

# Servlets, JSP y EJB

## *Servlets:*

### HISTORIA

El término “Servlet” deriva de uno anterior, “applet”, referido a pequeños programas que se ejecutan en el contexto de un navegador web. Sin embargo, un “Servlet” es un programa que se ejecuta en un servidor. El término “Servlet” fue creada por Sun Microsystems (1997).

### QUE SON Y PARA QUE SE UTILIZAN

El “Servlet” es una clase en el lenguaje de programación Java, que se utiliza para mejorar las capacidades de un servidor. Es un objeto que se ejecuta en un servidor o contenedor JEE, especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente HTML.

Se utilizan comúnmente para extender las aplicaciones alojadas por servidores web, de manera que pueden ser vistos como “applets” de Java que se ejecutan en servidores en vez de navegadores web.

Este tipo de “Servlets” son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.

### CARACTERÍSTICAS

Los “Servlet” son objetos diseñados principalmente para ofrecer contenido dinámico desde un servidor web, normalmente HTML.

Un “Servlet” implementa la interfaz `java.servlet.Servlet` o hereda alguna de sus clases. Al implementar esta interfaz, el “Servlet” es capaz de interpretar objetos de tipo “`HttpServletRequest`” y “`HttpServletResponse`”, los cuales contienen la información de la página que invoca al “Servlet”.

Es necesario señalar que los “Servlet” son del tipo HTTP por lo que se extiende de la clase `HttpServlet`. Podemos destacar las siguientes propiedades de los “Servlet”:

- **Manejo de Sesiones:** Se puede hacer seguimiento de usuarios a través de distintos “Servlet” a través de la creación de sesiones.
- **Utilización de Cookies:** Las cookies son pequeños datos en texto plano que pueden ser guardados en el cliente. La API su manejo.
- **Multi-thread:** Los “Servlet” soportan el acceso concurrente de los clientes.
- **Programación en Java:** Se obtienen las características de multiplataforma o acceso a APIs como JDBC, RMI, etc.

# *Java Server Pages (JSP):*

## **HISTORIA**

Tras la revolución de internet, hubo problemas relacionados a lo estático del modelo inicial que presentaba el uso de **HTML** en las páginas. Mediante el uso de la interfaz de **CGI** (Common Gateway Interface), los desarrolladores pudieron acceder a un mundo nuevo, en el cual podían crear aplicaciones del lado del servidor, las cuales eran accedidas a través del navegador.

El gran problema que surgió en cuanto a que la performance de los servidores se veía gravemente afectada por generar un desaprovechamiento de los recursos, ya que dos usuarios accediendo al mismo recurso utilizaban dos procesos distintos.

Microsoft lanzó en el año 1997 una nueva propuesta para resolver esta situación. La propuesta se basaba en la creación dinámica de código **HTML** a través de un concepto llamado **ASP** (Active Server Pages).

Surgieron otras opciones de la mano de JAVA con los “Servlet”, los scripts del lado del cliente eran populares (JScript y VBScript), pero brindaban una solución de un lado solo, **JSP** supuso la unión entre **Java** y **HTML**.

## **QUE SON Y PARA QUE SE UTILIZAN**

**JSP** (Java Server Pages) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en **HTML** y **XML**. El **JSP** es similar al **PHP**, pero usa el lenguaje de programación Java. Son un método de creación de páginas web dinámicas en servidor usando el lenguaje Java.

Para desplegar y correr **JSP** se requiere un servidor web compatible con contenedores “Servlet” como **Apache** Tomcat o Jetty.

**JSP** puede ser visto como una abstracción de alto nivel de los “Servlet” Java. Las **JSP** son traducidas a “Servlet” en tiempo real, que son guardados en caché y reusado hasta que la **JSP** original es modificada.

## **CARACTERISTICAS**

El rendimiento de una página **JSP** es el mismo que tendría el “Servlet” equivalente, ya que el código es compilado como cualquier otra clase **Java**. Sin embargo **JSP** tiene un buen desempeño y eso lo más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

La principal ventaja de **JSP** frente a otros lenguajes es que **Java** es un lenguaje de propósito general, lo que permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento **HTML** en el archivo **JSP**.

Otra ventaja es que **JSP** hereda la portabilidad de **Java**, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios.

Los **JSP** se ejecutan en una máquina virtual **Java**, permitiendo que se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada **JSP** se ejecuta en su propio hilo, pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo.

Las **JSP** son en realidad una forma alternativa de crear “Servlet” ya que el código **JSP** se traduce a código de “Servlet” **Java** la primera vez que se le invoca y en adelante es el código del nuevo “Servlet” el que se ejecuta produciendo como salida el código **HTML** que compone la página web de respuesta.

# ***Enterprise Java Beans (EJB):***

## **HISTORIA**

Fue presentada por **IBM** y **Sun Microsystems**, de manera que fueron rápidamente adoptados por grandes empresas. Sin embargo, aparecieron problemas y la fama de **EJB** comenzó a deteriorarse. Finalmente muchos programadores encontraron que las **APIs** eran demasiado complicadas, esparciendo así la percepción que los **EJB** introducían complejidad sin aportando beneficios reales.

La aparición de **EJB 2.0** añadió el concepto de interfaces locales que podían ser llamadas directamente sin penalizar el rendimiento para aplicaciones que no eran distribuidas por varios servidores.

Los **EJB 3.0** usan fuertemente las anotaciones (meta datos), una funcionalidad añadida al lenguaje Java en su versión 5.0, para habilitar un estilo de programación mucho menos farragoso.

## **QUE SON Y PARA QUE SE UTILIZAN**

Las **Enterprise JavaBeans (EJB)** son una de las interfaces de programación de aplicaciones (**API**) que forman parte del estándar de construcción de aplicaciones empresariales **JEE** de **Oracle Corporation**.

Los **EJB** proporcionan un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los **EJB** es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial y así poder centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

No hay que confundir los **Enterprise JavaBeans** con los **JavaBeans**. Los **JavaBeans** también son un modelo de componentes creado por **Oracle** para la construcción de aplicaciones, pero no pueden utilizarse en entornos de objetos distribuidos al no soportar nativamente la invocación remota (**RMI**).

## **CARACTERISTICAS**

Una característica importante de **EJB**, es que los componentes permiten reusar tanto código como datos, ya que se despliegan en un servidor. Las tres principales ventajas del desarrollo basado en componentes, son: **La Reusabilidad, la Modularidad y la Interoperabilidad.**

Estos componentes se caracterizan por la permisión de desarrollo de aplicaciones débilmente acopladas, y su comportamiento especificado por interfaces, pudiéndose configurar en tiempo de despliegue de forma declarativa.

Aun así podemos considerar como características esenciales de **EJB** las siguientes:

- Contiene la lógica del negocio que opera con **EIS** (Enterprise Information System).
- Las instancias son creadas y manejadas por el container **EJB**.
- Puede ser configurado y editados sus parámetros de entorno via archivos **XML**.
- La seguridad y transacciones se encuentran separadas de las clases **EJB**, lo que permite la operación de aplicaciones externas.

# Bibliografía

- [EcuRed – Servlets](#)
- [Java Servlet - Wikipedia](#)
- [Servlet-OracleAPI](#)
- [Java Servlet](#)
- [JSP](#)
- [JSP - Wikipedia](#)
- [EJB-Wikipedia](#)
- [EJB](#)
- [ArquitecturaJava](#)