

1.-Introducción. En la presente unidad de trabajo se ha estudiado como instalar y usar un entorno de desarrollo. Se ha visto:

- Cómo instalarlo.
- Configurar y personalizar el entorno de desarrollo.
- Crear un proyecto, insertar código y ejecutarlo.
- Añadir y eliminar un plugin.

2.-Práctica. Busca cuales son los IDE más usados actualmente (que sirvan para programar en Java). Elige uno de ellos y haz una guía de usuario para manejar dicho entorno. No puede ser ni NetBeans ni Eclipse.

En 2022 los 7 mejores IDEs para Java son:

1. Eclipse

Es un IDE muy popular de código abierto y ampliamente extendido. Fue desarrollado originalmente por IBM aunque actualmente cuenta con una gran comunidad que le da soporte con una gigantesca biblioteca de plugins. Es multiplataforma (Windows, Linux y Mac) y dispone de nuevas versiones cada año. Con Eclipse es posible desarrollar todo tipo de aplicaciones para web, escritorio o móviles usando Java, C, C++, JSP, perl, Python, Ruby y PHP.

Su mayor punto fuerte reside en su asistente y capacidad para depurar errores. A medida que vas programando Eclipse lanza recomendaciones que nos ayudan a optimizar nuestro código hasta niveles que rozan el elegante guante blanco. Sin duda una excelente elección.

2. NetBeans

NetBeans es otro entorno de desarrollo integrado basado en código abierto. En lo personal adoro este IDE porque gracias a su enfoque podemos desarrollar nuestras aplicaciones como módulos (como un componente de un software). Lo cual resulta sumamente cómodo pues podemos reutilizar nuestro código para diferentes proyectos las veces que queramos sin mayor complicación. NetBeans es el IDE oficial de Java pero también podemos desarrollar en otros lenguajes como PHP, C, C++ e incluso HTML 5.

Entre sus detractores hay quien comenta que el desarrollo de una GUI sobre NetBeans es algo complicado y que no cuenta con la ingente cantidad de plugins y extensiones de las que sí cuenta Eclipse, además de carecer de soporte nativo para trabajar con webapps (.war, .jsp, y servlets).

3. IntelliJ Idea

Desarrollado por Jet Brains es un IDE que cuenta con dos versiones.

La Community Edition, libre descarga, y la versión Ultimate Edition que cuesta la friolera de más de 500 dólares de suscripción anual, según el plan que contratemos, pero que podemos probar gratis durante 30 días si lo que queremos es echarle un vistazo a todo lo que puede ofrecer. Con IntelliJ podemos programar sobre diferentes lenguajes y proporciona soporte para trabajar con JAVA, Node JS, PHP, Python, Ruby, Sass, TypeScript, AngularJS, CoffeeScript, CSS, HTML, JavaScript, LESS, etc.

Además ofrece integración con GIT lo que permite trabajar con diferentes versiones de nuestro software sin afectar a la línea maestra de nuestro desarrollo.

4. BlueJ

Estoy seguro que has visto este IDE antes. Está presente en innumerables cantidades de vídeos y tutoriales en Youtube. La principal razón es que en cierto momento James Gosling, el creador de Java, dijo que BlueJ era uno de sus IDE favoritos. Esto desencadenó que de pronto en muchas escuelas y academias se usará BlueJ para dar clase pues era el IDE del momento.

Como los IDE anteriores de esta lista, es multiplataforma. Podemos usarlo sobre Windows, macOS y GNU/Linux y cuenta con una gran documentación que lo hace especialmente indicado para estudiantes.

Su principal característica es que nos permite generar una vista UML de todos los objetos que componen nuestra aplicación.

5. Codenvy

Codenvy es un IDE comercial de Java que está construido sobre el Eclipse Che. Como muchas otras opciones de la competencia podemos encontrar diferentes variantes en función del tipo de desarrollador o cliente que seamos. Entre las 3 variantes podemos elegir entre la edición developer, enterprise y team.

Lo que hace interesante a Codenvy es que está pensado para ofrecer un espacio de trabajo en la nube, colaborativo y bajo demanda. Esto hace que sea muy fácil moverse de una licencia a otra y aprovechar solo aquellas características que necesitemos ahorrando costes.

Desde este IDE podemos trabajar cómodamente desde nuestro navegador y beneficiarnos de potentes funciones como la finalización de código, la refactorización y el resaltado de sintaxis.

Además de Java, Codenvy proporciona soporte para una gran cantidad de lenguajes de programación interpretados y compilados. Esto incluye C ++, C #, JavaScript y PHP.

6. Xcode

Los usuarios de Mac que quieran usar un IDE completamente gratis deben probar Xcode. Se trata de un IDE desarrollado para macOS por la propia Apple que además de permitir la programación en Java, el IDE viene con una multitud de herramientas que nos permiten desarrollar software para iOS, macOS, tvOS y watchOS.

Algunos de los aspectos más destacados importantes de Xcode son un depurador incorporado, un generador de GUI y un autocompletado de perfiles. Además de Java, Xcode proporciona soporte para AppleScript, C, C ++, Objective-C, Objective-C ++, Python, Ruby y Swift.

7. jGRASP

Por último, el séptimo de la lista pero no por ello el peor. Los anteriores IDE tienen un “defecto” común. Todos devoran ingentes cantidades de memoria RAM. Si buscas algo ligero para comenzar tu desarrollo quizá jGRASP pueda ayudarte.

Desarrollado en JAVA por el Samuel Ginn College of Engineering de la Universidad de Auburn en su departamento de ciencias de la computación e ingeniería de software. Funciona bien en todas las plataformas con JVM. Cuenta además con una función muy similar a la que comentaba con BlueJ, proporcionando visualizaciones de software CSD, Diagramas de Clase UML, así como vistas Dinámicas y vistas Canvas. Produce CSD (diagramas de estructuras de control) para Java. Con jGRASP podemos desarrollar además aplicaciones basadas en C, CPP, Objective-C, Python, ADA y VHD.

Sin embargo, no voy a utilizar ninguno de los mencionados y voy a instalar Visual Studio porque es un IDE extremadamente popular, compatible con Java y me ha creado curiosidad.

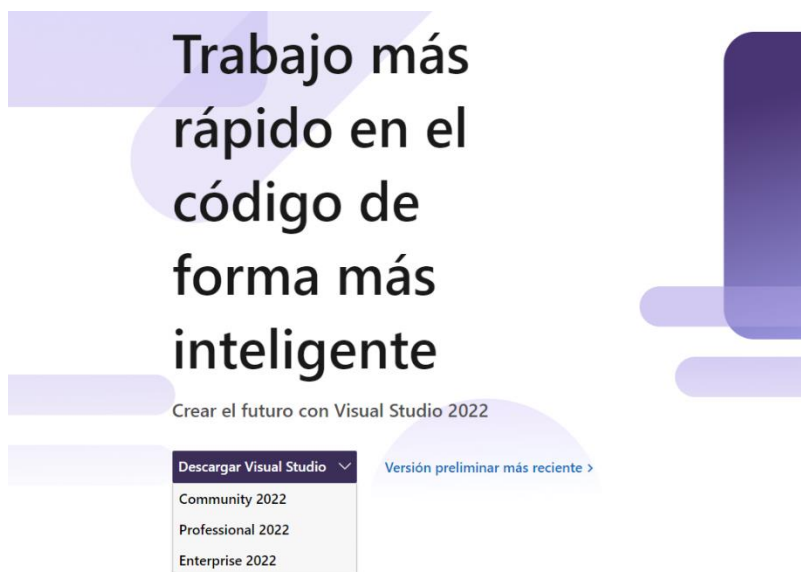
Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónico.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

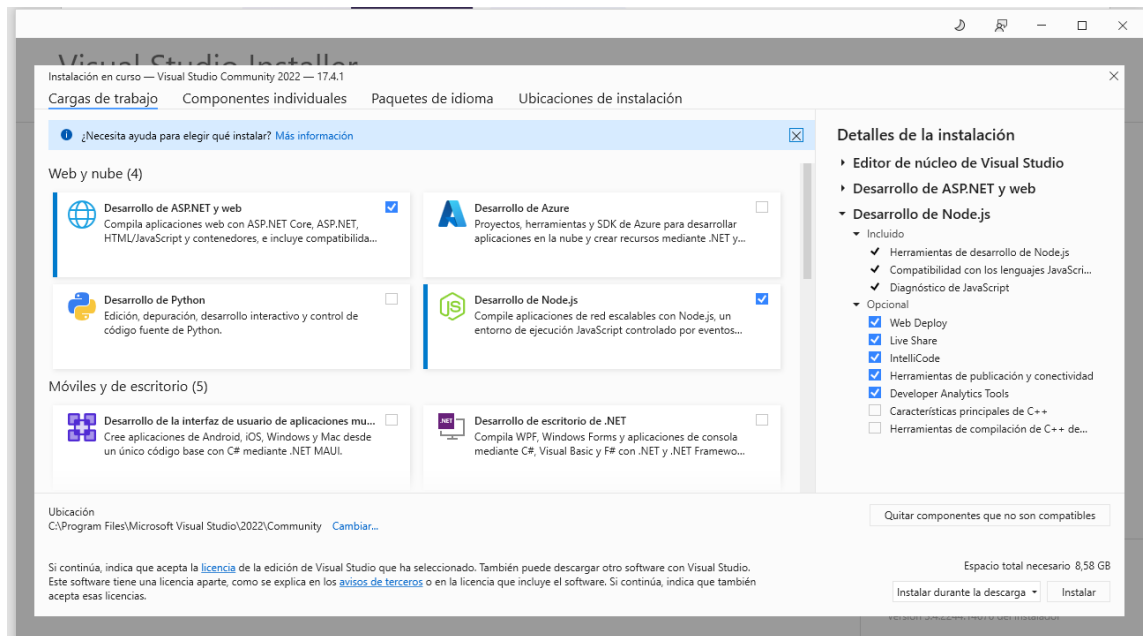
- Instalación:

Nos vamos a la página oficial y descargamos la versión “Community 2022” que es la gratuita.



Después de ejecutar el instalador tendremos que elegir qué queremos descargar ya que pueden ser aplicaciones móviles, de escritorio, web, videojuegos. Podemos seleccionar desde ‘Cargas de trabajo’ o ‘Componentes individuales’, para lo segundo se requiere un

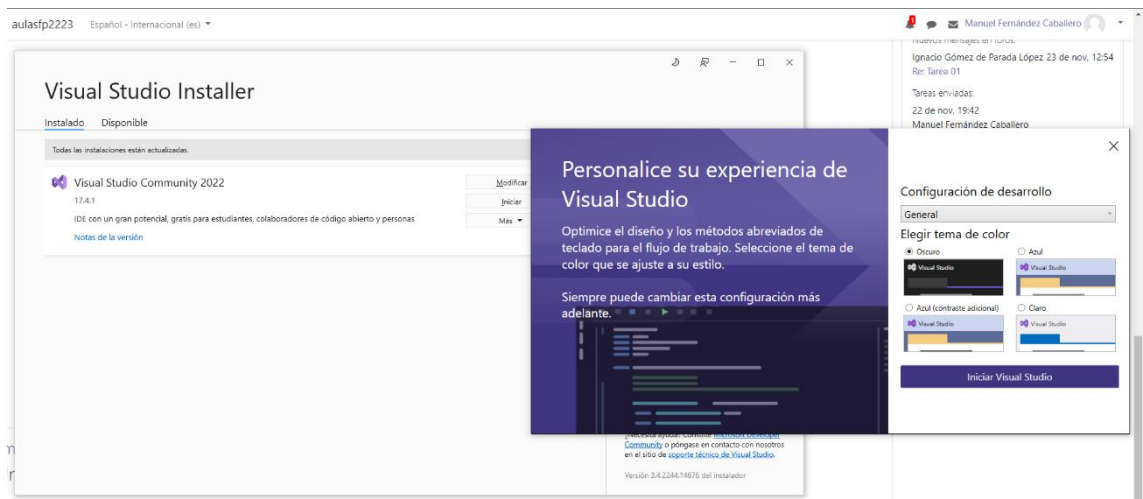
poco de experiencia trabajando con Visual Studio para saber lo que necesitas así yo lo seleccionaré desde 'Cargas de trabajo'.



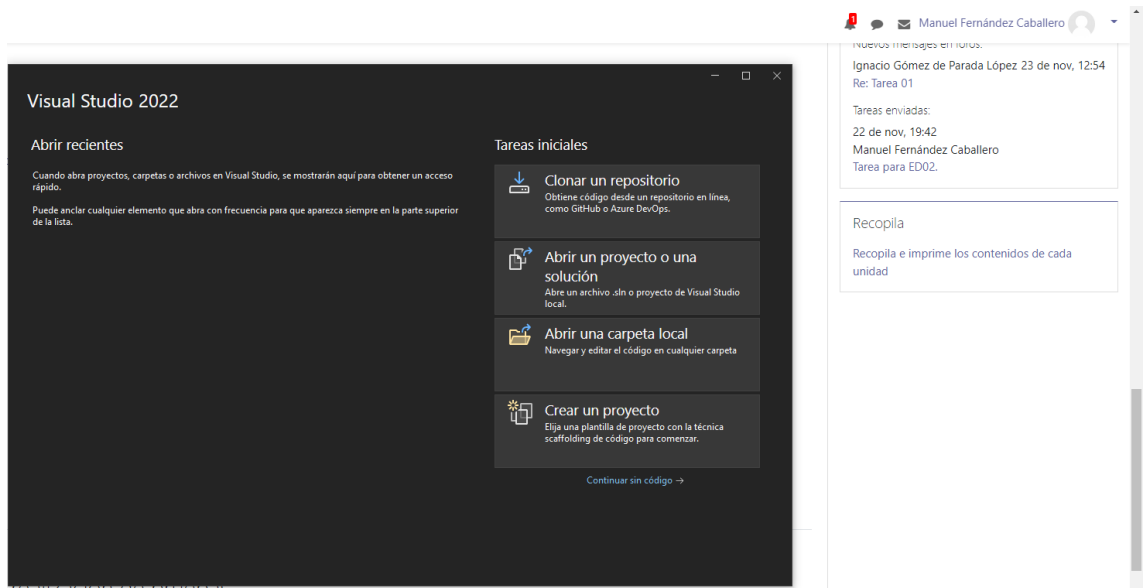
Yo he seleccionado ASP para desarrollo Web y Node, ambos compatibles con JavaScript y pulsamos en instalar. Se iniciará la descarga para la posterior instalación de los complementos seleccionados.

Al terminar aparecerá la ventana para iniciar Visual Studio donde también podremos modificar el tema, yo lo suelo poner oscuro e iniciamos Visual Studio.

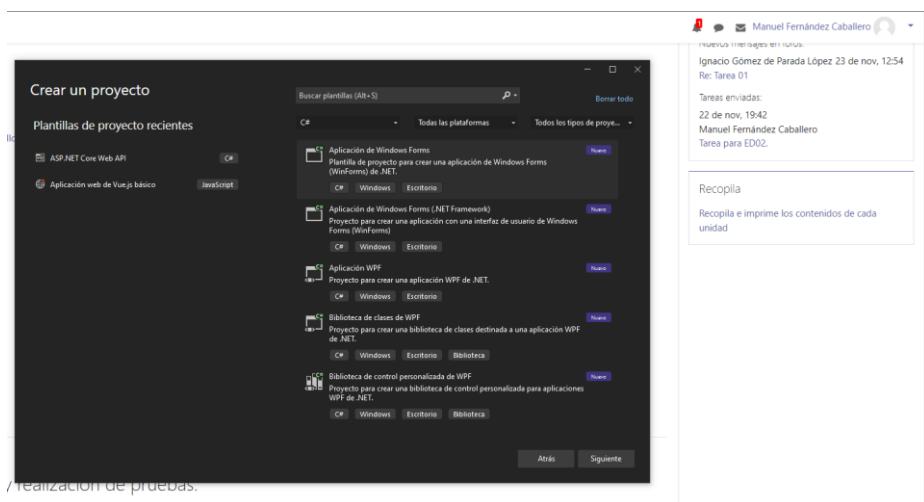
- Configurar y personalizar el entorno de desarrollo.



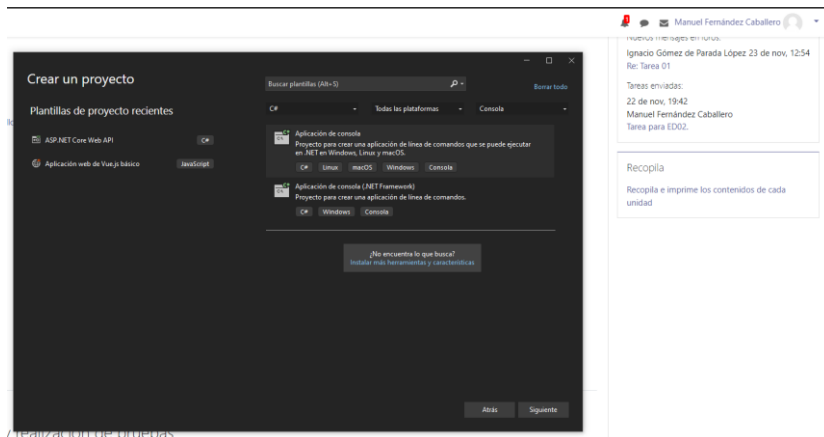
Nos saldrá una ventana donde podremos crear un nuevo proyecto o clonar un repositorio. Hacemos clic en crear Proyecto donde seleccionamos tipo de proyecto, basado en los paquetes instalados.



Voy a utilizar C# ya que me requería las dependencias de npm y npx para JavaScript.

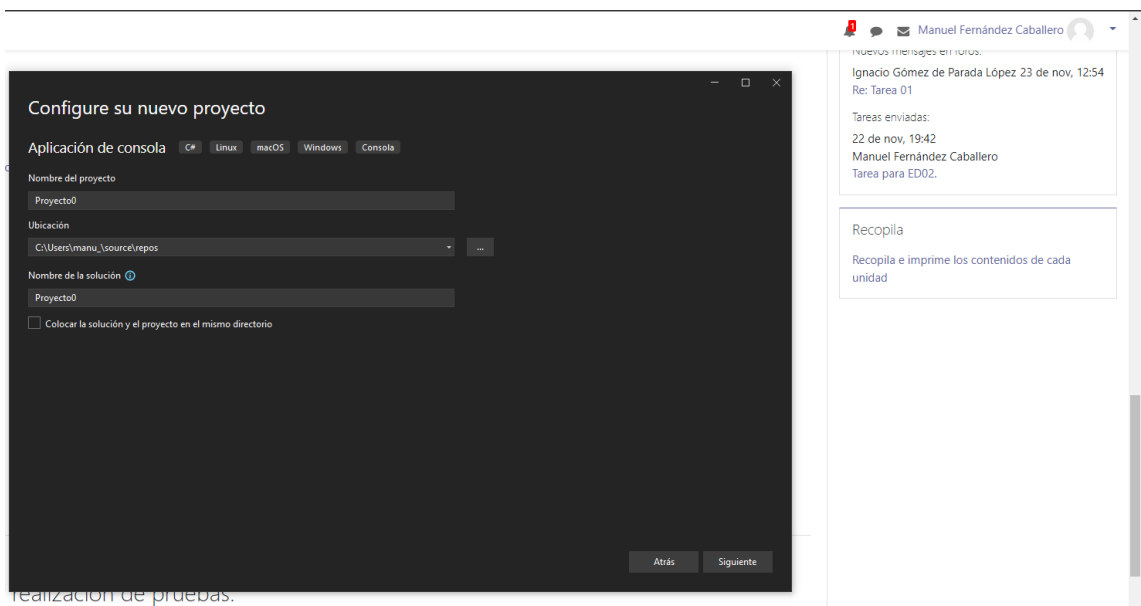


Y seleccionamos aplicación de consola para este proyecto.



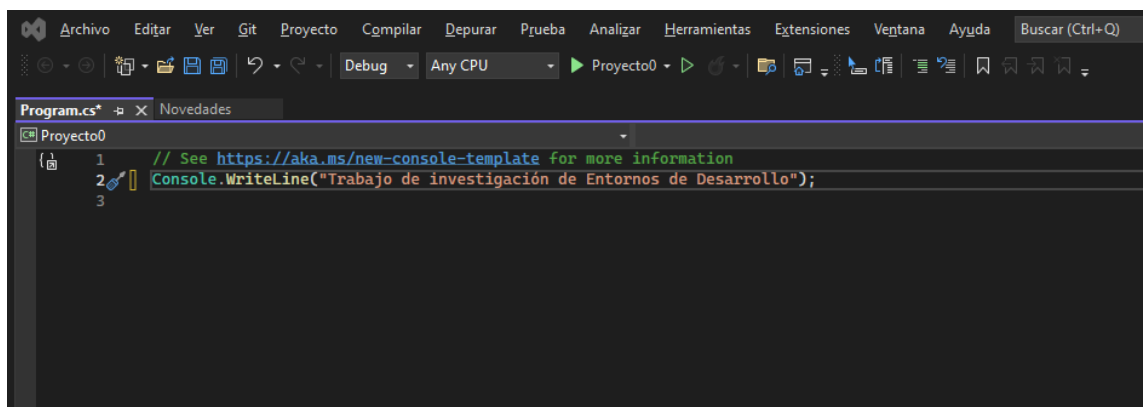
Y marcamos Colocar la solución y el proyecto en el mismo directorio.

- Crear un proyecto, insertar código y ejecutarlo.

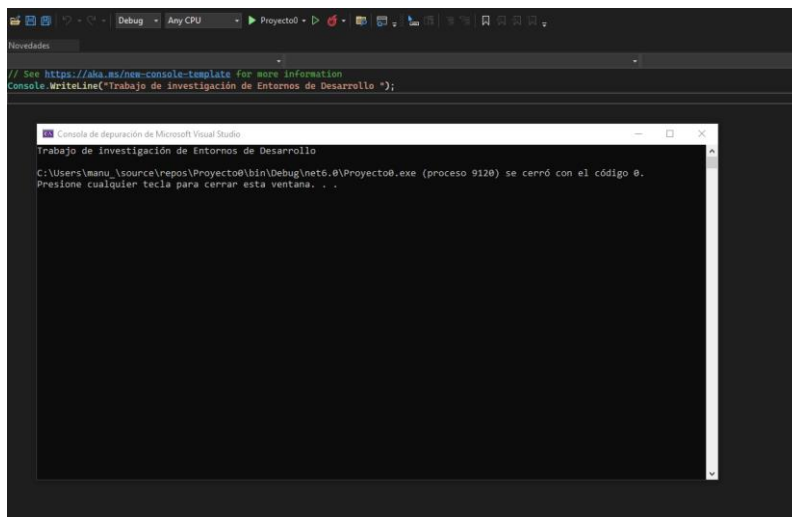


realización de pruebas.

Escribimos un código sencillo que nos imprima por pantalla el siguiente texto:

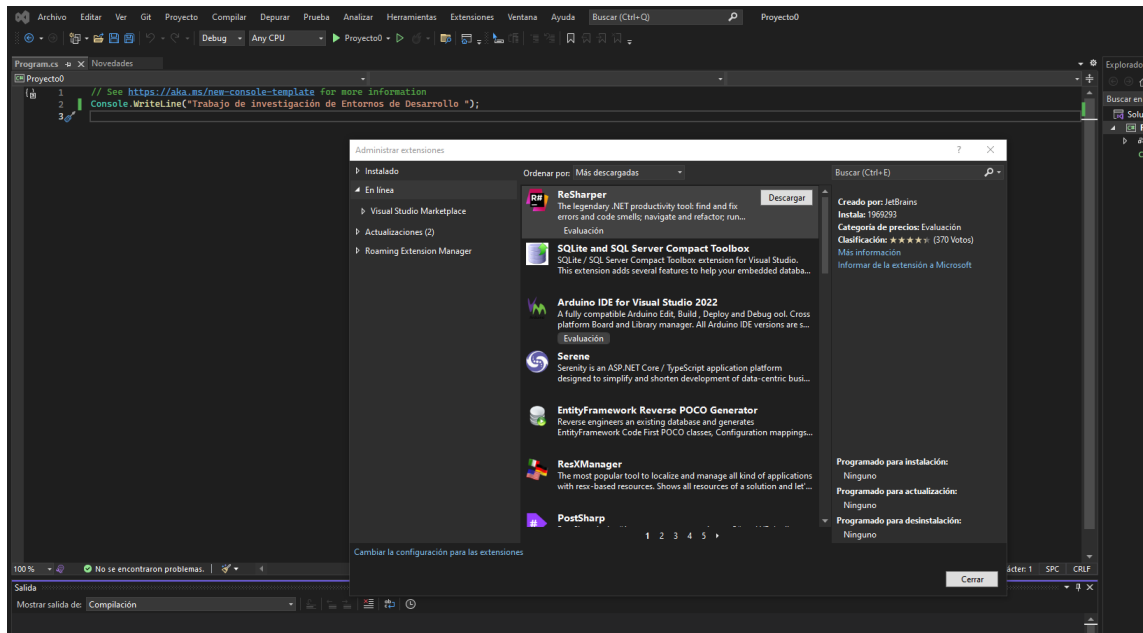


Y al ejecutarlo y compilarlo, como se trata de una aplicación de consola recibimos lo siguiente:

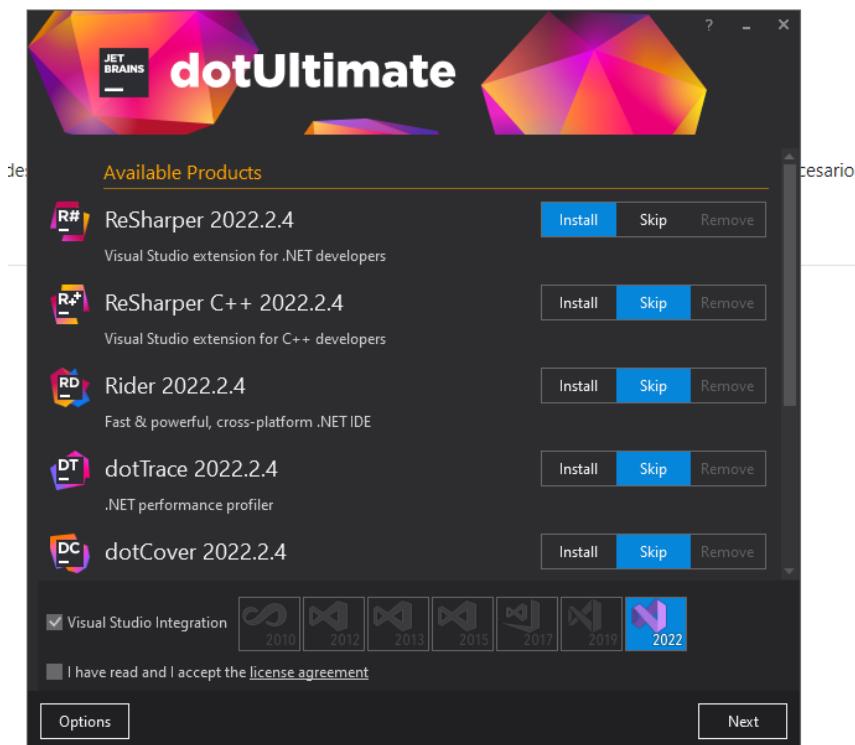


- Añadir y eliminar un plugin.

Hacemos click en Extensiones>Administrar extensiones



Descargamos en este caso ReSharper, ejecutamos e instalamos las funciones deseadas:



Aceptamos la licencia e instalamos. Si queremos dejar de usar una extensión, puede deshabilitarla o desinstalarla. Al deshabilitar una extensión esta sigue instalada, pero está descargada. Buscamos la extensión en Extensiones>Administrar extensiones y hacemos clic Desinstalar o Deshabilitar.