

Introducción a .NET - Diferencias entre ASP 3.0 y .NET

Juan Antonio Breña Moral

Índice.

1. Objetos.
2. Librerías
3. CLR.
4. Web services.
5. Conclusiones.
6. Direcciones de interés para más información.

La plataforma de desarrollo, que Microsoft ha lanzado al mercado, es la apuesta de futuro de la empresa de Redmond para impulsar definitivamente las tecnologías de Internet. Las diferencias encontradas entre la tecnología ASP y la tecnología .NET, se basan en los siguientes puntos:

- Objetos.
- CLR.
- Web services.

Objetos.

Cuando Microsoft decidió lanzar la tecnología ASP como parte de su arquitectura DNA, desarrollo un entorno de desarrollo para la construcción de aplicaciones de Internet caracterizado por dividir el entorno de desarrollo ASP en una serie de objetos con los cuales, los desarrolladores tenían en sus manos la capacidad para desarrollar aplicaciones avanzadas.

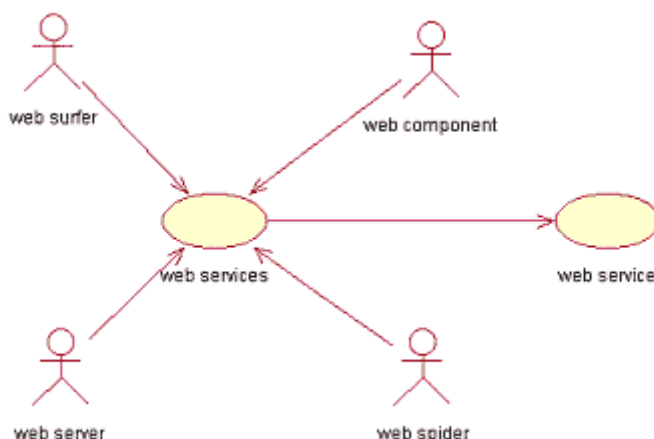
Los objetos que caracterizan ASP son:

- Application
- ASPError
- ObjectContext
- Request
- Response
- Server
- Session

En la actualidad, son cada vez mas complicadas las aplicaciones de esta naturaleza en cuanto a interoperación de múltiples componentes y manipulación de datos empleando numerosas fuentes de información. Con el paso del tiempo una idea brillante surgió sobre la comunidad de desarrolladores a continuación se sintetiza tal idea:

"Si Internet es un conjunto de Host interconectados que sirven información y servicios por que no pensar además que cada Host pueda ofrecer al resto, servicios exclusivos para una nueva generación de aplicaciones distribuidas. De esta manera tenemos como unidad atómica lógica el objeto de negocio y como unidad macro lógica al servicio".

A continuación se muestra el grafico de interoperación de los servicios web.



Como se aprecia en el grafico 1, los servicios web atienden a diferentes tipos de usuarios y los servicios pueden residir en diferentes servidores web.

Parece evidente como ASP no fue construido con ese propósito, pero otras tecnologías competidoras como J2EE, PHP o Cold Fusion tampoco ofrecían mejoras perspectivas.

Con el paso del tiempo diferentes avances se han producido hasta el lanzamiento comercial de la tecnología .NET de Microsoft. La idea de .NET gira en torno a los servicios web como base para la construcción de aplicaciones basadas en tecnologías de Internet para la próxima década. Por tanto el diseño de esta nueva plataforma es totalmente diferente al planteamiento de ASP ya que esta ultima se planteaba como la solución corporativa para la construcción de aplicaciones basadas en componentes COM. Esta fue la base del famoso concepto que inundo la informática corporativa de mediados de los 90's con la inclusión de los componentes COM y su famoso homologo JavaBeans de la tecnología JAVA™. Cabe afirmar como la tecnología de Sun Microsystems parte con bastante mas ventaja debido principalmente a la filosofía con la cual se construyo, ya que cuenta con una excelente colección de clases base y sus módulos que van añadiendo al núcleo de la tecnología. La tecnología .NET nace con una base de clases más potente que Java y con características avanzadas que iremos viendo.

La tecnología .Net divide su FrameWork en una serie de librerías de Clases con las cuales los desarrolladores pueden construir sus aplicaciones distribuidas. A continuación se muestran las diferentes librerías que componen la tecnología .NET:

Librería de sistema.

Las librerías son:

- Collections.
- IO.
- Security.
- Runtime.
- Configuration.
- Net.
- Services.
- InterOp.
- Diagnostics.
- Reflection
- Text.
- Remoting.
- Globalization.
- Threading.
- Serialization.

Librería de datos.

- OleDb.
- Common.
- SqlClient.
- SQL Types.

Librería de XML.

- Serialization.
- Schema.
- XSL.
- Xpath.

Librería de Windows.

- Design.
- ComponentModel.

Librería de Dibujo.

- Drawing2D.
- Printing.
- Imaging.
- Text.

Librería Web.

- Discovery.
- Protocols.

- HTML Web Controls.
- Web Form Controls.
- Caching.
- Configuration.
- SessionState.
- Security.

Tras analizar las librerías del Frame Work de .NET se puede observar como supera enormemente a ASP en todos los frentes y como ASP se ve como un elemento minúsculo en comparación con .NET.

.NET prioriza los siguientes apartados:

Tecnologías de Internet mediante la librería Web.

XML como vehículo transmisor de información.

Incrementa la presencia de modos de tratamiento de la información mediante la librería de datos.

Nueva filosofía de construcción de formularios web.

CLR.

CLR o Common Language Runtime constituye uno de los pilares de la tecnología .NET ya que confiere una nueva visión a sus tecnologías. Antes de .NET, los lenguajes de codificación o bien eran interpretados como era el caso de VScript o Jscript o bien eran compilados a binario como ocurre con lenguajes de programación como C , COBOL o FORTRAN. Con la entrada de JAVA en el mercado de las tecnologías, surge el concepto de VM o Máquina Virtual ya que de esta manera, el lenguaje era compilado a un lenguaje intermedio el cual gracias a la máquina virtual podía ser ejecutado en toda máquina con una máquina virtual, de esta manera surge una de las frases mas celebres en los últimos años de la década de los 90's:

"Escribe el código única vez y ejecuta todas las veces que quieras".

Microsoft adopta esta idea en .NET creando CLR. La diferencia fundamental con respecto a Java y su máquina virtual es que .NET no se limita a un único lenguaje. De esta manera acoge en su seno tecnológico a diferentes masas de desarrolladores con perfiles tecnológicos distintos, si un desarrollador viene de C, adoptara C#, si viene de Visual Basic, adoptara Visual Basic.NET, si viene de ASP puede adoptar o Visual Basic.NET o Jscript.NET. En la actualidad hay anunciados mas de 25 Lenguajes compatibles con la tecnología .NET. Un dato a destacar es el soporte de los lenguajes al paradigma de la programación orientada a objetos. Desde los inicios de la computación, el problema con el cual se encontraban los ingenieros de software era encontrar una manera para diseñar software. La propia naturaleza de la informática hace posible que un determinado problema sea resuelto de múltiples formas, uno de los primeros problemas que fue resuelto, fue la estructuración de los algoritmos y la exclusión de la sentencia GO TO en los algoritmos ya que era contrario al paradigma de la programación estructurada. Una vez superado este escollo en los años 70`s surgen diferentes iniciativas que evolucionan la ingeniería del software, una de las mas importantes fue el desarrollo de Small Talk que revoluciona el panorama con su nuevo enfoque de diseño de software, el paradigma de la programación orientada a objetos. El software desde esa fecha podría ser diseñado con dos enfoques diferentes: Como un conjunto de operaciones bien estructuradas en módulos o bien diseñado como diferentes objetos que están relacionados entre sí para solucionar un determinado problema. Con el paso del tiempo, el paradigma de la orientación a objetos se ha impuesto sobre la programación estructurada debido a sus ventajas en cuanto a diseño, seguridad, escalabilidad y reutilización.

Bien es cierto que el nuevo paradigma confiere al equipo de desarrollo una nueva formación para pensar en Objetos pero los beneficios a medio plazo son evidentes. Microsoft viendo este panorama ha obligado a la comunidad Visual Basic a cambiar a Visual Basic.NET que esta pensado para programarse en orientación a objetos.

Este mismo hecho se aprecia en Jscript.NET y C#.

CLR se caracteriza por ser independiente de sistema operativo por tanto se aprecia como Microsoft ve nuevos frentes donde expandirse o protegerse según se mire. El mercado mundial de sistemas operativos orientado al sector de microinformática esta dominado por la serie Windows. En años recientes un nuevo competidor GNU ha llegado para quedarse, Linux una reciente versión de UNIX esta comiendo mercado debido a sus ventajas heredadas de UNIX y suyas propias. Se puede apreciar como .NET podría desembarcar en Linux empleando .NET o bien podría ser un signo de debilidad del sistema Windows ante la llegada de Linux. CLR puede en un futuro implantarse en otros sistemas operativos distintos de la serie Windows como base para el desarrollo de aplicaciones de escritorio. Como se aprecia esta característica es nueva con respecto a ASP que no da soporte al desarrollo de aplicaciones de escritorio.

Web services.

Los servicios web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software esta distribuido en diferentes servidores.

La informática se inicio con programas monousuarios implantados en grandes ordenadores. Posteriormente estas primeras aplicaciones alcanzaron la capacidad de atender a diferentes usuarios. Pasaron los años y llego la arquitectura cliente-servidor, que gracias a este modelo de desarrollo, la aplicación se dividía en una parte que interaccionaba con el usuario y otra parte destinada al procesamiento de información. En este acercamiento se consiguió que cada una de las partes que constituían la aplicación pudiera residir en computadoras distintas. Con el paso del tiempo, la computación aumento y llego la era de las aplicaciones distribuidas en las cuales los procesos se realizaban en diferentes unidades. De este paso surgió la tecnología Internet para solventar las problemáticas asociadas a fallo de aplicación centralizado. Véase documentación acerca de los orígenes de Internet.

Como punto final a esta cronología, los web services son un paso adelante en la computación ya que de esta forma un ordenador ya no se considerara como un núcleo de computo sino como un repositorio de servicios de n aplicaciones distribuidas por internet. Para comprender este concepto pondremos el ejemplo siguiente:

Imaginemos el supuesto de un servidor X que da 2 servicios, el servicio A y el servicio B. También nos dan el caso de que tenemos otro servidor Y que ofrece el servicio C y el servicio D. Por ultimo también analizamos el servidor Z que da 2 servicios el servicio E y el servicio F.

Imaginar que el servidor W ofrece una aplicación on-line que se compone de su servicio principal G y de la inter operación con A, B, C, D, por tanto se da que los servicios A, B, C, D dan servicio a G independiente de su localización en G pero también están disponibles para el resto de Internet si así se ha diseñado. Por tanto viendo los datos anteriores vemos que las posibilidades combinatorias son grandes y sobre todo por primera vez y de manera mas transparente que con el uso de componentes, el uso del software se hace mas reutilizable ya que diferentes servicios pueden ser utilizados como componentes en una aplicación.

Microsoft para conseguir este propósito con su tecnología .NET emplea como protocolo de comunicación, una aplicación XML, llamada SOAP.

¿Qué es SOAP? Son las siglas de Simple Object Access Protocol. Este protocolo deriva de un protocolo creado por David Winer, XML-RPC en 1998. En su sitio web, Userland, <http://www.userland.com> se puede encontrar multitud de documentación acerca de este primer protocolo de comunicación bajo http mediante XML. Con este protocolo se podían realizar RPC o remote procedure calls, es decir, podíamos bien en cliente o servidor realizar peticiones mediante http a un servidor web. Los mensajes debían tener un formato determinado

empleando XML para encapsular los parámetros de la petición. Con el paso del tiempo el proyecto iniciado por David Winer interesó a importantes multinacionales entre las que se encuentran IBM y Microsoft y de este interés por XML-RPC se desarrolló SOAP.

SOAP es un protocolo mas completo que XML-RPC pero cabe decir que mas complejo.

La siguiente tabla comparativa muestra las diferencias entre ambos protocolos:

Características	XML-RPC	SOAP
Escalares básicos.	yes	yes
Estructuras.	yes	yes
Arrays.	yes	yes
Estructuras nombradas y Arrays.	no	yes
Manejo de fallos.	yes	yes
Curva de aprendizaje.	yes	no
Conjunto de caracteres.	no	yes (US-ASCII, UTF-8, UTF-16)
Tipos de datos definidos por usuario.	no	yes
Requiere entendimiento del cliente.	no	yes
Instrucciones de procesamiento Especificas.	no	yes

A continuación se muestra un ejemplo de SOAP:

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "http://example.org/2001/06/quotes"
```

```
<env:Envelope xmlns:env="http://www.w3.org/2001/06/soap-envelope">
  <env:Body>
    <m:GetLastTradePrice
      env:encodingStyle="http://www.w3.org/2001/06/soap-encoding"
      xmlns:m="http://example.org/2001/06/quotes">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </env:Body>
</env:Envelope>
```

En próximos artículos mostraremos la construcción de web services empleando tecnología .NET de Microsoft.

Conclusiones.

Tras el análisis de las 2 tecnologías de Microsoft, se puede comprobar como ASP 3.0 tiene un digno sucesor, pero hay que darse cuenta que no es simplemente una actualización de la tecnología para desarrollo de aplicaciones web, sino que es un nuevo enfoque de computo para el desarrollo de software. .NET supone un digno adversario para J2EE y como se puede comprobar en diferentes blue prints, supera esta última en algunos aspectos.

Si profundizamos en cuanto a todas las características que atesora, se aprecia como adquiere de la tecnología de JAVA muchas de sus funcionalidades pero con acceso más sencillo y con la facilidad de poder emplear diferentes lenguajes de codificación pero sin perder potencia de diseño ya que los principales: C#, Visual Basic.NET y Jscript.NET son orientados a objetos. Además gracias a su nuevo modo de desarrollo de formularios web empleando los controles web se puede concluir diciendo que en conjunto el desarrollo de cliente a nivel funcional, no en términos estéticos es más poderoso que los mejores clientes de Flash MX. Por tanto toda la estrategia de Macromedia con su plataforma MX está en peligro.

En próximos artículos describiremos como funciona la clave tecnológica de .NET que son los servicios web y como funciona SOAP, WSDL. Además realizaremos una serie de ejemplos prácticos para el desarrollo de servicios web.

Direcciones de interés para más información:

<http://www.userland.com>

<http://www.dotnet.com>

<http://www.topxml.com> ⁷

<http://www.oreilly.com/catalog/progxmlrpc/>

<http://xmlrpc.usefulinc.com/>

<http://xmlhack.com/>

<http://www.byte.com/heller/2001/09/>

<http://www.xmlrpc.com/>

<http://scottandrew.com/xml-rpc/test-mirror.html>

<http://www.ashleyit.com/rs>

<http://www.oreilly.com/catalog/webserveess/index.html>

<http://www.google.com/apis/download.html>

<http://www.w3schools.com/default.asp>

<http://www.webreference.com/js/column107/>

<http://msdn.microsoft.com/vstudio/productinfo/default.asp>