

# J2EE vs. .NET

Por Oscar González Moreno ([oscar.gonzalez@samelan.com](mailto:oscar.gonzalez@samelan.com)).

URL: <http://www.samelan.com/oscargonzalez>

Publicado en

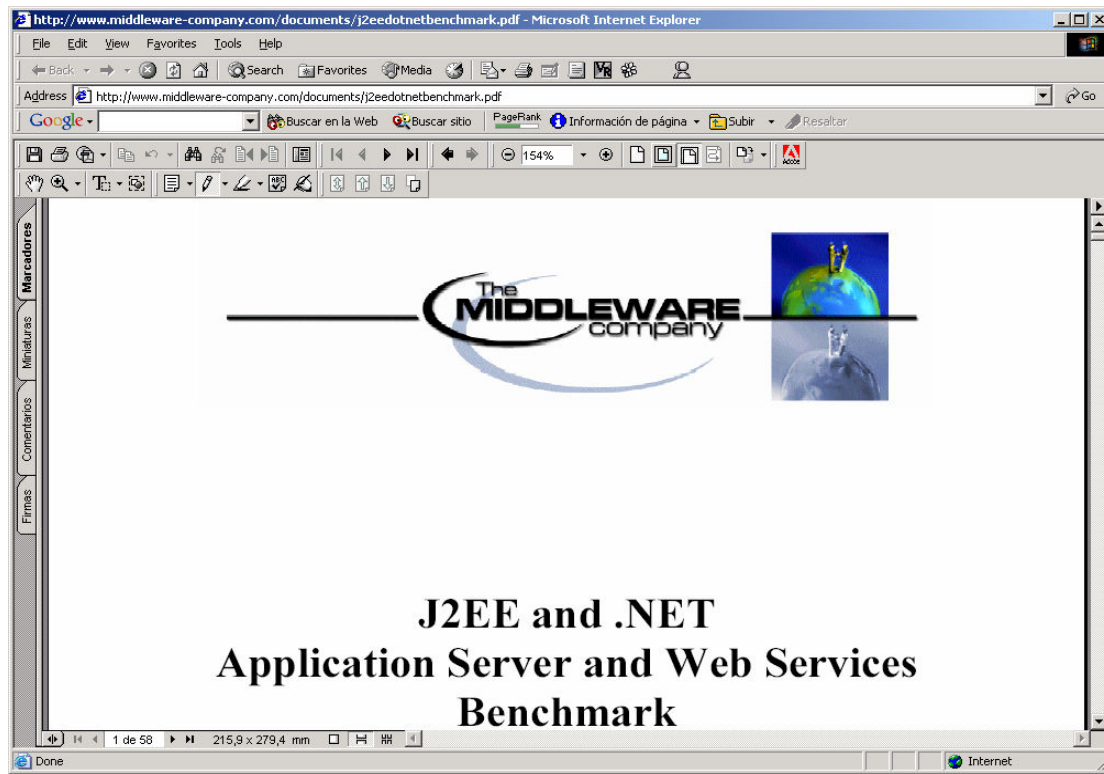
**Dr. Dobb's**  
E S P A Ñ A

El debate mas acalorado en el mundo de desarrollo de aplicaciones Web actual es el que enfrenta a J2EE y a .NET. Ambas tecnologías, de empresas rivales hasta la muerte (SUN Microsystems y Microsoft) pugnan por un mercado cada vez mas exclusivo, en el que los competidores van quedando cada vez mas rezagados.

Y es que hablamos de tecnologías de desarrollo Web que ofrezcan un modelo de desarrollo de componentes empresariales. Muchas son las alternativas de desarrollo para una aplicación Web: PHP, páginas ASP, ColdFusion, etc. y la mayoría de ellas están actualmente en un estado tecnológico excelente, que las permiten acometer la mayoría de los proyectos Web actuales (sea cual sea su tamaño). Hace poco tiempo las cabezas pensantes de Yahoo publicaban un estudio en el que se decantaban por el lenguaje PHP para el desarrollo de su sistema Web. Sin embargo, otros sistemas de alto rendimiento, como la versión digital de “El Mundo” sí que utilizan este lenguaje. Y podemos encontrar tantos ejemplos, tanto positivos como negativos, para el resto de las tecnologías.

Pero hay algo que las diferencia y es su orientación real hacia un modelo de desarrollo de componentes empresarial. Esto quiere decir que nuestra plataforma tecnológica tiene que servir y propiciar, desde el entorno de ejecución hasta el entorno de desarrollo, un modelo organizado de componentes que puedan actuar dentro de unos servicios básicos del sistema. Estos servicios pueden ser un sistema de autenticación y seguridad, un entorno transaccional, un soporte para granjas de servidores, instanciación remota de componentes, caché de objetos, Framework de acceso a datos, soporte a nuevas tecnologías como WebServices, XML, etc.

En este aspecto, la oferta de SUN siempre ha estado dividida en dos grandes grupos. Por un lado, la tecnología de Servlets y páginas JSP, que eran la réplica a las páginas ASP de Microsoft, y que permite abordar una gran gama de desarrollos, desde Intranets hasta portales verticales, con unas ciertas garantías de éxito. Pero por otra parte, SUN ofrece J2EE (Java2 Enterprise Edition), que ofrece al desarrollador el entorno empresarial antes citado. J2EE se trata en realidad de un estándar que otras empresas se encargan de implantar. Así, cuando hablamos de “servidores de aplicaciones J2EE”, no nos referimos a un producto exclusivo y realizado por SUN, sino a un servidor que cumple el estándar J2EE y que puede estar desarrollado por otras empresas, como BEA, Netscape o Allaire por citar unos cuantos.



*Imagen 1: La comparativa de rendimiento realizada por TMC*

Este estándar ya lleva muchos años en el mercado, y está a punto de alcanzar la versión 1.4, que ya se encuentra en fase Beta, y son muchas las aplicaciones que se han realizado bajo su estándar, entre ellas portales verticales, y muchos desarrollos de Banca Virtual.

Al mismo tiempo, Microsoft ofrecía su plataforma Windows DNA basada en una mezcla de tecnologías. Con el servidor Web Internet Information Server 4.0 como nexo común y el servidor de base de datos SQL Server, esta plataforma utilizaba la tecnología de objetos COM, el servidor transaccional MTS y MSMQ y las páginas ASP para el desarrollo de aplicaciones Web empresariales. Pero este modelo ofrecía algunas desventajas. Por un lado, la comunicación y utilización entre componentes se realizaba a través de una interfaz binaria definida por el estándar COM. Así, el desarrollador podía elegir entre diversos lenguajes de desarrollo como Visual Basic, Visual C++, o diversas librerías como ATL o MFC. Esto, que en un principio puede parecer una ventaja, en la práctica ofrecía un gran problema. Los desarrollos de componentes en Visual Basic escalaban mal por su pobre soporte a tecnologías multi-hilo, y los desarrolladores de Visual C++ se encontraban con una tecnología que sí escalaba muy bien, pero que como contrapartida resultaba compleja, en la que resultaba muy sencillo cometer errores de pérdida de recursos o memoria.

Más tarde había que integrar estos objetos con las páginas activas ASP. El modelo de desarrollo funcional de estas páginas hacía que la mayor parte de los proyectos (salvo honrosas excepciones) desembocase en una amalgama de páginas con “spaghetti code” y con una fuente interminable de errores.

Pero la actual oferta tecnológica de Microsoft es muy distinta. Mediante la tecnología .NET, Microsoft ofrece una serie de tecnologías ideales para el desarrollo de

aplicaciones Web. Por un lado, sus lenguajes (con C# a la cabeza) ofrecen una sintaxis mas moderna y productiva, similar a la de Java, y su modelo de desarrollo de aplicaciones ha ganado muchos puntos en sencillez y potencia.

Comparar las tecnologías .NET y J2EE es en realidad una tarea en algunos aspectos absurda. .NET es una tecnología multipropósito, que puede ser utilizada tanto para el desarrollo de aplicaciones Web como para el desarrollo de aplicaciones con interfaz Windows. J2EE por su lado es una tecnología exclusivamente empresarial, orientada hacia un tipo de desarrollo muy concreto, y que jamás servirá para desarrollar aplicaciones de “ventana” ni aplicaciones móviles. Para estas tareas, SUN distribuye sendos SDK denominados J2SE (Java2 Standard Edition) y J2ME (Java2 MicroEdition).

Por tanto, la comparativa debe de realizarse en lo que a aplicaciones Web empresariales se refiere.

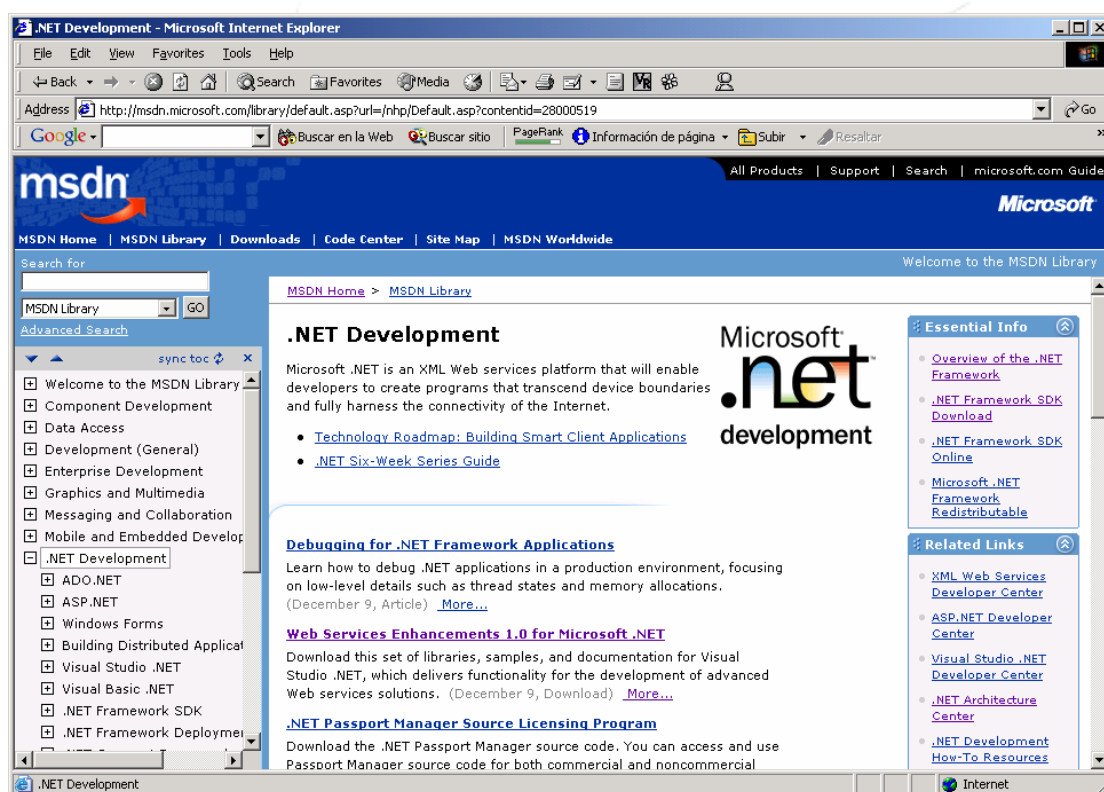


Imagen 2: Página de inicio de la tecnología .NET en el servidor de Microsoft

En este punto, lo primero con que nos encontramos es que J2EE está mucho mas especializado. Todas sus librerías, objetos e interfaces están orientados hacia este tipo de aplicaciones. Por su lado, .NET no ofrece como J2EE un estándar a la hora de desarrollar componentes de reglas de negocio y componentes de acceso a datos. Por el contrario, Microsoft “recomienda” una serie de pautas de desarrollo, dejando al analista/programador la libertad de seguirlos o no. Esto supone una sutil diferencia.

Existe en Internet una comparativa mantenida por la empresa Middleware Company (TMC) sobre la que ha surgido una gran polémica. SUN mantiene desde hace tiempo una aplicación de “ejemplo” denominada “PetStore” (“Tienda de animales”) para que

sirva como ejemplo de desarrollo de aplicaciones bajo J2EE. Microsoft por su lado, realizó una versión en .NET (C#) y se publicaron unos resultados en los que esta última versión salía ganando en rendimiento y facilidad de desarrollo. A partir de este momento, el revuelo fue enorme. Réplicas, contrarréplicas e incluso versiones alternativas de Macromedia (Flash) y el movimiento “open-source” han sido constantes.

La comparativa original habla de rendimientos muy superiores de la versión en .NET (mas del doble de páginas servidas en algunos casos y un coste monetario por transacción muchísimo menor). Nosotros no entraremos en esta discusión mas que para afirmar un punto importante: ambas tecnologías se presentan como mas que aceptables para abarcar un proyecto de este tipo. Tan simple y sencillo como eso.

Esto supone en cualquiera de los casos una ventaja clara para Microsoft, debido a que su tecnología es mucho mas nueva, y ya supone un logro haber alcanzado estas cotas en tan poco tiempo.

Uno de los puntos claros en la tecnología actual, es la interoperabilidad de aplicaciones mediante los denominados “Web Services” o “Servicios Web”. Las tecnologías, lenguajes y protocolos utilizadas para construir y utilizar un “Servicio Web” o “Web Service” son estándares en el mercado (definidos en su mayoría por el Consorcio W3C) y por tanto ofrecen una gran capacidad de comunicación entre aplicaciones, independientemente de la plataforma tecnológica sobre la que están construidas.

En realidad, un servicio Web no se distingue mucho de una aplicación “cliente-servidor” de las de toda la vida. Al fin y al cabo, un “cliente” necesita obtener datos y realizar operaciones en un “servidor”, y para ello utilizará un sistema de llamada a procedimiento remoto, mediante una serie de protocolos y lenguajes bien conocidos. No importa en este caso si detrás del servicio Web hay un servidor de aplicaciones J2EE o un servidor con .NET. Lo importante es saber su dirección y comunicarme con el mediante los protocolos y lenguajes adecuados (HTTP, SOAP, XMLRPC, XML, etc.)Bajo esta perspectiva, los servicios Web no traen nada nuevo al mundo del desarrollo de aplicaciones.

La diferencia fundamental está en las tecnologías que utilizaremos usando los servicios Web. Si las anteriormente citadas aplicaciones “cliente-servidor” se desarrollaban de forma mas o menos aislada unas de otras, la particularidad fundamental de los servicios Web es que todos utilizan una serie de protocolos y lenguajes comunes, haciendo así mucho mas sencilla la labor de interconectar unos con otros, y de construir aplicaciones cliente que sean capaces de entenderse con todos los servicios Web que se construyan.



Imagen 3: La comparativa de rendimiento realizada por TMC

## Conclusión

.NET en la actualidad tiene una gran ventaja, que es su magnífico entorno de desarrollo Visual Studio .NET. Trabajando con J2EE tenemos una variedad mucho mayor de entornos, pero ninguno con una productividad tan alta como la alternativa de Microsoft.

Por otro lado, J2EE cuenta con un parque de instalaciones mucho mayor, pero que en breve puede verse seriamente afectado por .NET. Como comentario final, no debe de escaparse un interesante cambio de posición de la comunidad de desarrolladores. Hace tiempo que nadie dice que “.NET no funciona” o “.NET no sirve”, como se hacía con las páginas ASP, Visual Basic, y el resto de tecnologías de Microsoft. Ahora la discusión se ha vuelto mucho más etérea porque realmente todos los datos y las pruebas hablan a favor de la nueva tecnología. Pero tampoco ninguno habla en contra de J2EE. Ambas tendrán que luchar por un mercado con armas ya no tanto tecnológicas, como comerciales.