

Las ventajas que ofrece C# frente a otros lenguajes de programación son:

Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.

Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.

Atributos: cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido y privado.

Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.

Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override

Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.

Inicializador: un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.

Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

DESVENTAJAS

Las desventajas que se derivan del uso de este lenguaje de programación son que en primer lugar se tiene que conseguir una versión reciente de Visual Studio .NET, por otra parte, se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, etc.

Además, para quien no está familiarizado con ningún lenguaje de programación, le costará más trabajo iniciarse en su uso, y si se quiere consultar algún tutorial más explícito sobre la programación en C# se tendría que contar además con una conexión a Internet.

¿Qué es un framework?

Es un conjunto de herramientas, librerías y buenas prácticas para **resolver una serie de problemas**. En Desarrollo Backend, todos los frameworks tienen la principal tarea de **crear un servidor web**.

¿Por qué usar un framework o una librería?

Primero, con estas herramientas puedes tener un **desarrollo más rápido** usando buenas prácticas y evitando escribir código repetitivo. Pero lo más importante, **enfocar tu tiempo en solucionar problemas propios de tu desarrollo**, en lugar de solucionar problemas que otros ya han solucionado.

¿Cómo escoger el framework o librería correcta para mi proyecto?

Aquí te dejo 7 puntos clave para el momento de escoger un framework o una librería:

- Disponibilidad y calidad de la documentación
- Tamaño de la comunidad
- Issues abiertos en GitHub
- Problemas que puede solucionar
- Flexibilidad
- Complejidad
- Compatibilidad con las otras herramientas que usarás

Python

Una de las principales ventajas de este framework de desarrollo backend es que es una lenguaje con una rápida curva de aprendizaje. Se suele combinar con Django a modo de framework para sacarle el máximo partido posible.

PHP

Los más utilizados para este lenguaje de programación son Laravel y Symfony. Ambos son simples, potentes y tienen una interfaz elegante y divertida de usar.

