

[monotematicos](#) ▾ [secciones](#) ▾ [servicios](#) ▾ [comunidad](#) ▾[Inicio](#) > [Manuales](#) > [Diferencias entre ASP 3.0 y .NET](#)

#### Indice del Manual

- + [Introducción a las Diferencias entre ASP y .NET](#)
- + [Objetos de ASP 3.0 y .NET](#)
- + [CLR](#)
- + [Web Services](#)
- + [Conclusiones a las diferencias entre ASP y .NET](#)

#### Descripcion capítulos

## 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 interoperacion 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

regis  
don  
con to

· redirecci  
· página de  
· panel de  
· gestión d



We  
Maestr  
WebEs  
Ciberte

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                          |
|---|---------|-------------------------------|
| Escalarares 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:encodingStyle="http://www.w3.org/2001/06/soap-encoding"

xmlns:m="http://example.org/2001/06/quotes">




DIS
```

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



Informe de [Juan Antonio Breña Moral](#) \*  
URL: <http://www.esmeta.com/>

\* Para consultas técnicas utilizar la [lista de correo](#).

-  [Versión imprimible del artículo](#)
-  [Enviar artículo por e-mail](#)
-  [Publicar un comentario del artículo](#)

[↑ Ir arriba](#)