

1. Haz un repaso histórico del nacimiento y la evolución del lenguaje de programación C# y de la plataforma .NET

La plataforma .NET de Microsoft surge en el año 2000, surge para ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

C# forma parte del proyecto .NET. Fue diseñado en el año 2000 por un danés (Anders Hejlsberg) experto en Java y C++, para corregir los errores de ambos.

Para programar con la tecnología .NET, hay más lenguajes además de C#, pero éste es su lenguaje nativo. Microsoft creó la plataforma .NET en el año 2000 y en el año 2001 se mandó las especificaciones de C#, así se estandarizó y se hizo público ([ECMA-334](#)).

2. Explica en qué consiste la POO y cuales son sus pilares más importantes

Es un enfoque particular para la construcción del software que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas.

Sus pilares fundamentales son:

Los objetos: contienen propiedades o atributos, y ciertos comportamientos asociados. Son el resultado de una abstracción de la realidad.

Clase: Es la que define las propiedades y los comportamientos de los objetos.

Método: Es un algoritmo asociado a un objeto que realiza cierto comportamiento.

Evento: Es un suceso del sistema que el sistema interpreta mandándonos algún mensaje.

También puede realizar un evento un objeto.

Mensaje: Comunicación dirigida al objeto.

Propiedad o atributo: Contenedor de datos asociado a un objeto, que puede ser alterado por algún método.

Estado interno: Propiedad del objeto que solo puede ser visible y alterada por un método del objeto.

Y las características más importantes son: Abstracción, encapsulamiento, principio de ocultación, polimorfismo y herencia.

3. Haz un programa en C# que contenga al menos una clase y que muestre por pantalla un mensaje de saludo. Comenta cada parte del código.

4. Mira la ayuda de C# y haz una relación de los métodos de la clase “Console” que permiten limpiar la pantalla, leer información del teclado y mandar información para que se muestre en la pantalla

Limpiar la pantalla →

Console.Clear(): Borra la información que se muestra en el búfer de pantalla y en la correspondiente ventana de la consola.

Leer información del teclado →

Console.Read(): Lee el siguiente carácter del flujo de entrada estándar.

Console.ReadLine(): Lee la siguiente línea de caracteres del flujo de entrada estándar.

Console.ReadKey(): Obtiene la siguiente tecla de carácter o de función presionada por el usuario. (Con sobrecarga para ocultar o no la tecla pulsada).

Mandar información a la pantalla →

Console.Write(): Escribe por pantalla. Sobrecarga para cada tipo de dato básico.

Console.WriteLine(): Escribe por pantalla con un salto de línea al final. Sobrecarga para cada tipo de dato.

5. Como ya sabemos en las clases hay métodos, algunos de ellos son especiales; constructores y destructores, habla de cada uno de ellos y de su utilidad.

Los constructores son unos métodos especiales que se ejecutan automáticamente al crear un objeto de la clase. Poseen el mismo nombre que la clase a la que pertenece. Pueden existir varios constructores sobrecargados aunque no puede existir constructores virtuales. Los destructores no hacen falta llamarlos explícitamente en la ejecución del programa, son invocados automáticamente al alcanzar el flujo del programa el fin del ámbito en el que está declarado el objeto. Estos se encargan de liberar los recursos ocupados por los objetos.

6. Los componentes principales de la plataforma .NET son: CLR, CLS, BCL y JIT.

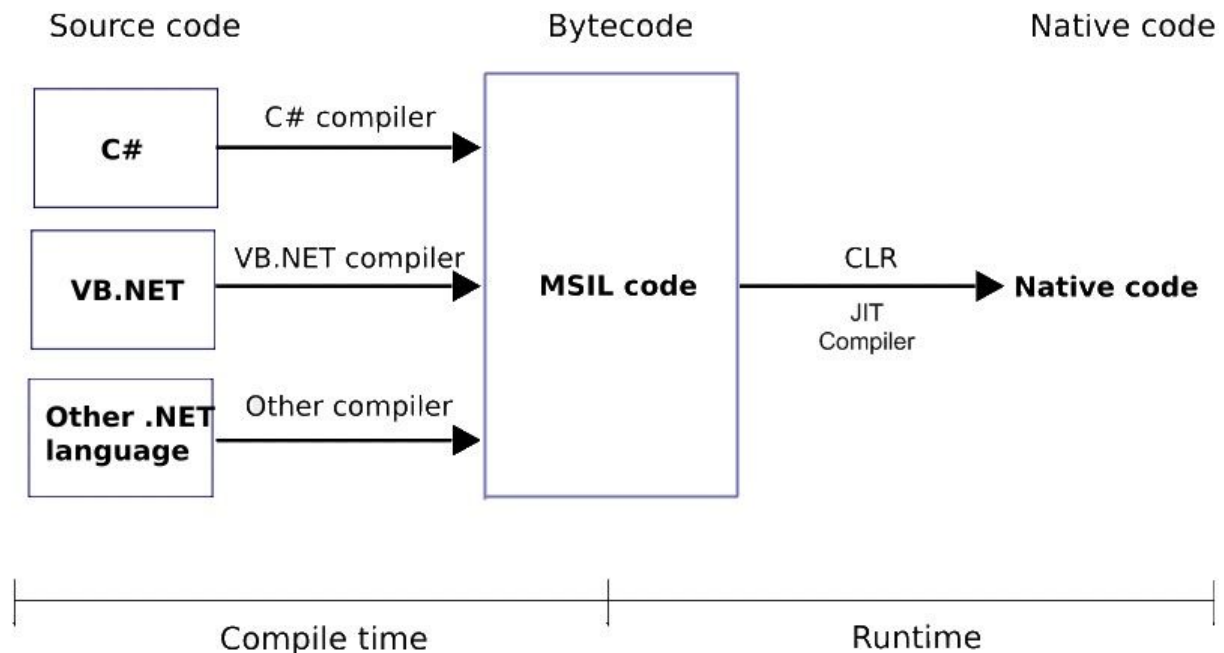
Habla de las características más importantes de cada uno de ellos.

- CLR (Common Language Runtime): es el entorno común de ejecución en el que están escritos los diferentes lenguajes de programación de .NET.
- CLS (Common Language Specification): indica las especificaciones que debe cumplir cualquier lenguaje que se pueda utilizar en la plataforma .NET.
- BCL (Base Class Library): contiene clases, que contienen métodos.
- JIT (Just In Time): Es el intérprete, encargado de transmitir cada instrucción del “Bytecode” a la máquina. Es capaz de detectar errores en el momento de la ejecución.

7. ¿Qué son y para qué sirven los namespace (espacio de nombres) en C#?

Es una palabra reservada que se usa en el código para declarar un ámbito que contiene un conjunto de objetos relacionados. Puede usar un espacio de nombre para organizar los elementos de código y crear tipos únicos globales, así se puede evitar el conflicto entre dos clases que se llamen igual declarándolas en ámbitos diferentes.

8. Explica y haz un gráfico del proceso de compilación y ejecución de un programa de consola en la plataforma .NET



Se compila el código fuente escrito en algún lenguaje que cumpla los estándares de .NET mediante un compilador de dicho lenguaje. Cuando es compilado se genera un bytecode (o MSIL code en .NET) que es el que interpreta el compilador JIT (Just In Time) que traduce al lenguaje nativo del sistema.

9. ¿En qué consiste el MSIL, que inconvenientes y ventajas tiene?

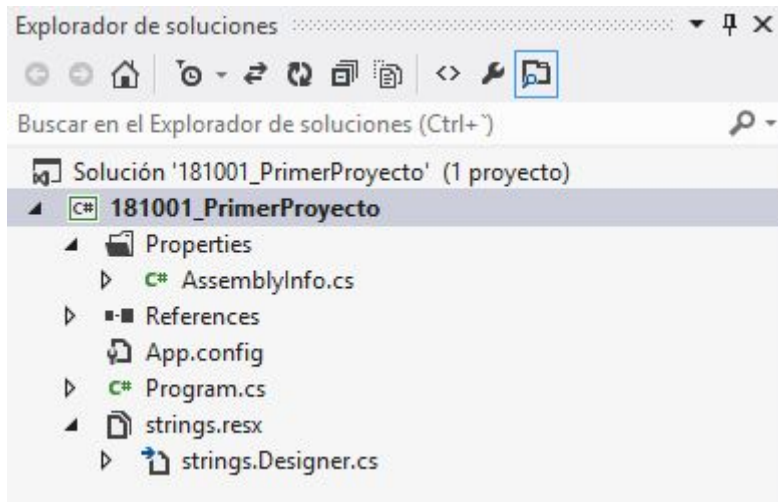
La compilación en tiempo de ejecución (JIT, just-in-time) involucra convertir el código intermedio en código inmediatamente ejecutable por la CPU. La conversión es realizada gradualmente durante la ejecución del programa.

Ventajas: La compilación JIT proporciona optimización específica para el ambiente, seguridad de tipos en tiempo de ejecución, y verificación de ensamblados. Para cumplir con esto, el compilador JIT examina los metadatos del ensamblado en busca de accesos ilegales y maneja las violaciones apropiadamente.

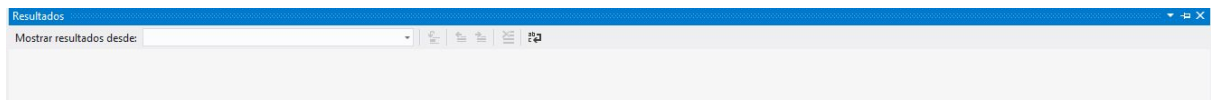
Inconvenientes: Al ser interpretado esto supone mayor consumo de recursos y por lo tanto menor velocidad de ejecución que códigos no interpretados.

10. Haz un dibujo con las partes o ventanas que aparecen por defecto en el compilador de C# "Microsoft Visual 2010".

Podemos encontrarnos con la ventana "Explorador de soluciones" donde podremos ver la estructura del proyecto, juntos con las clases y recursos que lo componen.

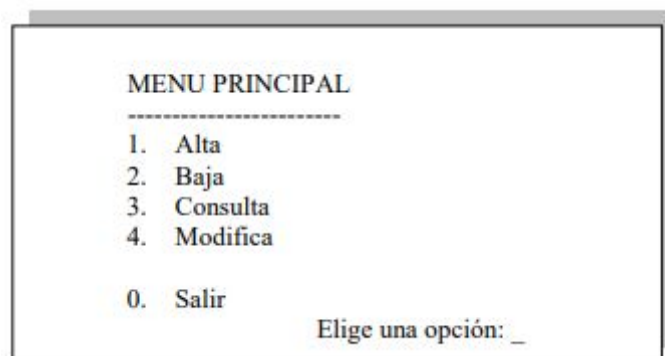


También podemos encontrar otras como Resultados: donde podremos comprobar los errores de compilación del programa y las advertencias. Además de los resultados de ejecución entre otras cosas.



11. Haz un programa de consola que pida por teclado tu nombre y muestre en pantalla: Sebastián, “Bienvenido al mundo del .NET”. Nombre será el nombre introducido por teclado.

12. Haz un programa de consola que muestre por pantalla en una posición determinada (10, 5) un menú, similar a este:



13. Cambia la versión del programa anterior para que se muestre el menu en otros colores

14. Haz un programa para implementar el ejercicio 1 de la relación 2 “Diseñar un algoritmo que calcule el área de un triángulo”, de base 8 y altura 5,5.

15. Haz un programa que calcule el área de una circunferencia. El radio se pedirá desde el teclado. Se mostrará por pantalla una frase indicando el valor del área.

16. Haz un programa en C# que calcule el área lateral y el volumen de un cilindro recto, introduciendo por teclado los valores del radio y de la altura.