

DAM

Contornos de desenvolvimento

UD2 **INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO**

PROFESORA | CRISTINA PUGA FDEZ

UNIDADE | UD2

TRIMESTRE | 1

MÓDULO | CD

Tabla de contenido

1.1	Características de los entornos de desarrollo	3
1.1.1	Funciones, herramientas y asistentes de un entorno de desarrollo	3
1.1.2	Entornos de desarrollo más utilizados	5
1.2	Entorno de desarrollo libre NetBeans	7
1.2.1	Instalación	7
1.3	Entorno de desarrollo propietario: VisualStudio.....	10
1.3.1	Instalación	10
1.3.2	Inicio	11
1.3.3	Primeros pasos	14
	ANEXO I: Materiales	15

1. Instalación de entornos de desenvolvimiento

1.1 Características de los entornos de desarrollo

Un entorno de desarrollo o IDE (Integrated Development Environment) o EDI (Entorno de Desarrollo Integrado) es una aplicación informática que asiste al técnico informático en la tarea de diseñar y codificar software mediante la inclusión de múltiples herramientas integradas relacionadas. Este tipo de aplicación se utiliza en diferentes momentos del desarrollo de software: en el diseño, en la codificación y en las pruebas.

La característica más importante es que consta de diferentes herramientas integradas, es decir, los resultados de una herramienta pueden ser utilizados para otra e quedarán relacionados. Por ejemplo, un programa Java y su correspondiente programa de pruebas unitarias en JUnit estarán relacionados, de tal forma que el entorno avisará si un cambio en el programa Java afecta al programa de pruebas; lo cual no sucedería si se codificase en un editor de texto.

Otra característica es que se pueden hacer muchas operaciones sin salir de la misma aplicación y siguiendo siempre la misma forma de trabajar (ventanas, menús, combinaciones de teclas, etc.).

Para crear un programa sólo es imprescindible un editor de textos sencillo y para ejecutarlo, sólo es imprescindible el traductor correspondiente, pero es muchísimo mejor tener una aplicación desde la cual se puedan realizar operaciones relacionadas, como por ejemplo: diseñar los diagramas UML que servirán de base para la codificación; generar código fuente mediante sugerencias o que se complete automáticamente; analizar y depurar código para detectar errores; ejecutar y probar programas; etc. y todo desde la misma aplicación.

1.1.1 Funciones, herramientas y asistentes de un entorno de desarrollo

Las funciones más destacadas de un entorno de desarrollo son:

- **Edición del código fuente:** el editor de texto realiza operaciones generales de edición como buscar, sustituir, copiar, pegar, etc. y además, tiene características diseñadas para simplificar y acelerar la escritura de código fuente como por ejemplo, resalta y colorea instrucciones según

su tipo, marca errores sintácticos, sugiere instrucciones, realiza completado automático de instrucciones al estar escribiendo, o tabulado automático de instrucciones.

- **Análisis de código:** el analizador de código puede ser estático o dinámico. El analizador estático hace el análisis léxico, sintáctico y semántico del código en tiempo de codificación y detecta ese tipo de errores. El analizador dinámico, valora el comportamiento del código al mismo tiempo que se está ejecutando y obtiene información como, por ejemplo: tiempo invertido en la ejecución o la ocupación de memoria, que permitirá optimizar el funcionamiento del programa.
- **Depuración de código:** El depurador es un programa utilizado para eliminar errores en otros programas (programa “objeto”). El código se ejecuta en un simulador de instrucciones (ISS) que permite detenerse cuando se verifiquen ciertas circunstancias establecidas por el programador. En ese momento, el programador podrá conseguir más información para localizar el error y solucionarlo.
- **Generación de código ejecutable:** Según el tipo de lenguaje utilizado (compilado, interpretado, etc.), el entorno de desarrollo estará configurado para localizar el compilador, las librerías y los intérpretes que necesite. Algunos entornos de desarrollo tienen la posibilidad de tener compiladores cruzados, es decir, compiladores para una plataforma diferente de sobre la que funciona el entorno de desarrollo.
- **Pruebas:** el entorno de desarrollo facilita la creación de pruebas unitarias o de integración, permite la ejecución de estas y emite un informe sobre el resultado.
- **Control de versiones:** La herramienta de control de versiones facilita la administración de las distintas versiones o especializaciones de un producto. Una versión es el estado en el que se encuentra un producto en un momento dado de su desarrollo. El control de versiones es imprescindible cuando se lleva a cabo un proyecto de desarrollo con la colaboración de varias personas.
- **Construcción de interfaces gráficas de usuario:** Las GUI (Graphical User Interface) son interfaces gráficas de usuario y proporcionan un entorno visual sencillo para permitir la comunicación del usuario con el sistema operativo mediante imágenes y objetos gráficos que representan la información y las acciones disponibles en la interfaz. Un entorno de desarrollo

puede tener un constructor GUI que simplifica el proceso de diseño y construcción de la GUI de nuestro desarrollo.

- **Generación de documentación:** el generador de documentación permite crear esta de forma automática a partir del código fuente.
- **Refactorización del código:** el entorno de desarrollo dispone de herramientas que permiten cambiar el código fuente alternando la estructura interna sin cambiar el comportamiento externo. El programador indica el tipo de refactorización que quiere realizar y la herramienta lo llevará a cabo de forma segura sin intervención del programador.

1.1.2 Entornos de desarrollo más utilizados

Existen muchos entornos de desarrollo: según tengan licencia propietaria o libre; plataformas sobre las que funcionan; lenguajes que permiten; herramientas disponibles; etc. Todo esto, hace difícil indicar los entornos de desarrollo más utilizados ya que ninguno es óptimo para todas las necesidades y por otro lado, el mejor entorno para un desarrollador es aquel que conoce en profundidad por lo que le puede sacar más provecho.

Visual Studio tiene licencia propietaria de Microsoft, solo sirve para la plataforma .NET de Windows y permite codificar en C/ C++, C#, JavaScript e Python.

NetBeans y Eclipse tienen licencias libres, pueden utilizarse en las plataformas Windows, Mac OS X y Linux, que permiten codificar en C/ C++, Java, PHP, JavaScript y Python. Ambos tienen muchas características comunes y permiten trabajar de forma muy parecida.

1.1.2.1 NetBeans

Empezó como un proyecto estudiantil en la República Checa en 1996 bajo la tutoría de la Facultad de Matemáticas y Física de la Universidad Carolina en Praga y originalmente se llamó Xelfi. En junio de 2000, Sun Microsystems patrocina NetBeans y lo declara de código abierto. En 2010 Oracle adquiere Sun.



El proyecto NetBeans está formado por dos productos: el IDE y la plataforma, ambas gratuitas y de código abierto para que cualquiera pueda utilizarlo dentro de los términos de uso que se detallan en la página oficial: <http://netbeans.org>

La mayoría del código de NetBeans IDE está disponible bajo una licencia dual CDDL (*Common Development and Distribution License*) v1.0 y GNU GPL (*General Public License*) v2 que lo hace un producto de código abierto, gratuito y sin restricciones de uso.

NetBeans IDE está escrito en Java pero puede servir para programar en otros lenguajes de programación como: Java, PHP, C/C++, XML, HTML, Groovy, JavaScript y JSP (tecnología Java para páginas web) y mediante complementos se pueden utilizar más lenguajes de programación. Puede ser instalado sobre cualquier sistema operativo que soporte Java: Windows, Linux, MacOS X, o Solaris.

1.1.2.2 Eclipse

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. En el año 2003 fue creada la Fundación Eclipse que es una organización independiente y sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. La página oficial es <https://www.eclipse.org/>.



Permite codificar en Java así como en otros lenguajes como C, C++, Python, Perl, PHP y Ruby. Igual que NetBeans, mediante complementos puede utilizar más lenguajes.

Eclipse está disponible bajo una licencia *Eclipse Public License* v 1.0¹. Esta licencia permite utilizar, modificar, copiar y distribuir las versiones modificadas y en algunos casos los cambios también tendrán que ser libres. Este tipo de licencia está aprobada por la *Free Software Foundation* y la *Open Source Initiative* como licencia de software libre.

1.1.2.3 Visual Studio

Visual studio es un entorno de desarrollo con licencia propietaria de Microsoft para la plataforma .NET. Permite crear aplicaciones en Visual Basic, Visual C#, Visual C++ y JScript, además de otras. La página oficial es: <https://visualstudio.microsoft.com/es/>



Visual Studio permite compilar el código fuente en un código intermedio denominado CIL (*Common Intermediate Language*) similar al BYTECODE de Java. Para ejecutar este código se

¹ Ver más información en <https://www.eclipse.org/org/documents/>

necesita el framework .NET que dispone del entorno de ejecución CLR (*Common Language Runtime*) que mediante un compilador JIT (*Just-In-Time*) realiza el paso a código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue independencia de la plataforma de hardware.

1.1.2.4 Mono

Mono es una plataforma de desarrollo de código abierto basado en el framework .NET, que permite a los desarrolladores construir aplicaciones para sistemas operativos diferentes de Windows. Fue patrocinado previamente por Novell y ahora por Xamarin. Incluye las herramientas de desarrollo y la infraestructura necesarias para ejecutar los clientes .NET y las aplicaciones de servidor; entre ellas destacan: compilador C#, Mono Runtime (que implementa el CIL y dispone de compilador JIT y las bibliotecas necesarias), biblioteca de clases base compatibles con .Net Framework classes, y biblioteca de clases Mono con funcionalidades adicionales para construir aplicaciones Linux. La página oficial es: <http://www.mono-project.com/>.



1.2 Entorno de desarrollo libre NetBeans

1.2.1 Instalación

El entorno de desarrollo libre NetBeans, en su última versión, se puede descargar de la página oficial:

<https://netbeans.apache.org/download/index.html>

NetBeans proporciona también instrucciones y documentación de ayuda que podemos encontrar en:

<https://netbeans.apache.org/help/index.html>

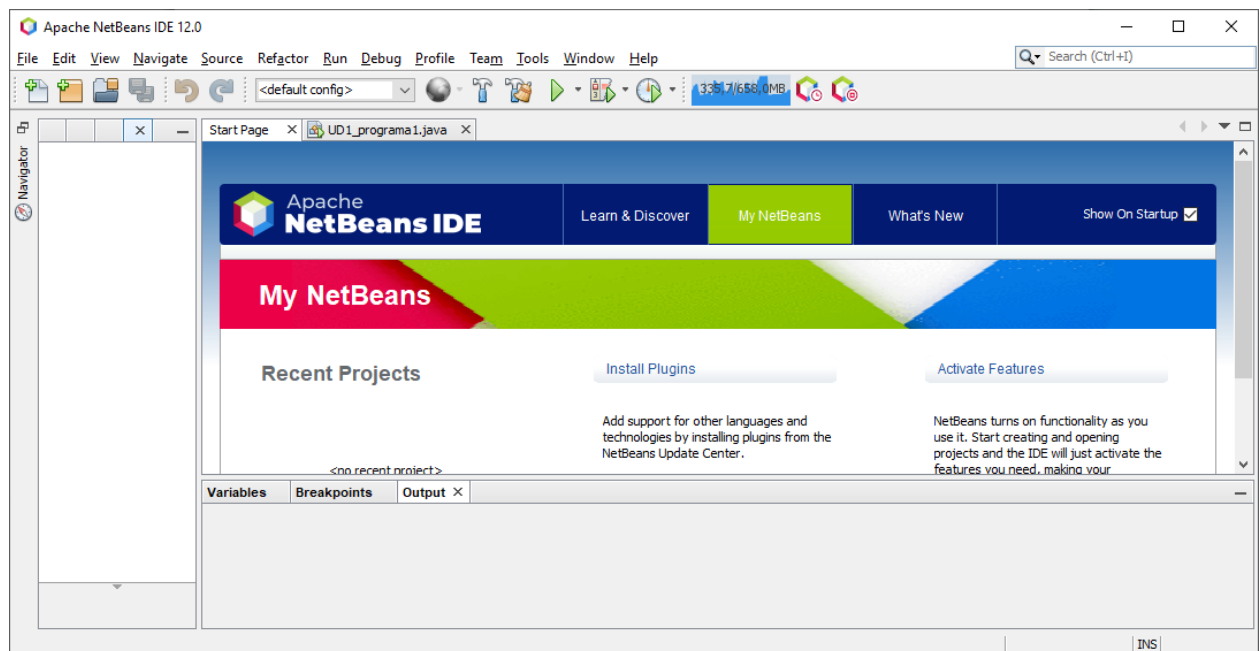
NetBeans está escrito en Java y por tanto antes de instalar NetBeans hay que tener instalado o instalar JDK (*Java Development Kit*).

JDK es un conjunto de herramientas para desarrollar aplicaciones Java que incluye entre otros JRE y el compilador Java. JRE (*Java Runtime Environment*) es el entorno necesario para ejecutar las aplicaciones Java e incluye entre otros la JVM (*Java Virtual Machine*) encargada de interpretar el bytecode Java.

1.2.1.1 Inicio

La forma más sencilla de iniciar NetBeans es hacer doble click sobre el icono del escritorio. El primer inicio puede llevar más tiempo.

El aspecto de NetBeans es muy austero la primera vez que se ejecuta después de la instalación pero a medida que se van abriendo y creando nuevos proyectos, se van incorporando nuevos elementos que van cambiando su aspecto. En general, consta de: barra de menús, caja de buscar, iconos de herramientas, y pestaña con la página de inicio.

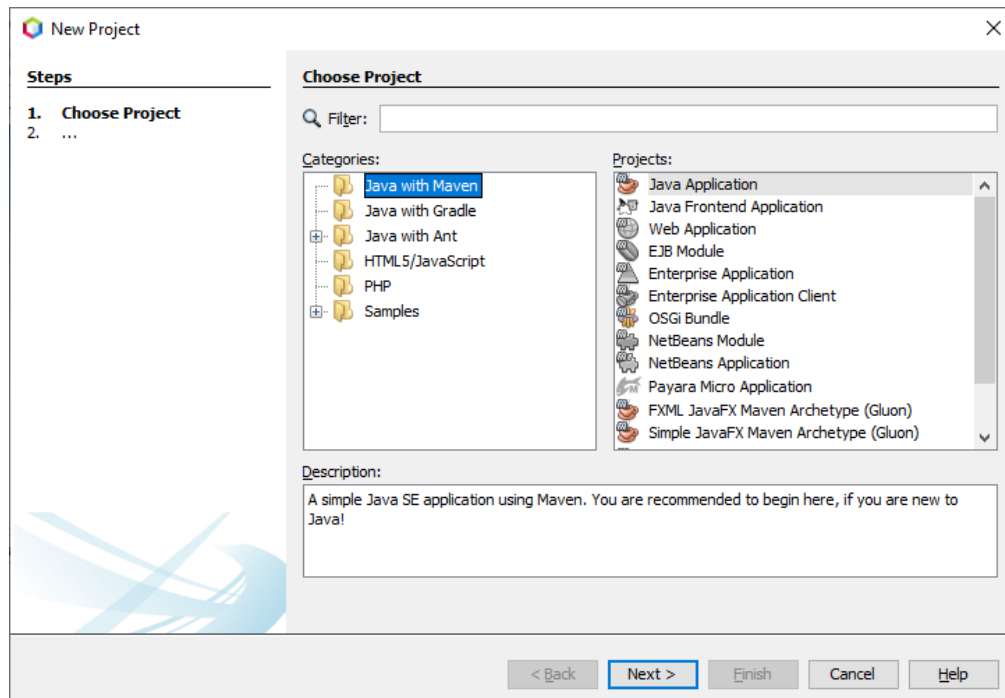


La página de inicio tiene enlaces a diferentes informaciones relacionadas con NetBeans y en el botón Mi NetBeans tiene accesos rápidos a los últimos proyectos manipulados y a la gestión de complementos. Puede obtenerse más información sobre NetBeans utilizando la opción Ayuda del menú principal o pulsando la tecla F1.

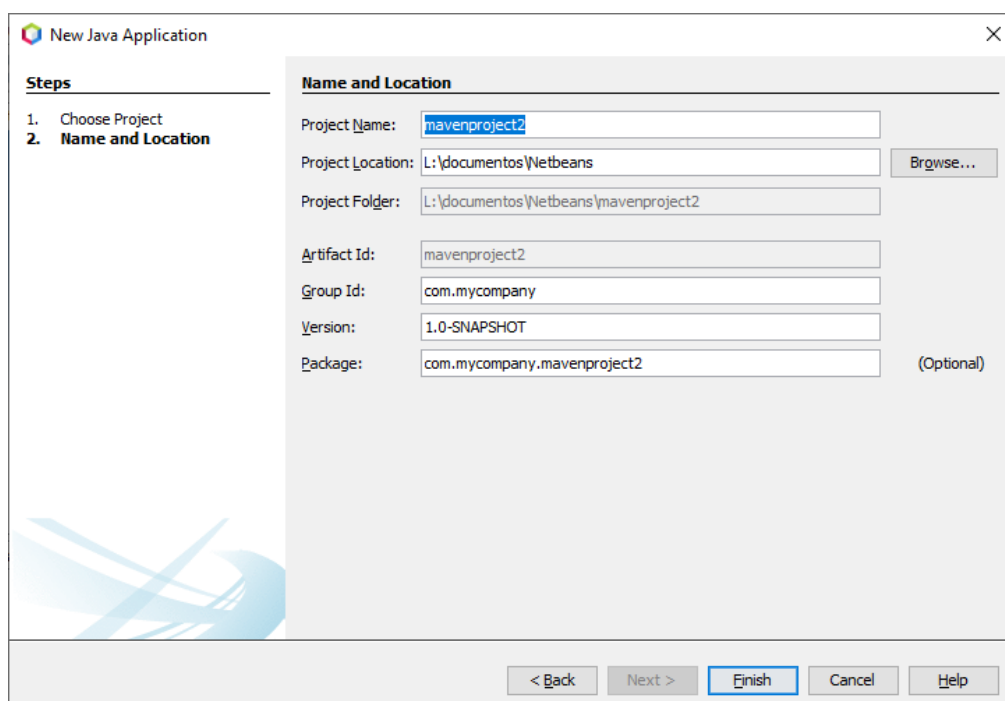
1.2.1.2 Primeros pasos

NetBeans permite editar un archivo independiente pero donde se obtiene todo el beneficio al IDE es utilizando un proyecto, es decir, una carpeta con una estructura controlada por NetBeans que puede tener ficheros de diferentes tipos relacionados que darán lugar a una aplicación.

Para crear un proyecto nuevo hay que elegir Archivo-> Proyecto nuevo, en el menú principal. En la ventana Proyecto Nuevo hay que indicar el tipo de proyecto o el ejemplo deseado.

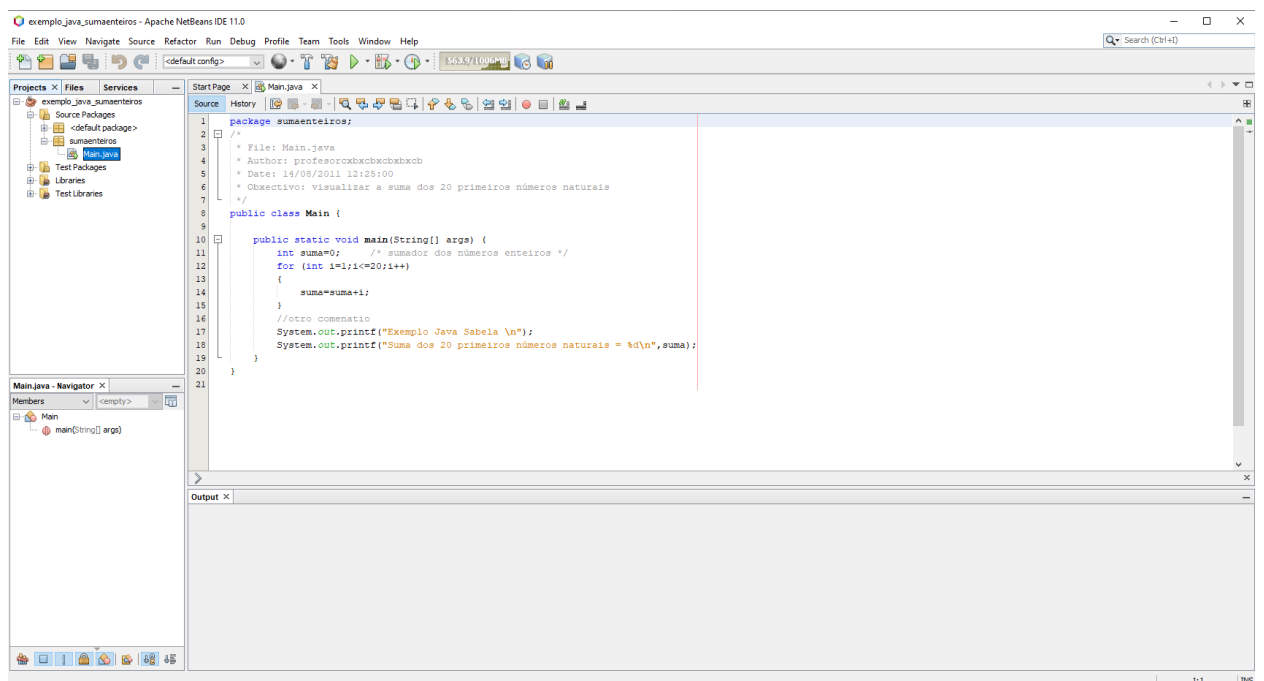


En la siguiente ventana hay que indicar por lo menos el nombre del proyecto y su localización.



El aspecto de NetBeans cambia y aparecen nuevas ventanas y pestañas dependiendo del tipo de proyecto creado. Normalmente aparecen las ventanas Proyectos, Prestaciones y Archivos que inicialmente están solapadas formando un grupo, la ventana Navegado que inicialmente está debajo y el archivo principal editado en una pestaña de la ventana de edición.

- La ventana Proyectos permite ver los proyectos abiertos y su estructura desde el punto de vista de NetBeans que no es la misma que la que se puede ver desde el sistema operativo.
- La ventana Archivos permite ver las carpetas y archivos de los proyectos abiertos tal y como se ven en el sistema operativo. La carpeta *nbproject* siempre aparece y contiene información que necesita NetBeans para gestionar el proyecto.
- La ventana Prestaciones permite enlazar con servicios.
- La ventana Navegador visualiza la estructura en forma de árbol de los elementos que forman el código editado en la ventana de edición.

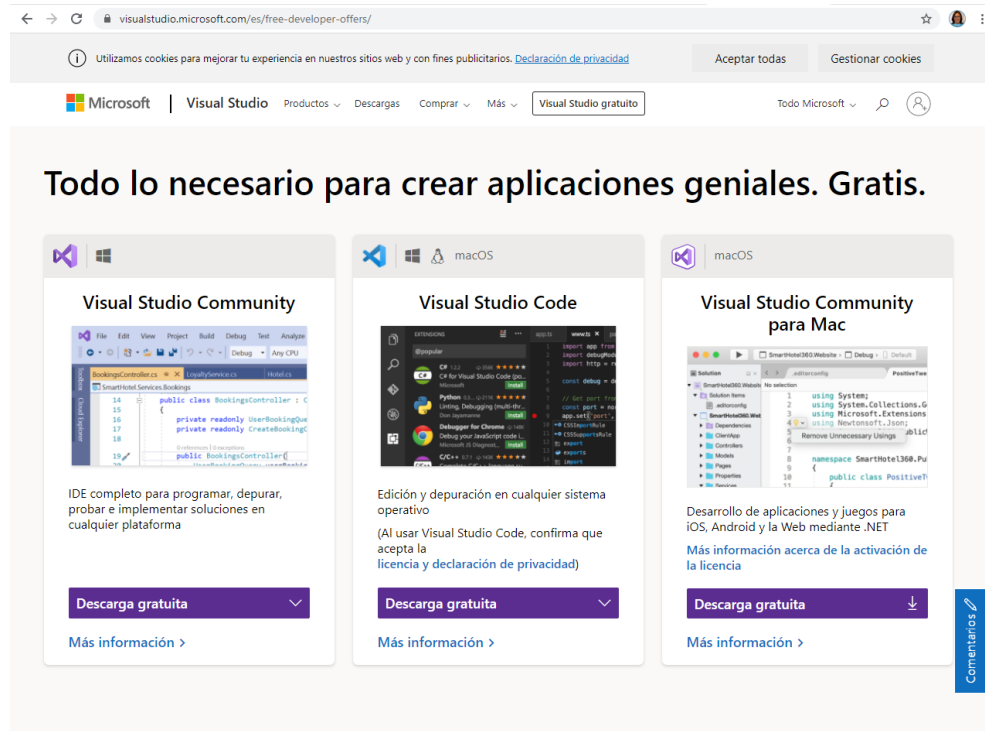


1.3 Entorno de desarrollo propietario: VisualStudio

1.3.1 Instalación

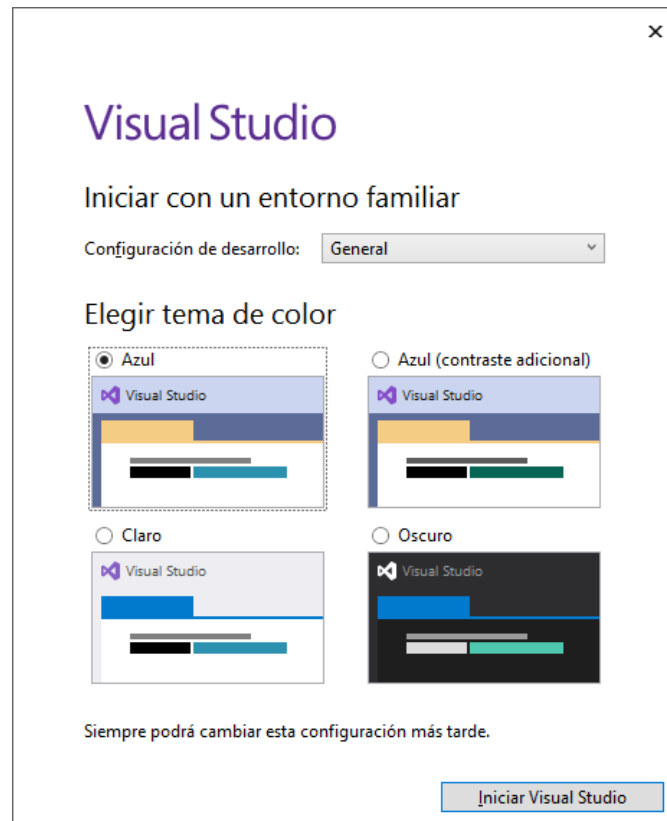
La versión de prueba gratuita de 90 días puede descargarse de:

<https://visualstudio.microsoft.com/es/free-developer-offers/>.



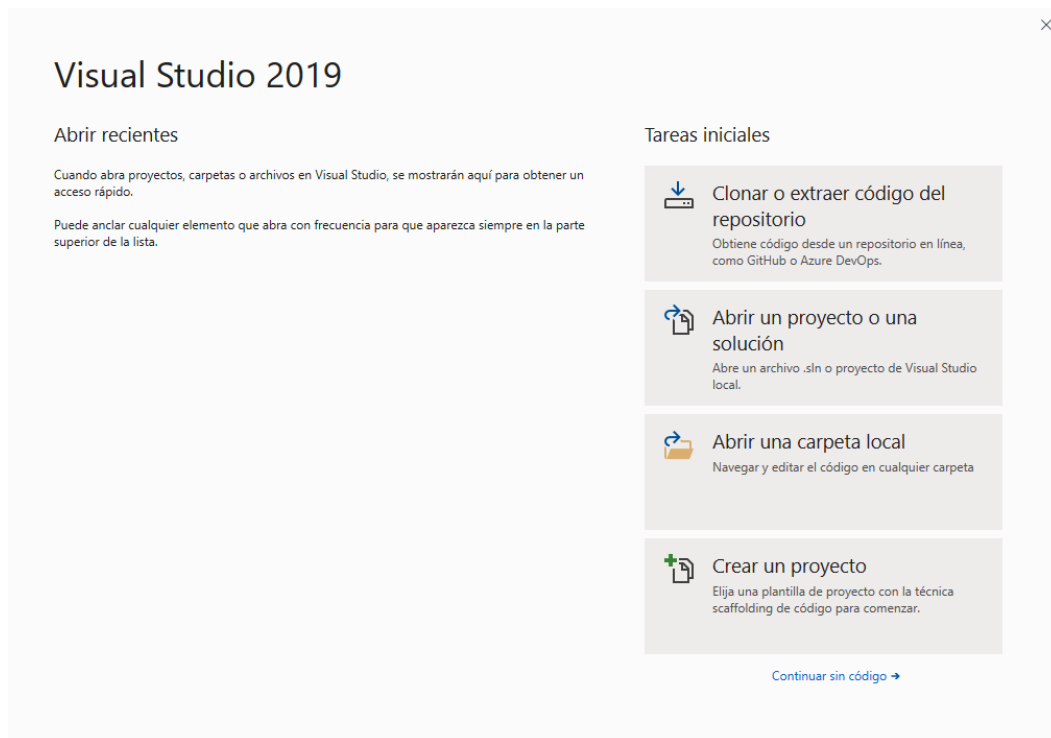
1.3.2 Inicio

La primera vez que se inicia habrá que elegir uno de los aspectos del entorno que se presentan. En general, el aspecto consta de: barra de menús, caja de buscar, iconos de herramientas, operaciones auxiliares en el borde izquierdo, pestaña con la página de inicio, y la ventana del explorador de soluciones vacío.

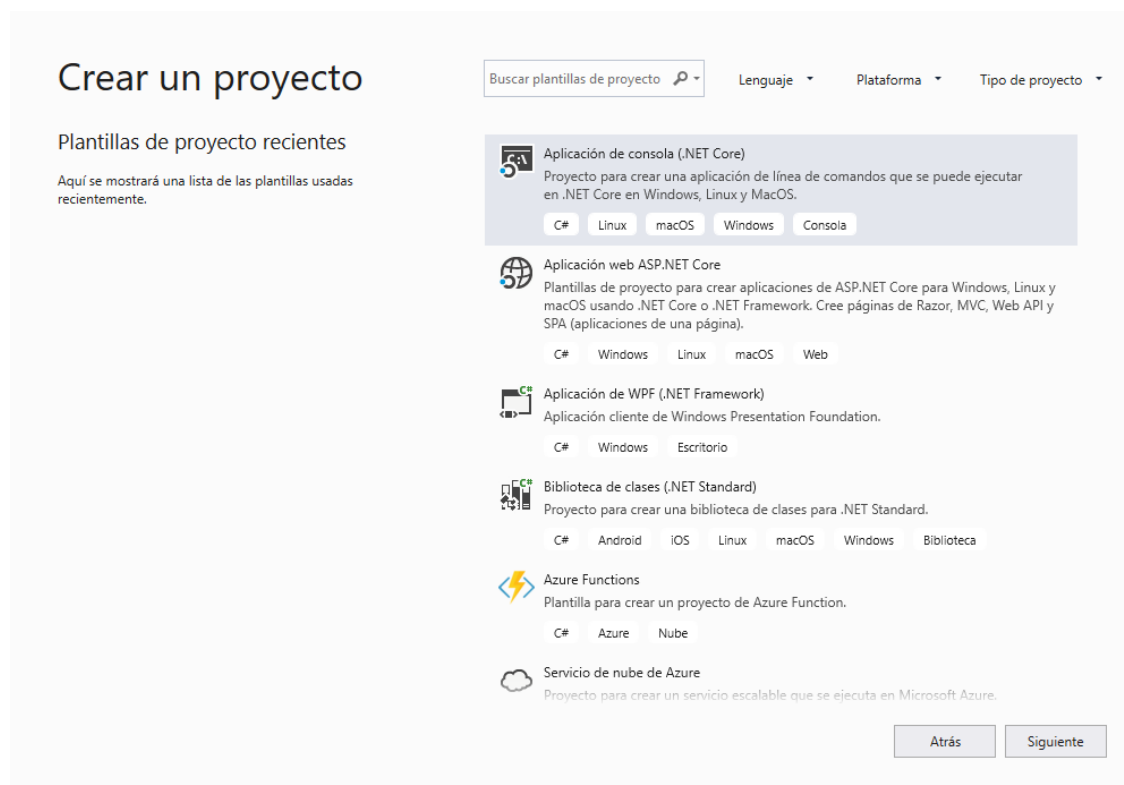


La página de inicio nos permite hacer diferentes opciones:

- Clonar repositorio
- Abrir proyecto
- Abrir una carpeta local
- Crear un proyecto.



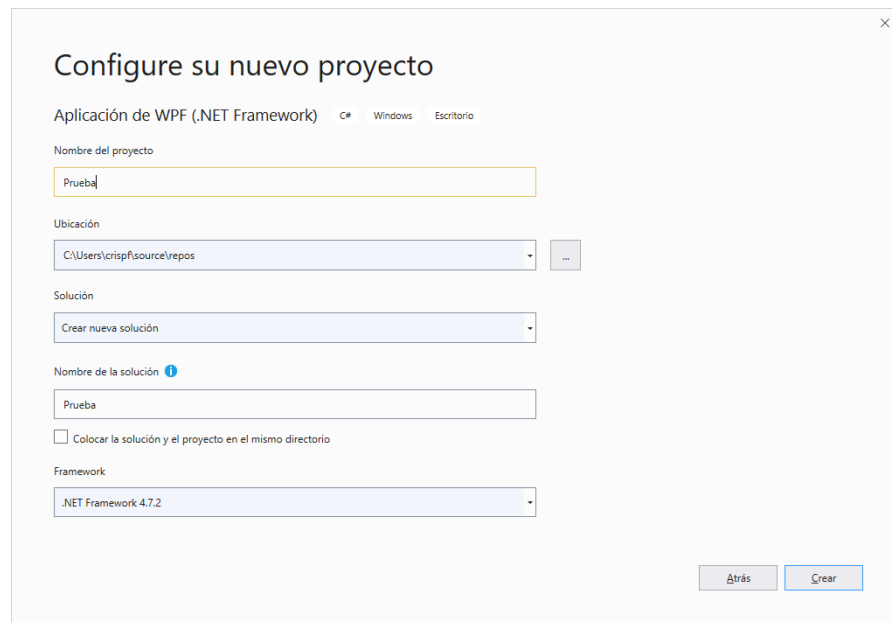
Si elegimos crear un proyecto tendremos que elegir qué clase de proyecto queremos hacer.



Se puede obtener más ayuda en el menú **HELP**.

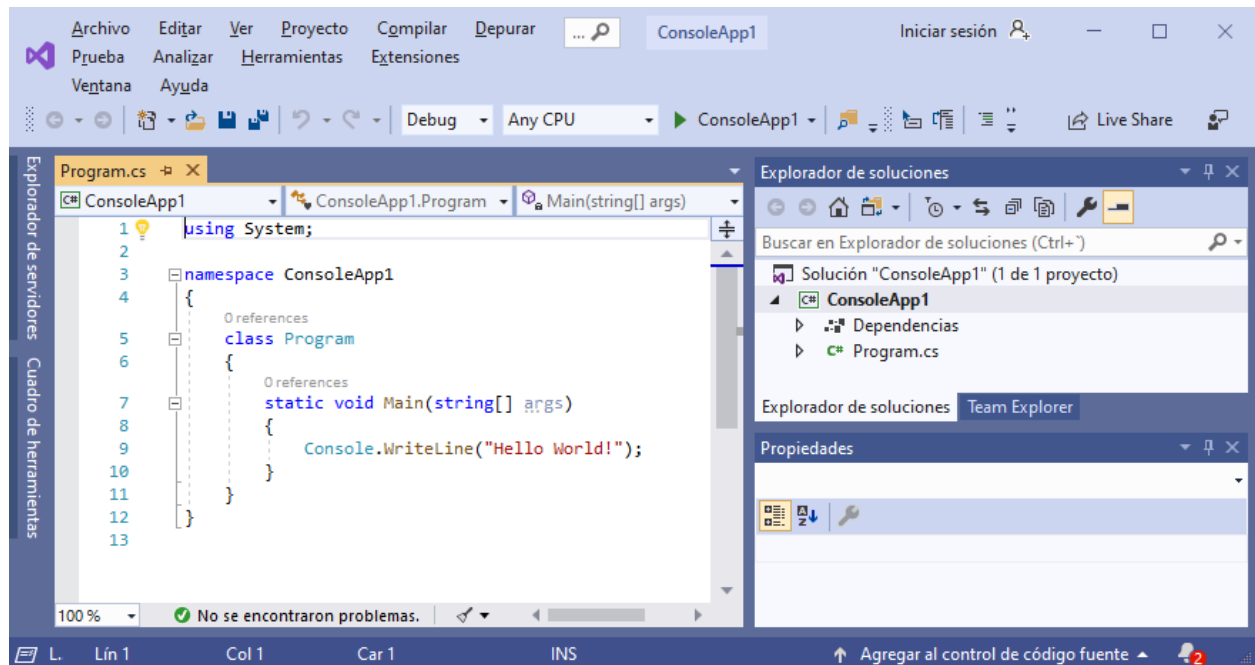
1.3.3 Primeros pasos

Un proyecto nuevo se crea desde el menú principal eligiendo *FILE>New>Project*. En sucesivas ventanas que van apareciendo habrá que elegir el tipo de proyecto a desarrollar y las características del mismo así como darle nombre y localización.



The screenshot shows the 'Configure your new project' dialog box in Visual Studio. The title bar says 'Configure su nuevo proyecto'. The main content area is titled 'Aplicación de WPF (.NET Framework)' with tabs for 'C#', 'Windows', and 'Escritorio'. The 'Nombre del proyecto' field contains 'Prueba'. The 'Ubicación' dropdown shows 'C:\Users\crispf\source\repos'. The 'Solución' dropdown shows 'Crear nueva solución'. The 'Nombre de la solución' field contains 'Prueba'. There is a checkbox 'Colocar la solución y el proyecto en el mismo directorio' which is unchecked. The 'Framework' dropdown shows '.NET Framework 4.7.2'. At the bottom right are 'Atrás' and 'Crear' buttons.

El aspecto de la pantalla inicial cambia después de crear un proyecto o una solución y aparecen elementos nuevos: el explorador de soluciones en la parte superior derecha con información sobre el proyecto o la solución, debajo aparecen las propiedades del elemento seleccionado en el explorador; el archivo principal con el código que Visual Studio tenga como modelo para este tipo de proyecto editado en una pestaña de edición e incluso puede aparecer la ventana de salida de información en la parte inferior.



El explorador de soluciones permite tener una vista organizada de los proyectos y sus archivos así como un acceso rápido a los comandos correspondientes pero la estructura que se ve en él no se corresponde con la estructura de carpetas que se ve en el sistema operativo.

Visual Studio dispone de un sistema de ayuda en línea para escribir código, adaptado al lenguaje que se esté utilizando.

El cierre de una solución se hace en *FILE>Close Solution* y la apertura de una nueva implica el cierre de la solución anterior.

ANEXO I: Materiales

I. Textos de apoyo o de referencia

- https://manuais.iessanclemente.net/index.php/Visual_Studio_Code
- <https://manuais.iessanclemente.net/index.php/Eclipse>

II. Recursos didácticos

- Apuntes en el aula virtual.
- Ordenador personal, con navegador web y conexión a internet.

- Software para elaboración de documentos de texto.