

# Servlets, JSP y EJB.

Curso 2016/2017

## TRABAJO SOBRE:

Definición de cada tipo, usos, para qué tipo de app es mejor cada uno...

**Aida López Sáez**

Desarrollo web en entorno servidor, 2ºDAW.

## Contenido

○ Servlets .....	2
- Definición .....	2
- Propiedades.....	2
- Usos de los servlets.....	2
¿Para qué sirven?.....	3
○ JSP (Java server pages).....	3
- Definición .....	3
- Propiedades:.....	4
- Usos de JSP.....	4
¿Para qué sirven?.....	4
○ EJB (Enterprise JavaBeans) .....	4
- Definición: .....	4
- Propiedades:.....	5
- Usos de los EJB: .....	5
¿Para qué sirven?.....	6
○ Conclusiones:.....	6
¿Es indiferente que se use uno en vez del otro?¿Qué característica diferenciadora tiene cada uno de ellos? .....	6
○ Bibliografía.....	6
- Servlets: .....	6
- JSP.....	7
- EJB.....	7

## ○ Servlets

### - Definición

Es un lenguaje de programación que nos permite realizar aplicaciones web interactivas, le permite al cliente modificar el contenido de ésta a su antojo.

Son pequeños programas que reciben peticiones desde el navegador a través del protocolo HTTP, procesan la información y las devuelve en formato HTML. El lenguaje es desarrollado por Oracle y Sun Microsystems, por lo tanto es parte del famoso lenguaje Java. Es más, es un objeto que pertenece a una clase de la librería javax.servlet.http.HttpServlet pudiendo usar si es necesario pudiendo usar otras clases incluidas en Java.

Todos estos programas son gestionados por otro programa capaz de recibir las peticiones de paginas web y redireccionar éstas a un objeto Servlet. Un ejemplo del “Hola Mundo” con un input donde se introduce el nombre:

```
public class MiPrimerServlet extends HttpServlet
{
    public void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        PrintStream out = new PrintStream(res.getOutputStream());
        res.setContentType("text/html");
        String Nombre = req.getParameter("NOMBRE");
        out.println("<p>Hola Mundo! Bienvenido : "+Nombre+"</p>");
    }
}
```

### - Propiedades

- Gestión de sesiones: Es posible hacer seguimiento de usuarios a través de distintos servlets a través de creación de sesiones.
- Gestión de cookies: Los servlets permiten un manejo fácil de las cookies, datos en texto plano que son guardados en el cliente.
- Muti-hilo: Permiten que cualquier número de clientes a la vez accedan al objeto.
- Rapidez y persistencia: Los servlets sólo cargan una vez por el servidor. Una vez se ha procesado la información devuelta al cliente, éstos no se apagan por lo que mejora la rapidez en las siguientes peticiones.
- Seguridad: Para llamar a un servlet la única forma es a través del servidor web. Por lo tanto, da un altísimo nivel de seguridad. Esto significa que el cliente no puede modificar ni eliminar nada del servidor. También tiene servicio de encriptamiento de mensajes.

### - Usos de los servlets

El mejor uso que se le pueden dar a los servlets es en las que el navegador manda datos para procesar en el servidor, sin ir más lejos el `$_POST` y `$_GET` que hemos

visto en clase son servicios que ofrece HTTP (protocolo de comunicaciones que se emplea para que el cliente envíe peticiones al servidor) y que están implementados por los servlets.

La clase `HttpServlet` incorpora métodos específicos para los servidores web. Su uso más común es la interpretación de formularios HTML. Su funcionalidad se resume en los siguientes puntos:

1. Leer los datos enviados por el navegador.
2. Extraer cualquier información útil incluida en la cabecera HTTP o en el cuerpo del mensaje de petición enviado por el cliente.
3. Generar los resultados de esa información mediante la programación.
4. Mostrar los resultados en un documento HTML.
5. Establecer los parámetros HTTP adecuados incluidos en la cabecera de la respuesta (por ejemplo: el tipo de documento, cookies, etc.)
6. Enviar el documento final al cliente.

### ¿Para qué sirven?

Son útiles para trabajar en entorno de servidor, creando así webs dinámicas y gestionando su dialogo entre el cliente y el servidor. Pero su gran inconveniente es su sintaxis poco entendible para el ser humano.

## ○ JSP (Java server pages)

### - Definición

JSP es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, entre otros tipos de documentos. Digamos que el PHP desarrollado por Java.

Esta vez, JSP está basado en los servlets, de hecho es más fácil programar en JSP ya que es un lenguaje más sencillo. Las páginas se crean parecido que en PHP o ASP, y también hay posibilidad de incrustar código dentro de la estructura HTML. Aquí un ejemplo, que muestra la fecha y la hora de hoy:

```
<HTML>
<HEAD>
<TITLE>JSP Example</TITLE>
</HEAD>
<BODY BGCOLOR="ffffcc">
<CENTER>
<H2>Date and Time</H2>
<%
java.util.Date today = new java.util.Date();
out.println("Today's date is: "+today);
%>
</CENTER>
</BODY>
```

</HTML>

## - Propiedades:

- Es independiente del servidor: JSP se puede ejecutar en los servidores y sistemas operativos más usados, como por ejemplo Apache o Microsoft IIS
- De código abierto.
- Creación y uso de tags: Este lenguaje te permite interactuar con las etiquetas y scripts de la página, aparte que el programador puede crearse sus propias etiquetas y no depender tanto de las ya creadas.
- Al ser de Java, tiene a nuestra disposición todas las librerías y es más entendible por parte de los desarrolladores para, por ejemplo, corregir errores. Además Java es un lenguaje que todo profesional de la programación se ha enfrentado alguna vez.

## - Usos de JSP

Una página JSP es básicamente una página web con HTML tradicional y código Java. La extensión de fichero de una página JSP es ".jsp" en vez de ".html". Al ser de Java, su comportamiento es muy parecido ya que necesita traducirse en servlets.

Una vez traducido, es manejado por el motor Servlet. Éste carga la clase ya mencionada (`javax.servlet.http.HttpServlet`) y lo ejecuta para crear HTML dinámico y enviarlo al navegador.

Todo esto se lleva a cabo normalmente cuando se recibe la primera solicitud de la página .jsp.

También existe la opción de precompilar el código en bytecodes de Java para evitar el tiempo de espera la primera vez que el cliente solicita la página.

Como curiosidad, si el cliente accede al código de la página que le llega sólo verá HTML, sin poder acceder al código JSP.

¿Para qué sirven?

Su base son servlets, sólo que este lenguaje al interpretarlo en el servidor se traducen en servlets para que genere la página en formato HTML. Su fácil comprensión hace que sea más fácil trabajar con ellos y dejar un poco de lado los servlet.

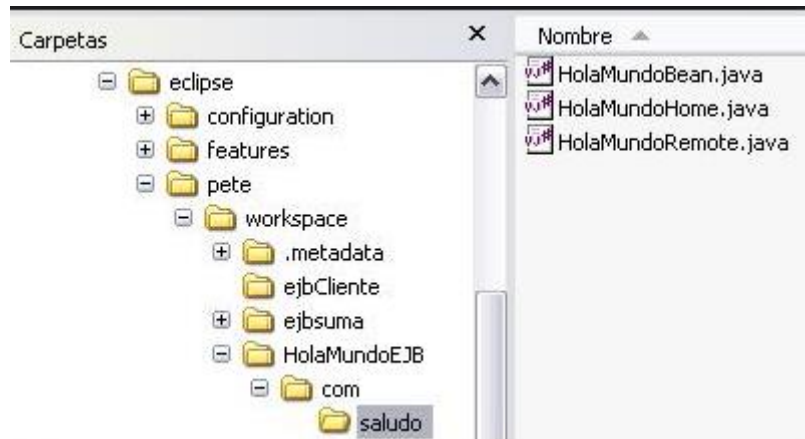
## ○ EJB (Enterprise JavaBeans)

### - Definición:

Primero para que quede claro la definición del EJB es mejor ir por partes y explicar lo que es un JavaBean: También llamado como bean, es un componente hecho en software reutilizable y que puede ser usado en otras aplicaciones en Java. Esto hace fácil la reutilización sin que el programador tenga que entender el código de este.

Un Enterprise JavaBeans es un componente Java Enterprise, y para su ejecución se necesita un contenedor (EJB Container\*) para ser funcional. Aparte que para que funcione requiere de otros componentes. Digamos que es un JavaBean compuesto por otros que le añade más usos que uno sólo.

\*Programa en Java que funciona en el servidor y que contiene todas las clases y objetos necesarios para el correcto funcionamiento. En el ejemplo de a continuación se puede ver una librería en Eclipse con las clases de java en la carpeta HolaMundoEJB.



#### - Propiedades:

- Servicios “Middleware”: Es el intermediario que se puede programar en caso de error, por ejemplo si no fue posible realizar cierto procedimiento, mostrar un mensaje informándolo y mostrando alternativas.
- Ofrece la posibilidad de trabajo repartido: Se reparte el trabajo entre los servicios (el programador concentra su tarea en diseñar el Middleware) sin tener en cuenta lo que su compañero, que está haciendo los componentes. Esa afirmación podría dar lugar a dudas ya que, ¿cómo se sabe si funcionarán bien las partes? Eso se responde con que existen especificaciones de EJB's definen los requerimientos de ambas partes.
- Mejora el rendimiento para el cual está diseñado: Por ejemplo, el EJBContainer de IBM es más eficiente con sus productos aunque también si lo vende a otras empresas funcionaría igual.
- Puede usarse para muchos aspectos, para bases de datos, applets, servlets...
- 

#### - Usos de los EJB:

Los servicios proporcionados que nos puede ofrecer los EJB son:

- Manejo de transacciones: Administra la apertura y el cierre de los métodos del bean.
- Seguridad: Para acceder a un método necesita permisos.
- Concurrencia: Un bean puede ser solicitado por varios clientes a la vez.
- Servicios de red: comunicación entre el cliente y el bean en máquinas distintas.

- Gestión de recursos: Gestión automática de múltiples recursos, como colas de mensajes, bases de datos o fuentes de datos en aplicaciones heredadas que no han sido traducidas a nuevos lenguajes/entornos y siguen usándose.
- Persistencia: Las tablas de las BBDD y los bean se sincronizan.
- Gestión de mensajes: Manejo de Java Message Service (JMS).
- Escalabilidad: Posibilidad de constituir clusters de servidores de aplicaciones con múltiples hosts para poder dar respuesta a aumentos repentinos de carga de la aplicación con sólo añadir hosts adicionales.
- Adaptación en tiempo de despliegue: Posibilidad de modificación de todas estas características en el momento del despliegue del bean.

¿Para qué sirven?

Su función principal parece ser la de servir a las empresas en su lógica de negocio. También favorece la reutilización de código al tener disponibles ciertos beans ya hechos y sin tener que estar desarrollados en los mismos lenguajes que el que se está siendo usado actualmente.

## ○ Conclusiones:

¿Es indiferente que se use uno en vez del otro? ¿Qué característica diferenciadora tiene cada uno de ellos?

La respuesta a la primera pregunta es NO. Todo buen programador siempre busca la eficiencia de su programa y los tres elementos vistos anteriormente cada uno tiene sus características que los hace mejor que el otro y viceversa.

Por ejemplo, en mi opinión los servlets no hacen tanta falta en el desarrollo de páginas web ya que son complicados de entender y ya los genera el jsp por eso es mejor programar web en JSP, ya que como hemos dicho antes éste lenguaje se traduce en servlets.

Los ejb son más usados para grandes empresas, que no les hagan falta desarrollar software “a mano” y se usan mejor en conjunto con otros beans, aunque por el contrario, en jsp pueden funcionar perfectamente solo un archivo html, algo parecido a JavaScript.

## ○ Bibliografía

- Servlets:

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

<http://losteatinos.com/servlets/servlet.html>

<https://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/jbarrios/general.html>

<http://wikis.uca.es/wikiii/index.php/Servlets>

<http://www.lab.inf.uc3m.es/~a0080802/RAI/servlet.html>

- JSP

La definición basada en Wikipedia.

<http://www.desarrolloweb.com/articulos/831.php>

[http://programacion.net/articulo/desarrollo\\_de\\_aplicaciones\\_web\\_con\\_jsp\\_y\\_xml\\_120](http://programacion.net/articulo/desarrollo_de_aplicaciones_web_con_jsp_y_xml_120)

- EJB

<https://www.osmosislatina.com/java/ejb.html>

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