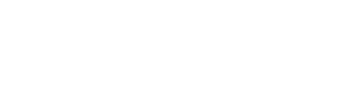
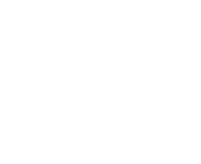


FP



**2º DAW**

http://iescomercio.com/images/fotos/logos/Logo2-iescomercio-77x45.png

Servidores de aplicaciones Java EE y Microsoft IIS

*[DWES] del Ciclo Formativo Desarrollo de Aplicaciones Web*

**Trabajo personal**

*Marcos Blázquez Villagrán*

*20-10-2016*

Contenido

[Servidores de aplicaciones 3](#_Toc465426950)

[¿Qué son? 3](#_Toc465426951)

[Servidor de aplicaciones Java EE 3](#_Toc465426952)

[Arquitectura de un servidor J2EE 4](#_Toc465426953)

[Extender un servidor J2EE 6](#_Toc465426954)

[Servidor IIS de Microsoft 7](#_Toc465426955)

[Arquitectura en un servidor IIS de Microsoft 7](#_Toc465426956)

[Componentes en modo Kernel 7](#_Toc465426957)

[Componentes en modo Usuario 8](#_Toc465426958)

[Flujo de trabajo e interacción entre componentes 8](#_Toc465426959)

[Extensiones para IIS 9](#_Toc465426960)

[Fuentes: 10](#_Toc465426961)

# Servidores de aplicaciones

## ¿Qué son?

Un servidor de aplicaciones es un servidor en una red de ordenadores que ejecuta ciertas aplicaciones. Normalmente se trata de un dispositivo software que proporciona servicios a un ordenador cliente gestionando la mayor parte de las funciones de la lógica de negocio y de acceso a los datos de la aplicación. Los beneficios del uso de esta tecnología son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

El concepto de servidor de aplicaciones está relacionado con el concepto de sistema distribuido que permite mejorar tres aspectos fundamentales en una aplicación: la alta disponibilidad, la escalabilidad y el mantenimiento.

La **alta disponibilidad** hace referencia a que un sistema puede estar funcionando las 24 horas al día durante todos los días del año, esto es posible gracia a las técnicas de balanceo de carga y recuperación de fallos (failover).

La **escalabilidad** es la capacidad de hacer crecer el sistema cuando se incrementa la carga de trabajo, en el caso de un servidor de aplicaciones esto se refiere a la cantidad de peticiones que son recibidas. Cada máquina tiene una capacidad limitada de recursos y por lo tanto solo puede recibir un número determinado de peticiones. En el caso de que se incremente la demanda de servicio se incorporarán nuevas máquinas para poder abastecer a todas las peticiones.

Un buen **mantenimiento** exige un buen mantenimiento del sistema, depuración de fallos y actualización. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares.

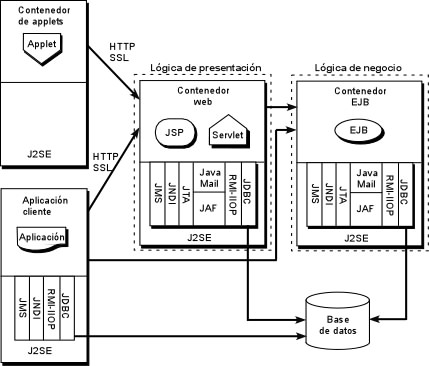
## Servidor de aplicaciones Java EE

Entre los principales servidores de aplicaciones Java encontramos WebLogic de Oracle y WebSphere de IBM (los 2 son software propietario). Por otro lado, tenemos EAServer que también ofrece soporte a otros lenguajes diferentes a java. Entre los servidores de aplicaciones libres podemos encontrar JOnAS, JBoss AS, Geronimo y TomEE de apache, Blazix, Enhydra Server y GlashFish de Oracle.

Es común que los servidores de aplicaciones incluyan Middleware, este es un software que permite la conectividad con otros servicios, para efectos de confiabilidad, seguridad, no-repudio, etc… Los servidores de aplicación brindan a los desarrolladores una API (Interfaz para programación de aplicaciones) para que no tengan que preocuparse por el sistema operativo o la interfaz requerida en una aplicación web moderna.

Un buen ejemplo de la utilización de servidores de aplicaciones son los portales de internet, que permiten la gestión y divulgación de información a las empresas, además de un único punto de entrada a los usuarios tanto internos como externos. Esto se realiza teniendo como base un servidor de aplicación que permite tener acceso a información y servicios de manera segura y transparente desde cualquier dispositivo.

## Arquitectura de un servidor J2EE



Podríamos decir que la arquitectura está dividida en 4 partes:

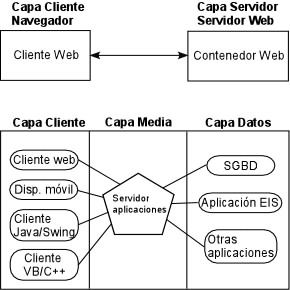
En primer lugar, está el **cliente web o contenedor de applets**, este es normalmente un navegador e interactúa con el contenedor web por medio de HTTP. Recibe páginas HTML o XML y además puede ejecutar applets y código JavaScript.

Después tenemos la **aplicación cliente** que son clientes que no se ejecutan dentro de un navegador y que pueden utilizar cualquier tipo de tecnología para comunicarse con el contenedor web o directamente con la base de datos.

El **contenedor web** es lo que denominamos servidor web. Este es la parte visible del servidor de aplicaciones y utiliza los protocolos HTTP y SSL (protocolo de seguridad) para comunicarse.

Por último, tenemos el **servidor de aplicaciones** que proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas, podríamos decir que es el corazón de un gran sistema distribuido.

Un servidor de aplicaciones tiene una estructura en tres capas que permite estructurar el sistema de forma más eficiente que la estructura en 2 capas. En la imagen inferior se muestra la diferencia entre la arquitectura en dos capas frente a la de tres capas utilizando el servidor de aplicaciones.



No todas las aplicaciones empresariales necesitan de un servidor de aplicaciones, en el caso de tener una pequeña aplicación que acceda a una base de datos sencilla y que no sea distribuida no necesitará un servidor de aplicaciones, tan solo con un servidor web (utilizando Servlets y JSP) sería suficiente.

## Extender un servidor J2EE

Los Java Servlets son aplicaciones Java que se ejecutan en el servidor bajo una arquitectura cliente-servidor, extendiendo las capacidades del servidor web. Dentro de las principales APIS (Interfaces de programación de Aplicaciones) para el desarrollo de la tecnología Java, se encuentran algunas que permiten extender la capacidad de los servidores de aplicaciones http y utilizarlos como respuestas sobre servicios web.

Entre estos servicios podemos encontrar:

**Los Servlets**: Este es uno de los más importantes pero el servidor donde se ejecuten debe contar con una máquina virtual de java. Los servlets responden a eventos generados en los clientes desde requerimientos HTML, y permiten construir una respuesta dinámica a estos. Los servlets son una mejor alternativa a otras tecnologías como CGI y presentan ventajas, como por ejemplo la portabilidad e independencia de la plataforma.

También tenemos otros mecanismos para extender los servidores de aplicaciones J2EE como:

**MS** (Java Message Service API): es un mensaje estándar que permite a componentes de aplicaciones J2EE crear, enviar, recibir y leer mensajes, habilitando comunicaciones distribuidas (asincrónicas).

**TA** (Java Transaction API): provee una interfaz estándar para transacciones demarcadas. Maneja un auto commit por defecto para transacciones commits y rollbacks (una aplicación con manejo de datos que puede actualizarlos luego de cada operación de escritura o lectura).

**avaMail** **API**: API para envío de notificaciones email en Internet.

**AF** (JavaBeans Activation Framework): provee servicios para determinar tipos de datos dentro de Java Mail API.

**JAXP** (Java API for XML Processing): procesamiento de documentos XML usando Document Object Model (DOM), Simple API for XML (SAX) y Extensible Stylesheet Language Transformations (XSLT).

**AX-RPC** (Java API for XML-Based RPC): Esta API usa el estándar SOAP y HTTP para que el cliente pueda programar procedimientos en XML llamados RPCs sobre Internet.

**AAJ** (SOAP with Attachments API for Java): API de bajo nivel que habilita la producción y consumo de mensajes SOAP 1.1.

**AXR** (Java API for XML Registries): registra el acceso de negocio y propósito general de aplicaciones sobre la red.

**DBC** (Java Data Base Connection): Quizás el API de J2EE más conocida, permite el trabajo con base de datos permitiendo comandos SQL para la programación de métodos de acceso a distintas bases de datos. Posee dos partes: una de nivel de aplicación usada por el componente que accede a los datos.

# Servidor IIS de Microsoft

Todas las versiones de Windows incluyen un servidor web local llamado: “Internet Information Services” también llamado IIS.

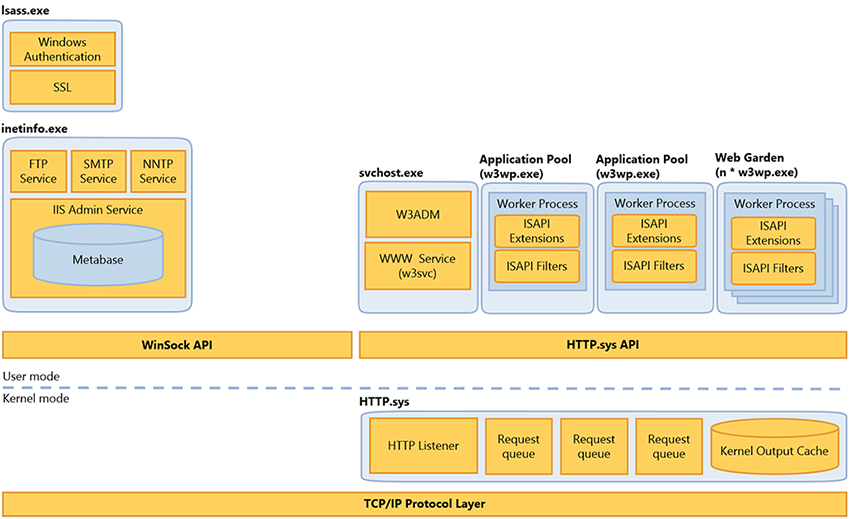
Este servidor de aplicaciones es poco conocido y utilizado por los usuarios de Windows, los que en ocasiones se descargan software externo para instalar un servidor en su ordenador sin saber que su sistema operativo posee uno.

IIS es un servidor web y un conjunto de servicios para el sistema operativo Windows que convierte el Pc en un servidor web para internet o intranet, es decir un servicio instalado con el que publicar páginas web tanto en local como remotamente.

Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas. Por ejemplo, incluye los módulos de ASP (Active Server Pages) y ASP.NET. Por otro lado, también se pueden incluir módulos de otros fabricantes como PHP o Perl.

## Arquitectura en un servidor IIS de Microsoft

Los componentes de IIS están divididos en 2 grandes grupos, por un lado, están los componentes que se encuentran en el modo Kernel, es decir, los que vienen incorporados con el sistema operativo Windows y por otro lado tenemos los componentes que están incorporados en procesos de usuario.



## Componentes en modo Kernel

Están ubicados por encima de la capa de protocolo TCP/IP y que se comunican con el driver TCPIP.sys. En el caso de IIS se utiliza el driver HTTP.sys cuya función es escuchar todo el tráfico HTTP por medio del HTTP Listener, interpretarlo y asegurarse de que el tráfico es válido. Una vez verificado la validez del tráfico, HTTP.sys coloca las peticiones en el “request queue” o cola de solicitudes correspondiente para que un worker process las procese.

## Componentes en modo Usuario

Uno de los componentes del modo usuario es **inetinfo.exe**, en él se ejecutan los servicios SMTP, FTP, NMTP y el servicio de administración de IIS, cuyo propósito es administrar y cargar la metabase (archivo de configuración) de IIS en memoria.

Por otro lado, tenemos los **application pools** que son una agrupación de URLs enrutadas a uno o más worker process, este a su vez es responsable de procesar las peticiones web generando una salida como respuesta. Podemos detallar también que los **Web gardens** son application pools que utilizan más de un worker process en un solo servidor.

El procesosvchost.exe es el encargado de alojar los servicios W3ADM Y WWW Service. Para que la comunicación entre estos 2 servicios y HTTP.sys funcione correctamente, deben configurar HTTP.sys para que esté en la capacidad de notificar a estos servicios al momento de necesitar un worker process para procesar una petición.

Por último, el componente Isass.exe es utilizado para realizar autenticación Windows y operaciones de decodificación a través de SSL.

## Flujo de trabajo e interacción entre componentes

Al iniciarse el servidor que corre IIS, el servicio IIS Admin Service lee la información del archivo Metabase.xml y crea su representación en memoria. El servicio WWW Service lee esta información e inicializa la tabla de enrutamiento de HTTP.sys con una entrada para cada application pool. HTTP.sys utilizará esta tabla para determinar que application pool corresponde a las peticiones entrantes.

Cuando HTTP.sys recibe una petición, la validez de esta es verificada. En este punto pueden ocurrir dos cosas:

* Que la petición sea inválida: En este caso HTTP.sys enviará al cliente el error correspondiente.
* Que la petición sea válida: En este caso HTTP.sys verifica la posibilidad de dar respuesta a la petición desde su caché. Si existe en caché la envía inmediatamente, de lo contrario HTTP.sys coloca la petición en la cola de peticiones para el application pool correspondiente según la tabla de enrutamiento.

Posteriormente HTTP.sys valida si existe un worker process que pueda procesar la petición, dando lugar a dos escenarios adicionales:

Si no hay un worker process para procesar la petición, HTTP.sys notifica a W3ADM para que éste inicie y configure uno basado en la información alojada en la memoria según lo encontrado en el archivo de la metabase.

Si hay uno o más worker process ejecutándose y escuchando la cola de peticiones, el worker process coge la petición de la cola, la procesa y envía la respuesta a HTTP.sys y al cliente.

## Extensiones para IIS

Hay gran cantidad de extensiones para los servidores Internet Information Services (IIS) de Microsoft, estas añaden nuevas funcionalidades entre ellas podemos destacar:

**Servicio de publicación FTP**: Servidor web con autenticación basada en SSL y transferencia de datos.

**Paquete de administración**: Agrega una interfaz de ayuda para administrar los componentes del servidor IIS, incluye autorización ASP.NET, configuración FastCGI, filtro de peticiones y personalización de errores.

**Application Request Routing**: Proporciona un módulo de enrutamiento basado en proxy que reenvía solicitudes HTTP a servidores de contenido basado en cabeceras HTTP variables de servidor y algoritmos de equilibrio de carga.

**Gestión de la base de datos**: Permite fácilmente gestionar de forma local o remota la base de datos.

**Servicios de contenido multimedia**: Integra una plataforma para poder gestionar y administrar contenido multimedia o contenido web.

**Módulo de reescritura URL**: Provee un mecanismo de reescritura basado en reglas para el cambio de direcciones URL antes de que sean procesados por el servidor web.

**WebDav**: Permite publicar contenidos de forma segura y administrar las opciones WebDav tanto a administradores como a proveedores de alojamiento.

**Herramienta de implementación web**: Sincroniza los servidores IIS y migra los servidores IIS6 a IIS7, además puede desplegar las aplicaciones en un servidor IIS7.

# Fuentes:

<https://es.wikipedia.org/wiki/Servidor_de_aplicaciones>

<http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm>

<https://es.wikipedia.org/wiki/Internet_Information_Services>

<https://blogs.msdn.microsoft.com/cmarin/2014/09/30/arquitectura-de-iis-6/>

<https://en.wikipedia.org/wiki/Internet_Information_Services#Extensions>