

# .NET y J2EE

## VALORACIÓN Y COMPARACIÓN DE LOS ELEMENTOS DE LAS DOS PLATAFORMAS

Definiciones .....	2
C# y Java.....	3
Similitudes .....	4
Ventajas.....	4

## **Definiciones**

### **Sobre J2EE**

J2EE (Java 2 Platform Enterprise Edition) es, según Sun Microsystems, un conjunto de especificaciones que permiten el desarrollo de aplicaciones basadas en la tecnología Java.

La plataforma J2EE ha sido construida con la participación de varios cientos de empresas de forma conjunta y no exclusiva, por este motivo la plataforma no es propiedad de ninguna de las compañías participantes.

El lenguaje base es Java. Este lenguaje se caracteriza por ser portable, robusto, estable, independiente de la plataforma y orientado a objetos. Puede ser considerado el heredero de C++, siendo su filosofía una mezcla entre este lenguaje y el lenguaje pionero de la orientación a objetos Smalltalk. Al igual que C++, Java es fuertemente tipado por lo que se realiza comprobación de tipos en compilación y no en ejecución como ocurre con Smalltalk. Sin embargo, en Java todas las clases derivan de una clase base (llamada Object) como ocurría en Smalltalk pero no en C++. Otra característica que Java comparte con C++ es la herencia múltiple, una clase puede derivarse de varias clases, utilizando un mecanismo al de C++, el concepto de interfaces. Una diferencia fundamental entre C++ y Java es que en Java no se pueden definir funciones o procedimientos libres, en todo momento se tiene que trabajar con clases y objetos. Los tipos de datos primitivos de Java que no son clases se asocian a clases con sus correspondientes métodos. Tal vez, el concepto principal que ha hecho popular a Java es su portabilidad a través de múltiples plataformas. La solución original consiste en incorporar el conocido concepto de máquina virtual a los lenguajes de programación. Cuando compilamos código Java no se traduce a lenguaje máquina de la plataforma en la que se va a ejecutar, si no que se traduce a código binario propio que se puede interpretar y ejecutar en la máquina virtual Java (VM). Así el código Java compilado en una plataforma Windows o en una plataforma HP-UX genera el mismo código binario que se puede ejecutar sobre las diferentes máquinas virtuales Java de cada plataforma.

A partir de la propuesta Java 2, el entorno Java se ha dividido en dos, un entorno de ejecución o run-time que permite la instalación de todo lo necesario para la carga de la máquina virtual Java y un kit de desarrollo o SDK que incorpora todo lo necesario para poder escribir programas en Java.

### **Sobre .NET**

.NET es, según Microsoft, una plataforma de desarrollo de servidores, clientes y servicios.

La plataforma .NET es una implementación de Microsoft basada en estándares abiertos como CLI (Command Line Interface), SOAP (Simple Object Access Protocol) y WSDL (Web Services Definition Language) que permite al programador disponer de un entorno único para trabajar en múltiples lenguajes (más de 30) . La innovación de esta plataforma es la utilización de un lenguaje intermedio MSIL (Microsoft Intermediate Language) que incluye un sistema de tipos común (CTS) y un lenguaje común de especificación (CLS); y que permite que los programas escritos en diferentes lenguajes de programación puedan ser traducidos a un lenguaje único (el MSIL) en el que todos se entienden. Así es posible que un programa escrito en C++ se entienda o haga uso de

elementos de una clase escrita en Java o que un determinado componente de comunicaciones escrito en ADA sea utilizado en una clase C#.

Los lenguajes base del entorno .NET son C# y VisualBasic.

El lenguaje C# es un lenguaje que se ha definido sobre un estándar abierto y que ha sido patrocinado en su construcción por Microsoft. Este lenguaje se caracteriza por ser fuertemente tipado, simple, orientado a objetos y a componentes, y aportar mecanismos que permitan que el código sea consistente y durable.

El lenguaje VisualBasic ha sido modificado de manera importante y en la estructura actual puede considerarse como un lenguaje orientado a objetos en su totalidad (ha perdido los vestigios que le quedaban del antiguo Basic).

La propuesta de Microsoft respecto al entorno .NET suministra tanto el entorno de compilación (para tres lenguajes C#, VisualBasic, y Jscript), que permite pasar código de estos tres lenguajes a código intermedio, como un entorno de ejecución (denominado paquete redistribuible) que permite la ejecución del código intermedio generado para .NET.

Uno de los elementos fundamentales de .NET es su orientación a permitir el desarrollo de servicios (y especialmente de servicios Web). Un servicio web es un software que se suministra a aplicaciones basadas en componentes y que resuelve una determinada necesidad. La orientación tradicional de aplicación cerrada o de aplicaciones con librerías se modifican y se debe tener en cuenta que las aplicaciones deben ser abiertas para poder en un determinado momento tomar servicios nuevos suministrados por un servicio Web y que permiten no tener que realizar modificaciones completas o parciales sobre toda una plataforma en explotación.

## **C# y JAVA**

C# ha sido creado como una evolución de Java y C++ ( de hecho en algunos momentos su nombre fue Java++) por lo que existen coincidencias en elementos fundamentales de los lenguajes como son:

1. No existe compilación a código máquina nativo. Los dos lenguajes se compilan a código “intermedio” que es independiente de la plataforma y del sistema operativo en el que se van a ejecutar. Por lo tanto podemos desarrollar software en un entorno, generar el código intermedio portable y que funcione en la plataforma destino diferente a la inicial.
2. No usan punteros. En el caso de Java ha sido uno de las características fundamentales desde su inicio, y en el caso de C# se permite su uso sólo en un modo de programación restringido (unsafe).
3. No se utilizan ficheros de declaración (como los ficheros de cabecera de C o C++), todo el código se debe organizar en unidades que almacenan clases, en el caso de JAVA conocidas como packages y en el caso de C# conocidas como assemblies.
4. Admiten herencia múltiple utilizando interfaces.
5. Admiten clases internas
6. Todo es una clase.

## **Similitudes entre .NET y J2EE**

- **Multiplataforma.** Al separar los pasos de compilación a código intermedio de la ejecución de este código intermedio, las dos orientaciones tienen la posibilidad de crear entornos de ejecución propios de las diferentes arquitecturas. Las dos plataformas han tomado la orientación de suministrar un entorno de ejecución (run-time) y un entorno de desarrollo (kit de desarrollo).
- **Paquetes similares para el desarrollo de aplicaciones.** Las clases desarrolladas en las dos plataformas son muy similares, por ejemplo, JSP (Java Server Pages) y .NET ASP (Active Server Pages), coinciden en su arquitectura de clases.
- **Servidores de aplicaciones.** Los dos entornos proporcionan un modelo de acceso a los componentes de las aplicaciones para manejo de datos y lógica del negocio, separado por un nivel intermedio, ASP en el caso de .NET y Servlets en el caso de J2EE.
- **Servicios Web.** Los dos entornos tienen como propósito primordial disponer de la capacidad de crear y utilizar servicios web.
- **Lenguajes orientados a objetos.** Las dos plataformas tienen en su base el uso de lenguajes orientados completamente a objetos, JAVA en el caso de J2EE y C# en el caso de .NET.

## **Ventajas**

### **De .NET frente a J2EE**

- **Múltiples lenguajes de programación.** El hecho del multilenguaje ha sido una constante en la evolución de la informática y pese a la definición de nuevos lenguajes más completos y de más alto nivel, todavía al día de hoy existe una gran cantidad de código funcionando en lenguajes como COBOL, BASIC, C o FORTRAN. La aparición de .NET puede permitir que el desarrollo de compiladores de estos lenguajes a código intermedio MSIL.
- **C# supone una evolución en los lenguajes de programación orientados a objetos,** incorpora capacidades que los programadores de lenguajes como JAVA o C++ podían echar de menos y simplifica alguno de los mecanismos más complejos de estos lenguajes.
- **El entorno .NET está especialmente orientado a la creación de servicios Web.** La existencia de una arquitectura clara y sencilla de clases para crear y distribuir estos servicios, incluyendo formatos como SOAP (Simple Object Access Protocol), WSDL (Web Services Description Languages) y UDDI (Universal Description, Discovery, and Integration).

### **De J2EE frente a .NET**

- Las implementaciones de J2EE pueden adquirirse a varias compañías(en lugar de a una única como es .NET).
- JAVA tiene un largo camino andado en relación al desarrollo de su arquitectura sobre diferentes plataformas.
- La tecnología JAVA es abierta y se base en gran medida en estándares de organizaciones de normalización y “de facto”.