones> Compartir proyecto en GitHub*. Luego, se nos pedirá que escriba nuestro GitHub nombre de usuario y contraseña, y el nombre del repositorio (por defecto, IntelliJ sugiere el nombre de nuestro proyecto). Finalmente, podemos elegir los archivos para la confirmación inicial.







- Cuando queramos **cometer** nuevos cambios, elegimos *VCS> Confirmar ...* menú, y luego elegir qué cambios queremos confirmar, y el comentario asociado a este compromiso.



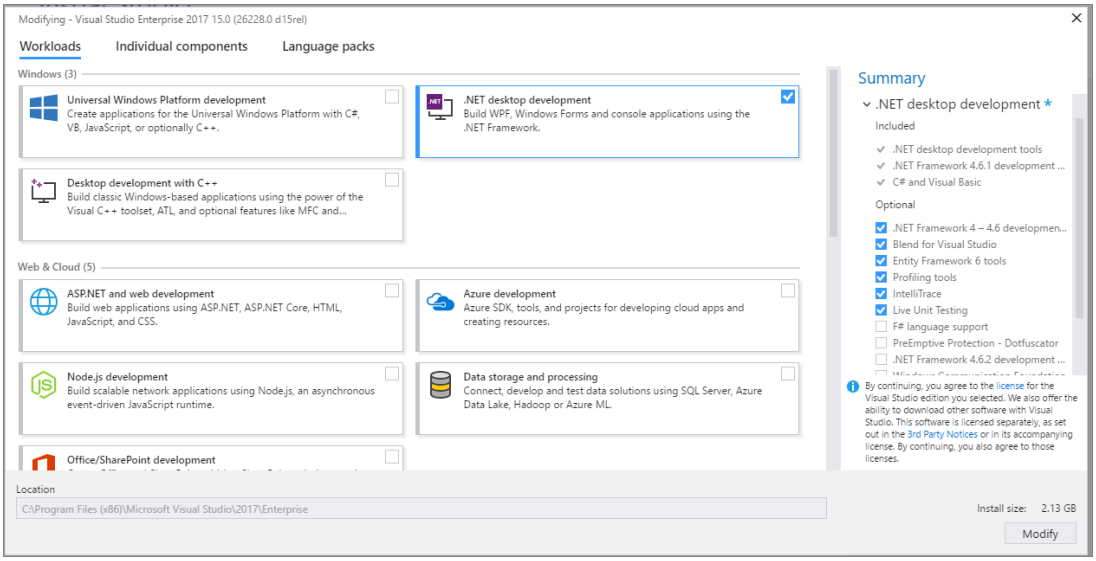
- Si queremos **empujar** un compromiso, o **Halar** los cambios del repositorio remoto, podemos encontrar estas opciones dentro *VCS> Git* submenú.

> **Ejercicios propuestos**
>
> **1.8.2.1.** Cree un nuevo proyecto Java local llamado *IntelliJGit,* y luego comparte este proyecto en tu cuenta de GitHub. Luego, agregue una nueva clase llamada Prueba , y comprometa e impulsa estos cambios.

1.8.3. Estudio visual

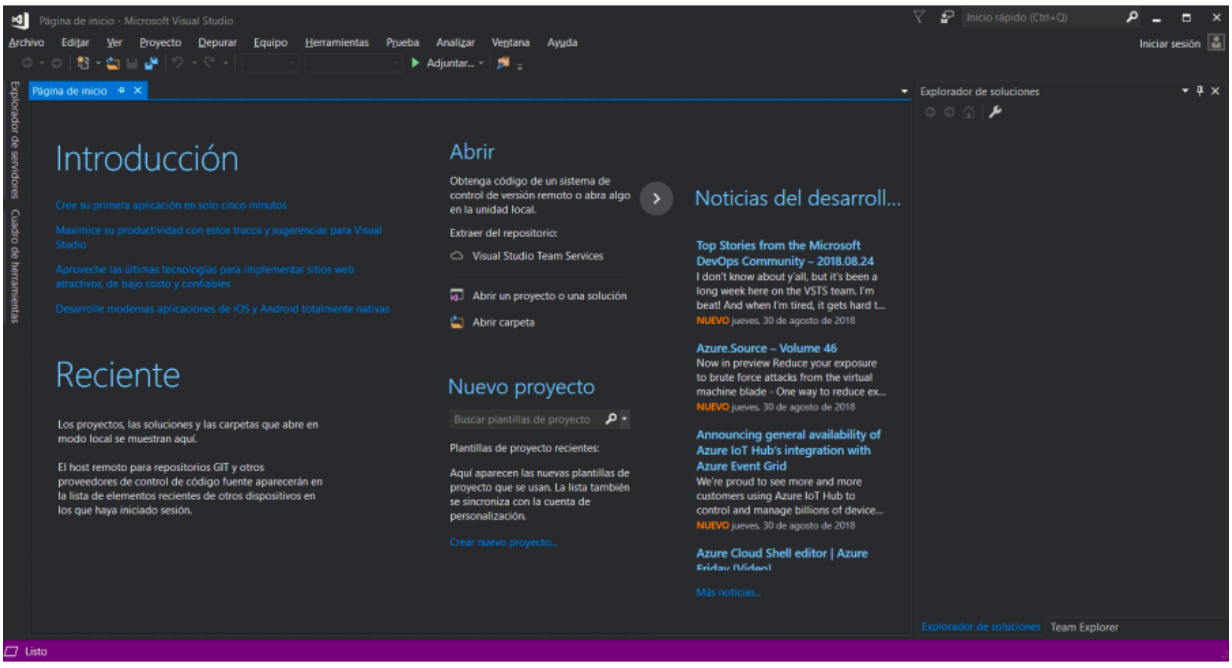
Estudio visual fue un IDE creado en 1997. Inicialmente fue creado para desarrollar aplicaciones C ++ y Visual Basic bajo Windows, pero ha evolucionado y ahora incluye algunos otros lenguajes, como C #, ASP.NET ... En general, puedes desarrollar cualquier tipo de aplicación soportada por la plataforma .NET, pero también hay algunos otros lenguajes que también son soportados, como Java o Python, entre otros

Existen diferentes distribuciones para Visual Studio: Community, Professional o Enterprise. El primero es gratis y puedes descargarlo aquí . Está disponible para sistemas Windows y MacOSX. Una vez que ejecute el instalador, debe elegir su carga de trabajo. Por ejemplo, si planea desarrollar aplicaciones de escritorio o de consola con C # y Windows Forms, puede elegir. *Desarrollo de escritorio .NET.*

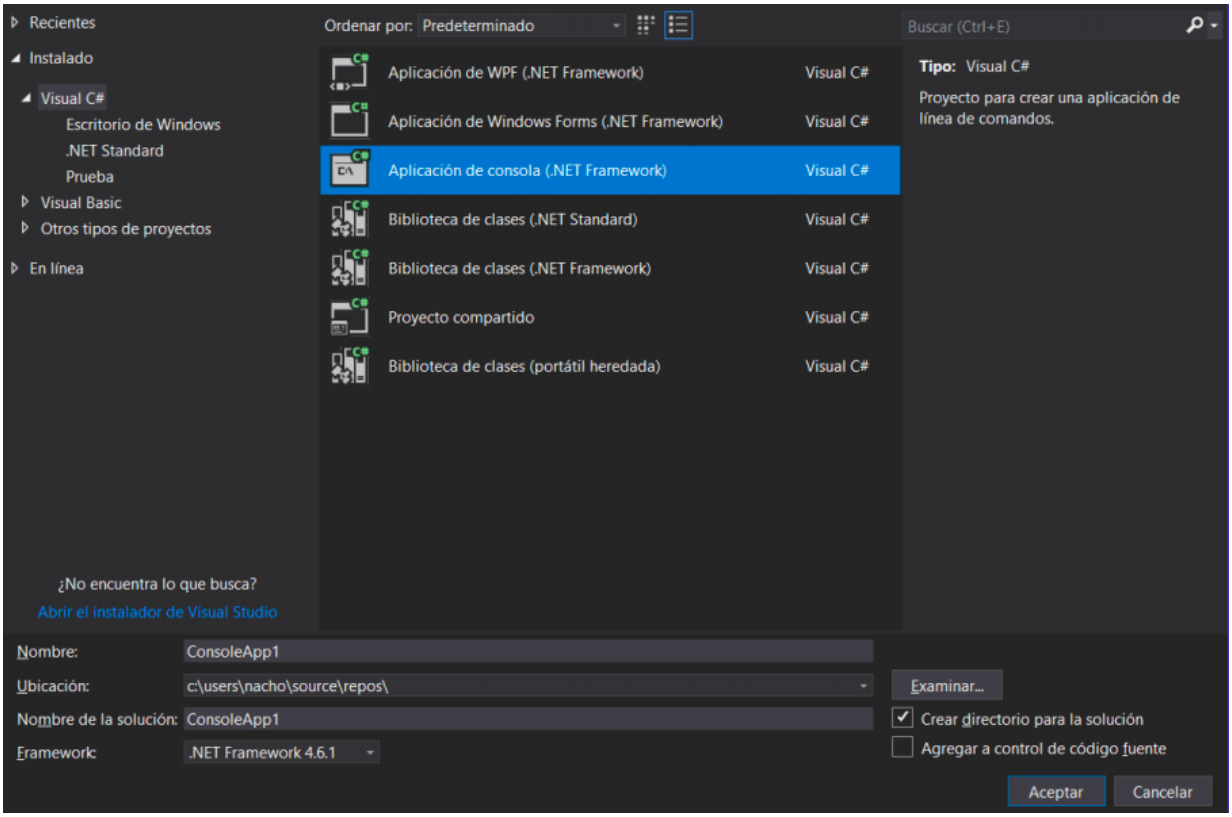


1.8.3.1. Creando proyectos

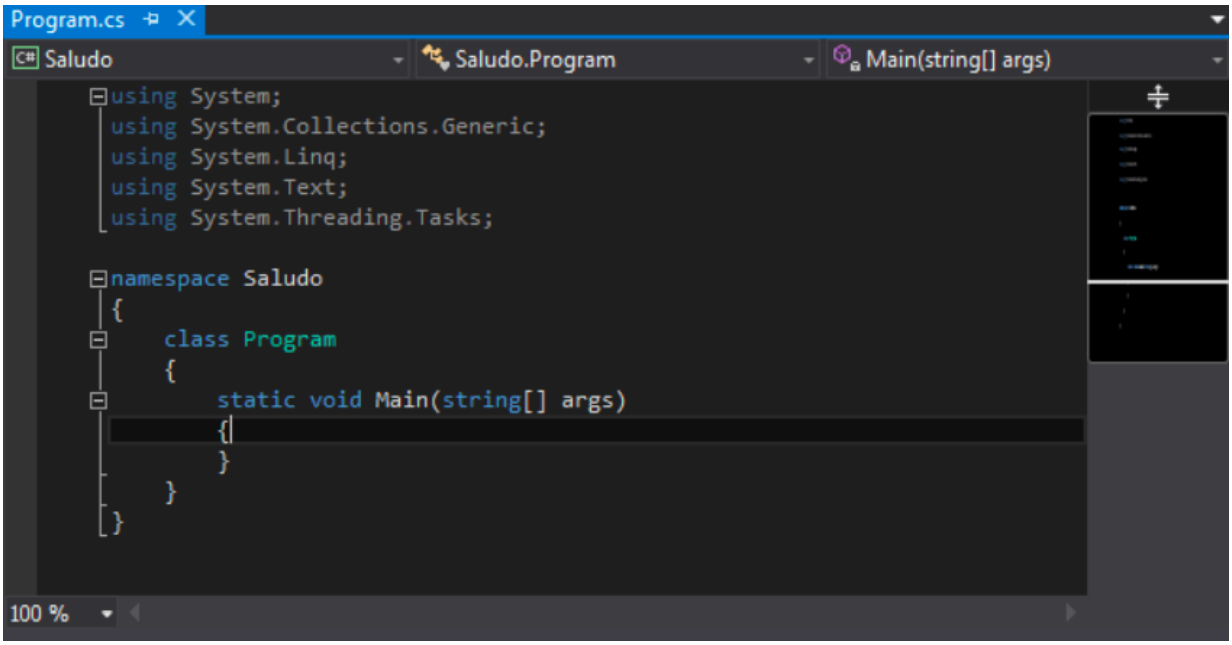
Esta es la pantalla de bienvenida de Visual Studio:



Si queremos crear un nuevo proyecto, vamos a *Archivo> Nuevo proyecto* menú. Luego, generalmente elegimos una aplicación de consola C #:



En el formulario inferior, debemos especificar el nombre del proyecto y la ubicación (podemos dejar la ubicación predeterminada). Luego, se mostrará un nuevo proyecto, con un archivo fuente inicial predeterminado llamado Program.cs , con algún código predeterminado ya escrito en él:



Para ejecutar el programa, simplemente hacemos clic en el *comienzo* en la barra de herramientas, o presione F5 , o Ctrl + F5 Si desea que el programa se detenga después de terminar, antes de cerrar el terminal.

1.8.3.2. Atajos

En las siguientes tablas, puede ver algunos de los atajos de Visual Studio más comunes (en Windows).

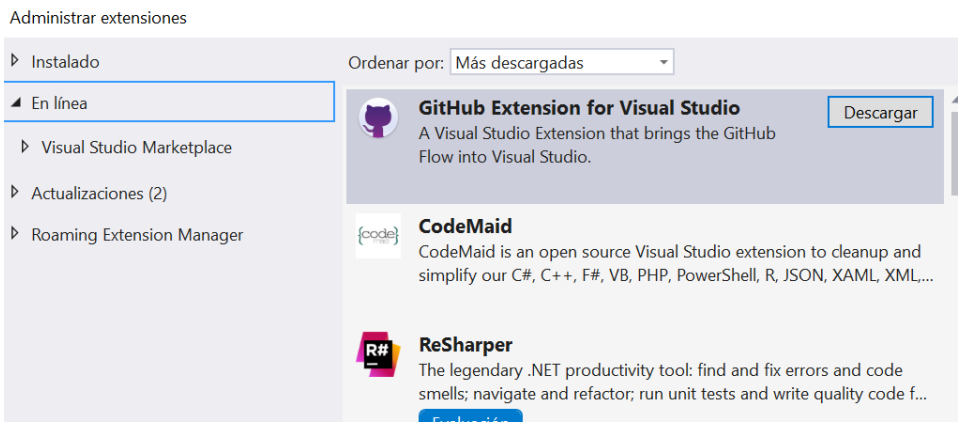
Atajo	Acción
Ctrl + Cambio + N Ctrl + norte	Crea un nuevo proyecto. Crea un
	nuevo archivo. Guarda el archivo
Ctrl + S	actual.
Ctrl + Cambio + S Ctrl + C / V	Guarde todos los archivos abiertos. Copia,
/ X Ctrl + Z	pega y corta texto. Deshace la última
	acción.
Ctrl + Y	Rehacer la última acción deshecha.
Ctrl + F	Mostrar diálogo de búsqueda.
Ctrl + H	Mostrar diálogo de reemplazo.

Atajo	Acción
Ctrl + L	Quite la línea.
Ctrl + R	Cambiar el nombre del elemento seleccionado.
F5 / Ctrl + F5	Ejecute la aplicación con / sin depuración. Reconstruye el
Ctrl + Cambio + B F11 / F10	proyecto.
	Paso a la función / siguiente paso (en *depurar* modo). Función de salida
Cambio + F11	(en *depurar* modo).

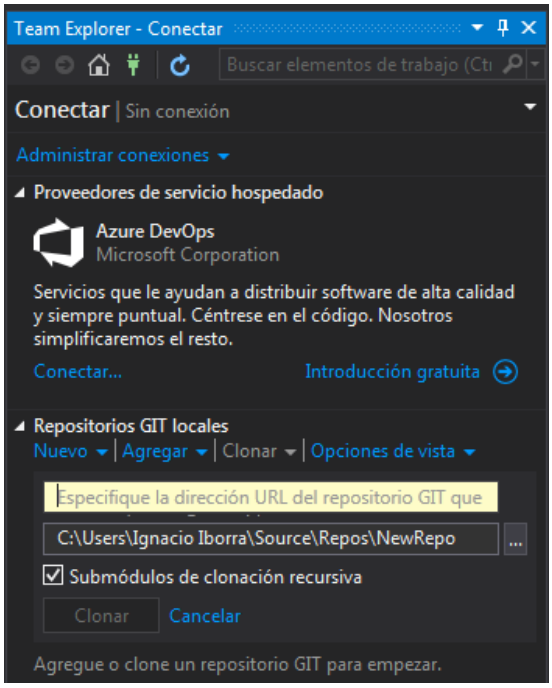
Respecto a MacOSX, debes reemplazar Ctrl clave con Cmd llave. Puedes encontrar más atajos aquí

1.8.3.3. Integración con Git

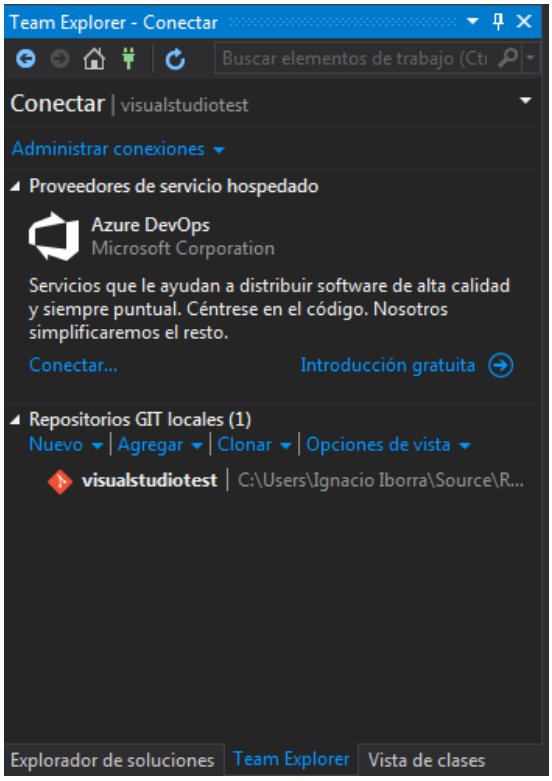
Visual Studio tiene una opción para lidiar con repositorios de Git. En primer lugar, es posible que necesitemos instalar la extensión GitHub para Visual Studio, desde *Extensiones> Administrar extensiones* menú:



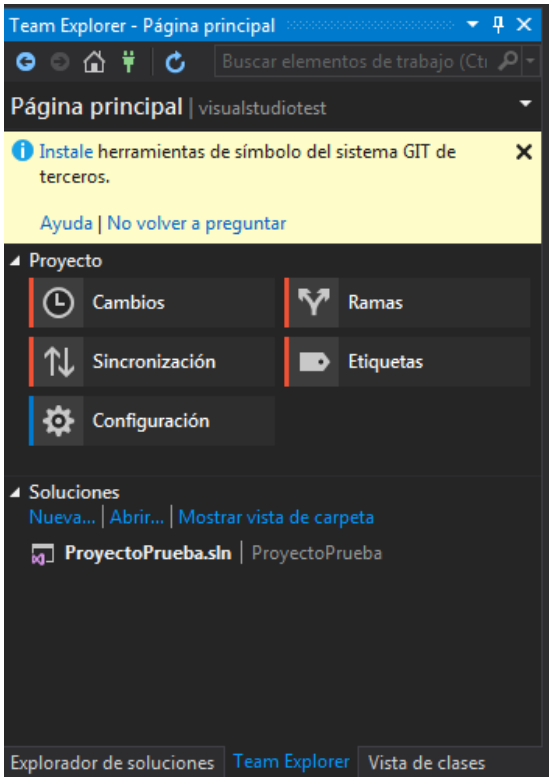
Luego, si queremos comunicarnos con un repositorio remoto, debemos ir a *Equipo> Administrar conexiones*
menú. La primera vez que elijamos esta opción, necesitaremos iniciar sesión en GitHub para otorgar algunos permisos a Visual Studio. Luego, podemos crear un repositorio de Git local o clonar uno remoto.

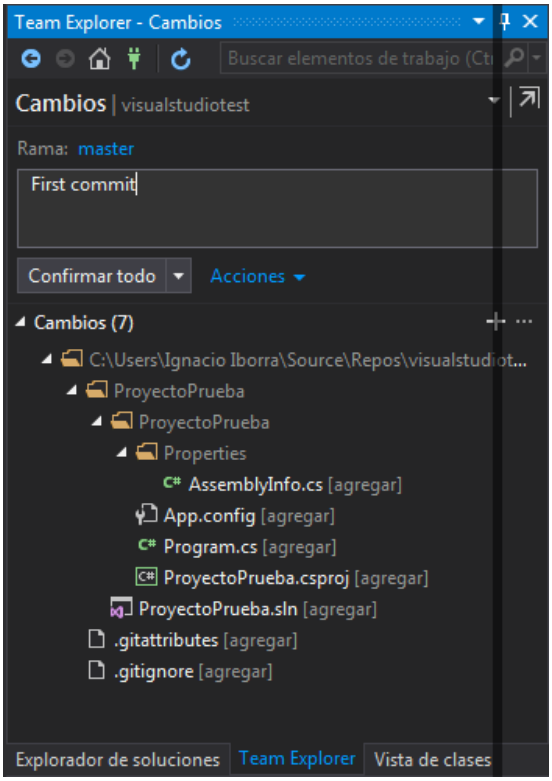


Si elegimos el *Clon* opción, necesitamos especificar la URL remota que se clonará y la carpeta local para descargarla. Luego, podemos agregar proyectos a este repositorio y realizar cambios. Cada vez que queremos cargar estos cambios, debemos hacer doble clic en el repositorio:



Luego, podemos hacer clic en el *Cambios* opción para ver todos los cambios que están pendientes de cargar. Podemos agregar un comentario para todos ellos y confirmar el compromiso:





Finalmente tenemos que elegir el *Sincronizar* opción para cargar (*empujar)* los cambios en el repositorio remoto. Es posible que se nos solicite que ingresemos nuestras credenciales para permitirnos impulsar el contenido.

También podemos utilizar esta opción al inicio de nuestra sesión, para descargar (*Halar)* los contenidos actualizados desde el repositorio remoto, para que podamos realizar nuestros cambios y subirlos desde una versión previamente actualizada.

> **Ejercicios propuestos**
>
> **1.8.3.1.** Cree un nuevo repositorio remoto en su cuenta de GitHub o Bitbucket llamado *VisualStudioTest.*
> Clónelo en Visual Studio, agregue un nuevo proyecto en él y cargue los cambios de Visual Studio en el repositorio remoto.

1.8.3.4. Otras características

Hay otras configuraciones que se pueden configurar desde *Herramientas> Opciones* menú. Por ejemplo, podemos mostrar / ocultar los números de línea del *Editor de texto* subsección, para cada lenguaje específico (C #, Básico ...), o las líneas verticales de sangría para ver fácilmente los límites de un Si o para declaración...

1.8.4. Aprender más...

Puede obtener más información sobre todos estos IDE en los siguientes enlaces

- IntelliJ IDEA
- IntelliJ IDEA (Wikipedia)
- Estudio visual
- Descarga Visual Studio

- shortcutworld.com (base de datos general con accesos directos para muchos IDE)