

# Arquitecturas y lenguajes de programación

---

Servlets, JSP y EJB

Inés Castel

Este proyecto trata sobre las arquitecturas que disponen los lenguajes de programación, principalmente Servlets, JSP (Java Server Pages) y EJB (Enterprise JavaBeans). Muestra y compara sus principales usos y características proporcionando una visión conjunta de las tres estructuras.

## ÍNDICE

<b>JSP .....</b>	<b>3</b>
Características .....	3
Ejecución .....	4
Ubicación.....	4
Ventajas.....	5
Desventajas .....	5
<b>Servlets .....</b>	<b>6</b>
Características .....	6
Ejecución .....	7
Ubicación.....	7
Ventajas.....	8
Desventajas .....	9
Ciclo de vida .....	9
<b>EJB .....</b>	<b>10</b>
Características .....	10
Ejecución .....	11
Ventajas.....	12
Desventajas .....	12
<b>Fuentes.....</b>	<b>13</b>

## JSP

Es un lenguaje multiplataforma, pensado para desarrollar páginas web dinámicas principalmente en Java. Se ejecuta en el lado del servidor. Está soportado por múltiples plataformas como ASP y PHP.

Fue desarrollado por Sun Microsystems.

Este tipo de tecnología permite desarrollar y mantener fácilmente páginas dinámicas con una gran cantidad de información. Separa la interfaz del usuario de la parte lógica permitiendo cambiar las plantillas de la forma que más se ajuste a las necesidades del diseñador, sin modificar el contenido.

JSP facilita herramientas para la introducción de código en la generación dinámica de HTML.

### Características

El lenguaje JSP presenta las siguientes características:

- Complementa las bases de java en el entorno del servidor con la flexibilidad que presenta el lenguaje HTML.
- Permite también el uso de XML y WML.
- Permite reusar componentes con JavaBeans.
- Sabe cómo resolver una solicitud para crear una respuesta automática.
- Soporta contenido de tipo dinámico.
- Es un lenguaje rápido y fácil de utilizar.
- Puede instanciar cualquier clase de Java.
- Permite ejecutar aplicaciones en múltiples plataformas sin cambios.
- Separa en niveles la creación de aplicaciones web dejando la parte que genera el HTML en el archivo JSP.

## Ejecución

Los JSP se ejecutan en una máquina virtual. Esto permite que se puedan utilizar desde cualquier tipo de ordenador en el que se encuentre instalada una máquina virtual Java.

Cada JSP se ejecuta en su propio hilo y realiza las peticiones de forma ordenada, pasando de una a la siguiente. De esta forma se evita la pérdida de tiempo en invocarlo.

También permite realizar la conexión con la base de datos y controlar el manejo de sesiones de una forma muy eficiente.

El proceso de ejecución es el siguiente:

1. Se compila el JSP a Servlet: al convertir un JSP a Servlet eliminamos visualmente las funciones de arranque, ejecución y terminación.
2. Carga el JSP.
3. Se inicializa
4. Se ejecuta.
5. El JSP se limpia y restaura.

## Ubicación

JSP se encuentra principalmente ubicado en el entorno del servidor, es decir, se ejecutan en el servidor antes de que la información sea enviada al cliente a través de internet.

Para utilizar Java Server Pages es necesario un servidor web compatible con Servlet (Apache Tomcat o Jetty).

Tiene acceso a bases de datos, conexiones de red y diversas tareas.

Estos tipos de lenguajes son independientes del cliente y permiten el cambio de un servidor a otro, así como una actualización más eficiente de las versiones del mismo.

Los scripts almacenados en el servidor permanecen ocultos para el cliente. De esta forma el trabajo realizado presenta una mayor protección.

## Ventajas

Una de las principales ventajas de los lenguajes JSP es que permite agregar código Java a una página que ha sido diseñada con HTML.

También puede generar código HTML en el cliente dinámicamente para controlar situaciones en las cuales la información se encuentre en el lado del cliente.

Otra ventaja importante se encuentra en el código. Presenta un código muy bien estructurado y fácil de comprender.

Como la parte dinámica de JSP se encuentra escrita en Java, permite una integración total con sus módulos, así como el uso de un motor de páginas basado en Servlets.

## Desventajas

La principal desventaja de JSP es que como principalmente se encuentra en el entorno del servidor, presenta un alto riesgo de sobrecargar al mismo. Tampoco podremos acceder a la aplicación si no disponemos de conexión con la red.

Es un lenguaje complejo que necesita mucho tiempo de aprendizaje.

Es necesario un servidor Tomcat para hacer posible el alojamiento web.

Presenta una interfaz muy limitada y su carga es más lenta que una aplicación de escritorio.

## Servlets

Servlet es un tipo de lenguaje de Java que se utiliza principalmente para ampliar servidores. Extienden las aplicaciones que se encuentran en los servidores web y se ejecutan en los propios servidores.

Pertenecen a otras tecnologías dinámicas, como por ejemplo PHP ó ASP.NET.

Se encargan de generar páginas web dinámicas mediante los parámetros enviados por el navegador web en la petición.

Fue introducido por Sun en 1996.

Los Servlets se ejecutan desde el servidor. Cuando un cliente ejecuta un Servlet, este genera una página html con los resultados obtenidos y se la envía al cliente.

Se utilizan para generar páginas web de forma dinámica a partir de los parámetros enviados en el navegador web a través de la petición.

## Características

Los sistemas Servlets presentan las siguientes características:

- Son independientes del servidor y del sistema operativo que se utiliza. Gracias a esta característica el servidor puede estar programado en cualquier lenguaje.
- Pueden llamar a otros Servlets ó a los métodos de otros Servlets para distribuir el trabajo de forma más eficiente.
- Son capaces de obtener información sobre el cliente, como su dirección IP, el puerto de la llamada, el método ...
- Permiten el uso de cookies y sesiones, lo que permite almacenar información sobre un usuario determinado.
- Pueden actuar como enlace entre el cliente y las bases de datos.
- Pueden realizar tareas de proxy para un applet, ya que estas se encuentran limitadas por las restricciones de seguridad.
- Generación dinámica de código HTML.

## Ejecución

Los Servlets se ejecutan a través de un Motor de Servlets.

Un motor de Servlets es un software que se instala en el servidor que se encarga principalmente de comprobar cuando se ejecuta un Servlets, los parámetros que recibe desde un cliente HTTP, pasarlos a los programas, recibir su resultado y enviarlo al cliente correspondiente.

Si queremos ejecutar un Servlet necesitamos disponer de este tipo de motores. El motor gratuito más conocido es el Jakarta Tomcat, desarrollado por Apache.

## Ubicación

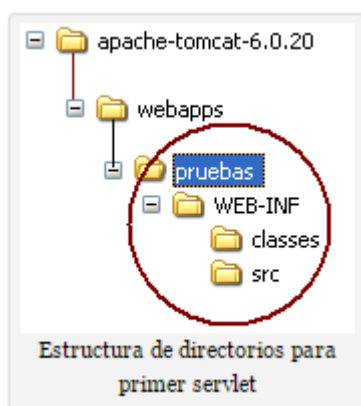
Una vez que se inicia el motor de Servlets, por ejemplo Tomcat, se crea una estructura de directorios que permiten ejecutar el Servlet.

Dicha estructura se forma en el directorio webapp, que se encuentra en el directorio de Tomcat.

Debemos crear un directorio para alojar el proyecto dentro de webapps. Supongamos que nuestro directorio se llama pruebas.

En el directorio anterior se creará WEB-INF, y dentro tendremos dos directorios más: clases (que almacena el Servlet compilado) y src (que guardará el archivo fuente del Servlet).

Por lo tanto, la estructura de ubicación quedará de la siguiente manera:

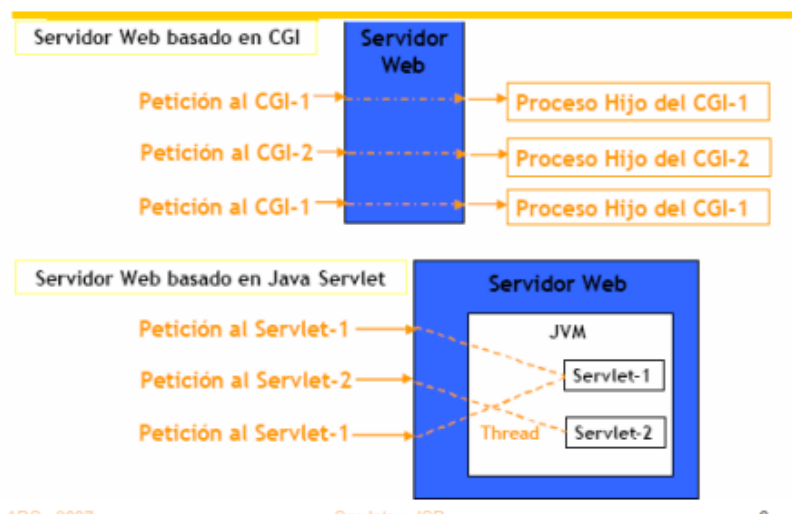


## Ventajas

El uso de esta tecnología presenta numerosas ventajas:

Los Servlets se cargan la primera vez que son ejecutados y no necesitan volver a cargar. Cada petición crea un hilo nuevo. Esto evita la sobrecarga en el servidor y hace que se ejecute de forma más eficiente reduciendo el uso de memoria y el tiempo de respuesta.

Comparación de la creación de procesos entre CGI y Java Servlet:



Los motores de Servlets también administran los recursos del servidor de forma más eficiente.

Pueden mantener varias conexiones abiertas que se utilizarán cuando sea necesario.

Se pueden ejecutar en cualquier sistema que disponga de una máquina virtual Java y un servidor web, por lo que dispone de una alta portabilidad y seguridad, evitando la mayoría de los problemas que se encuentran en los CGIs.

Al estar programados en java pueden utilizar todas las clases y herramientas de dicha plataforma, por lo que adquieren gran potencia de ejecución.

Tienen persistencia, ya que siguen vivo cuando se termina la petición.



## Desventajas

Una de las desventajas principales de Servlet es que si es muy especializado debemos tener cuidado con el tipo programas o aplicaciones que realicemos basados en esta tecnología, ya que el Servlet se mantendrá vivo en memoria y puede consumir rápidamente los recursos del servidor y generar algunos problemas de rapidez y eficiencia.

## Ciclo de vida

### 1. Inicializar el Servlet

Cuando un servidor inicia un Servlet, se ejecuta el método `init`. Este proceso se completará antes de empezar a responder peticiones de los clientes y antes de que el Servlet se destruya.

El método `init` se ejecutará sólo una vez a no ser que se recargue de nuevo el Servlet.

Antes de recargar un Servlet, será necesario destruir el Servlet actual a través del método `destroy`.

### 2. Interacción con los clientes

Una vez inicializado, el Servlet comenzará a responder a las peticiones de los clientes, que serán atendidas por la misma instancia del Servlet.

### 3. Destrucción del Servlet

Los Servlets son ejecutados hasta que el servidor los destruye por el mismo cierre del servidor ó por petición del administrador.

Para destruir un Servlet el servidor ejecutará el método `destroy` sólo una vez. Este método puede ser llamado aunque queden aún respuestas en proceso.

Si el servidor quiere ejecutar de nuevo otro Servlet, deberá repetir todo el proceso anterior.

## EJB

Las Enterprise JavaBeans son una de las interfaces de programación de aplicaciones que forman parte del estándar J2EE pensado para la construcción de aplicaciones empresariales.

Fueron desarrolladas por Sun Microsystem y actualmente pertenecen a Oracle Corporation.

Se ejecuta en el lado del servidor de forma distribuida.

Su objetivo es abstener al programador de los problemas que presenta una aplicación empresarial, como concurrencia, persistencia, seguridad...

Se centra principalmente en el desarrollo de la lógica de negocio.

Como está basado en componente, permite gran flexibilidad y reutilización de los mismos.

## Características

Las características principales de las interfaces de programación EJB son las siguientes:

- Permiten desarrollar aplicaciones débilmente acopladas.
- Contiene la lógica de negocio que opera con Enterprise Information System.
- Su comportamiento está basado en interfaces.
- Las características de seguridad y las transacciones están separadas de las clases EJB, permitiendo la operación de aplicaciones externas.
- Se puede configurar su funcionamiento en tiempo de despliegue de forma declarativa.
- Las instancias son creadas y utilizadas por el contenedor de EJB.
- Los componentes EJB son multiplataforma.
- Se pueden editar sus parámetros de configuración a través de archivos XML.

## Ejecución

Los EJB se encuentran en un contenedor EJB, n el servidor de aplicaciones. El EJB se comunica con el contenedor y el código del cliente a través de la especificación.

Cada EJB dispone de una clase y dos interfaces (home e interfaz remota) implementadas en Java.

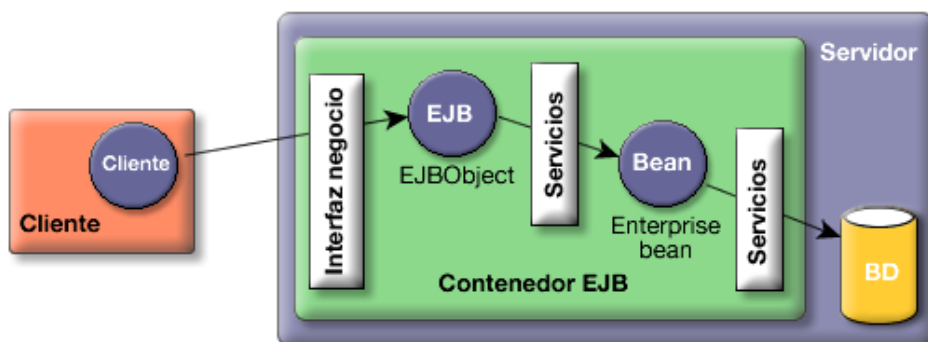
Dicho contenedor crea instancias de la propia clase facilitando así su implementación. El código del cliente es el que utiliza las interfaces especificando las firmas de los métodos remotos. Estos métodos se dividen en dos grupos:

1. Métodos no ligados a una instancia específica. Se encuentran declarados en la interfaz home.
2. Métodos ligados a una instancia específica. Se encuentran declarados en la interfaz remota.

El contenedor crea clases para las interfaces, que actúan como un proxy del cliente. El cliente llama a un método del proxy y éste envía un mensaje al servidor EJB. El proxy se comunica con el servidor mediante RMI-IIOP.

Una vez que el mensaje ha llegado al servidor, controlará la llamada a través de un método de la instancia.

Ejemplo del funcionamiento de los Enterprise JavaBeans:



## Ventajas

Las principales ventajas de los sistemas EJB son las siguientes:

Interoperabilidad entre aplicaciones. Su arquitectura facilita la integración de aplicaciones entre los diferentes vendedores.

Permiten la integración con sistemas no-Java. Algunas de las especificaciones como Connector ó los beans que manejan los mensajes, permiten integrar sistemas que no están desarrollados en Java, como ERPs ó mainframes.

Contiene recursos educativos y herramientas de desarrollo. Estas herramientas facilitan el trabajo que realiza el desarrollador de aplicaciones EJB.

Posibilidad de construir aplicaciones completas, ya que la propia arquitectura basada en componentes y establecida en interfaces simplifica el desarrollo de las mismas.

La aplicación dispone de portabilidad en cualquier servidor que soporte J2EE.

## Desventajas

Los sistemas EJB presentan algunas desventajas, por ejemplo:

El desarrollo de una aplicación con EJB es realmente complejo.

Como es uno de los principales componentes de J2EE, depende de otras partes del estándar, cómo RMI, JNDI ó JDBC, lo que hace que en ciertas ocasiones no tenga un buen soporte.

## Fuentes

Para realización de este proyecto se han consultado las siguientes fuentes:

<http://aplicaciones-web-lenguajes-programaci.blogspot.com.es/2011/12/jsp.html>

[https://es.wikipedia.org/wiki/JavaServer\\_Pages](https://es.wikipedia.org/wiki/JavaServer_Pages)

<http://www.registrodominiosinternet.es/2013/08/lenguajes-programacion-web-ventajas.html>

<https://blogdeaitor.wordpress.com/2008/10/20/servlet-frente-a-jsp/>

<http://library.gxtechnical.com/gxdlsp/pub/genexus/java/docum/manuals/7.5/mjava75spa4.htm>

<https://kikev.wordpress.com/2009/07/29/ejecutar-un-servlet-por-primera-vez-sin-dolores-de-cabeza/>

<http://www.programacionfacil.com/java/servlets.html>

<http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-traspas.pdf>

<https://users.dcc.uchile.cl/~jbarrios/J2EE/node44.html>

<http://www.profesys.com/Soluciones/EJB3.htm>

[https://es.wikipedia.org/wiki/Java\\_Servlet](https://es.wikipedia.org/wiki/Java_Servlet)