



UNIVERSIDAD DE OVIEDO

**ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA
EN INFORMÁTICA DE OVIEDO**

MÁSTER OFICIAL EN INGENIERÍA WEB

Comparativa plataformas Java - .Net

Programación orientada a Objetos

Asignatura: Programación Orientada a Objetos

Alumno: Francisco Adolfo Cifuentes Silva

Email: Uo214916@uniovi.es

Fecha: 3 de Noviembre de 2009

Índice

Índice	2
Resumen	3
Reseña histórica.....	4
Plataforma .NET	4
Plataforma Java	4
Características generales	5
Ventajas y desventajas de cada plataforma	9
Referencias	10

Resumen

A continuación se presenta una comparación sobre aspectos generales de las plataformas Java y .NET, se realiza un paralelo entre las características consideradas fundamentales. La idea principal del informe es mostrar las potencialidades básicas de cada una de estas plataformas, visualizando ventajas y desventajas desde una perspectiva exclusivamente técnica sin generar conclusiones que puedan evidenciar preferencias de algún tipo.

Reseña histórica

Plataforma .NET

La plataforma .NET nace el año 2000 como un proyecto de Microsoft Corp. Con la idea de crear una plataforma de desarrollo de software que permitiera unificar todos los productos Microsoft, desde las herramientas de desarrollo hasta el sistema operativo. La idea central de .NET en términos tecnológicos es generar software independiente de la plataforma de hardware y del sistema operativo, poniendo especial énfasis en el desarrollo de aplicaciones distribuidas sobre redes y la interoperabilidad de lenguajes, cada uno asociado a diferentes dominios de resolución de problemas.

La primera versión estable de .NET (1.0 RTM) fue lanzada el 5 de enero de 2002 y la última versión estable es la 3.5.30729.1 del 11 de Agosto de 2008.

Plataforma Java

La plataforma Java nace en el año 1991 como un proyecto de Sun Microsystem bajo la dirección de James Gosling llamado The Green Project, orientado a ser un lenguaje para ser utilizado en primera instancia en sintonizadores de televisión (set-top-box). El nombre del lenguaje cambió desde Oak a Green y finalmente a Java, esto dado que los nombres anteriores ya eran marcas registradas.

El objetivo principal de la plataforma era lograr la implementación de un entorno de ejecución para diferentes sistemas operativos en donde podría correr un mismo binario, agregando que los programas fueran escritos con una sintaxis similar a la de C++.

El primer lanzamiento público del kit de desarrollo de Java (Java Development Kit 1.0) fue el 23 de Enero de 1996, seguido de 8 versiones hasta la fecha, siendo la más actual la versión 1.6.0_16-b01 liberada el 31 de agosto de 2009.

Características generales

Característica	Plataforma .NET	Plataforma Java
Compañía propietaria	Microsoft Corp.	Sun Microsystems
Versión estable	3.5.30729.1 (11-Agosto-2008)	1.6.0_16-b01 (31 de agosto de 2009)
Lenguajes de programación compatibles	C#, Visual Basic .NET, ASP.NET, Turbo Delphi for .NET, C++, C, Java, J#, Perl, Pitón, Fortran, Cobol.NET, ADA, APL, Haskell, Javascript, Lisp, Prolog, Smalltalk y muchos otros, que deben cumplir con las reglas de CLS (Common Language Specification)	Java
Paradigma de programación	Orientado a objetos, Declarativo, funcional	Orientado a objetos de forma no estricta, no permite herencia múltiple ni sobrecarga de operadores
Premisas de la tecnología	<p>Disponer de la tecnología apropiada para un problema específico.</p> <p>Distribuir aplicaciones en forma de servicios.</p> <p>Comunicar y combinar aplicaciones independiente del modelo de programación, plataforma y lenguaje.</p> <p>Integrar todos los productos Microsoft, desde herramientas de desarrollo al sistema operativo.</p>	<p>“Write Once, Run Anywhere”</p> <p>Escríbelo una vez, ejecútalo donde sea.</p> <p>Permitir su la ejecución de un mismo programa compilado en múltiples sistemas operativos.</p> <p>Utilizar POO</p> <p>Incluir soporte para trabajo en red.</p> <p>Trabajar de forma segura en sistemas remotos.</p> <p>Desarrollar aplicaciones distribuidas.</p>
Entorno de ejecución	.NET Framework (Entorno común de ejecución para	Java Runtime Environment - JRE (Máquina virtual de Java – <i>Java</i>

	lenguajes - <i>Common Language Runtime</i> CLR, Base Class Library BCL, Common Language Specification CLS)	<i>Virtual Machine</i> JVM y Biblioteca de clases API)
Implementaciones de la máquina abstracta	CLR, MONO, DotGNU Portable .NET, SSCLI	JVM, KVM
Formato del código compilado	Compilador JIT del CLR ejecuta código IL (MSIL) (bytecode)	Compilador JIT de la JVM ejecuta .class, código intermedio (bytecode)
Compilador JIT (Just In Time)	Sí	Sí
Kits de desarrollo	Distribuidos en múltiples lenguajes de programación, permite crear cualquier tipo de aplicación.	J2SE (aplicaciones de escritorio), J2EE (desarrollo empresarial), J2ME (dispositivos con recursos limitados)
Entornos de desarrollo	Visual Studio .NET	Eclipse, Netbeans, Visual Studio .NET y muchos otros
Licenciamiento	Comercial y Educativo	GNU
Implementaciones más relevantes en el API	<ul style="list-style-type: none"> • Interacción con dispositivos periféricos • Manejo de datos • Gestión de memoria • Transmisión por XML y TCP/IP • Manejador de excepciones • ASP.NET • Sistema de ventanas • Despliegue gráfico GDI+ • Seguridad e interacción con otras aplicaciones • Cadenas de caracteres y expresiones regulares • Colecciones • Windows API, API Win32 • Reflexión • Bibliotecas Específicas: <ul style="list-style-type: none"> ▪ System ▪ System.Data ▪ System.WinForms 	<ul style="list-style-type: none"> • JDBC (Conexión con bases de datos) • RMI (Invocación remota de métodos) • Reflexión • Asertos • Expresiones Regulares • Manejo de Excepciones • Logging • Parser XML • Plantillas • Colecciones • Metadatos • Enumeraciones • Autoboxing / unboxing • Servicios Web • JavaBeans • Sockets • Transacciones • Servicios de nombre • GUI: AWT, SWT, • Autenticación • Persistencia de Objetos • Gestión de memoria

	<ul style="list-style-type: none"> ▪ System.Drawing ▪ System.XML ▪ System.Web ▪ System.Collections ▪ System.IO ▪ System.Reflection ▪ Etc. 	
Aceso a características de bajo nivel	Directamente mediante código no administrado	Utilizando JNI (Java Native Interface)
Posee un estándar de desarrollo para componentes	No, solo ofrece recomendaciones	Posee un estándar de desarrollo para componentes de acceso a datos y reglas de negocio.
Tipos de estructuras que es posible crear	Clases, interfaces, struct y enum	Clases, interfaces, enum
Niveles de protección	Public, private, internal, protected.	Public, private, protected y package
Elementos que se pueden definir dentro de una clase	Atributos, métodos, clases internas, propiedades, eventos y delegates	Atributos, métodos y clases internas
Herencia	No hay herencia múltiple de implementación, solo a través de interfaces.	No hay herencia múltiple de implementación, solo a través de interfaces.
Polimorfismo	Es posible redefinir métodos, excepto cuando se usa la palabra sealed.	Es posible redefinir métodos, excepto cuando se usa la palabra final.
Sobrecarga	Se permite sobrecargar operadores.	No se permite la sobrecarga de operadores.
Encapsulamiento de atributos	Es posible acceder a los atributos a través de propiedades definidas mediante los métodos set y get.	Se pueden implementar los métodos set y get para el acceso y escritura de los atributos.
Paquetes y espacios de nombre	Se utilizan los espacios de nombre <i>namespaces</i> .	Se utilizan paquetes <i>packages</i> .
Paso de parámetros a un método	Se pueden pasar parámetros por valor, referencia y número de parámetros variable.	Se pueden pasar parámetros solo por valor

Ventajas y desventajas de cada plataforma

	Ventajas	Desventajas
.NET	<p>Entorno de desarrollo altamente productivo</p> <p>Posibilidad de utilización de una amplia gama de lenguajes de programación</p> <p>C# es un lenguaje muy completo y con excelentes características</p>	<p>Entorno de desarrollo no disponible en otras plataformas</p> <p>Implementaciones del .NET framework aun no son totalmente suficientes</p>
Java	<p>Tecnología altamente madura</p> <p>Muchos entornos de desarrollo alternativos</p> <p>Existe una implementación de la JRE para casi cualquier plataforma</p>	<p>Permite solo la utilización de Java como lenguaje de programación</p> <p>No ofrece un entorno de desarrollo altamente productivo</p>

Referencias

1. The Java Language Specification, Second Edition, Addison-Wesley, ISBN 0-201-31008-2
2. MSDN Library, Visual Studio 2008
3. Presentaciones sobre plataformas POO, Máster en Ingeniería Web
4. http://es.wikipedia.org/wiki/Microsoft_.NET
5. http://es.wikipedia.org/wiki/Plataforma_Java
6. http://es.wikipedia.org/wiki/Java_%28Sun%29
7. <http://www.sun.com/software/opensource/java/>