Servlets, JSP y EJB

# Servlet

## ¿Qué es un Servlet de Java?

Los servlets son módulos escritos en Java que se utilizan en un servidor, que puede ser o no un servidor web, para extender sus capacidades de respuesta a los clientes al utilizar las potencialidades de Java.

Los servelts pueden ser incluidos en servidores que soporten la API de Servlet. La API no realiza suposiciones sobre el entorno que se utiliza, como tipo de servidor o plataforma, ni del protocolo a utilizar, aunque existe una API especial para HTTP.

Los servlets son un reemplazo efectivo para los CGI ya que proporcionan una forma de generar documentos dinámicos utilizando las ventajas de la programación en Java como conexión a alguna base de datos, manejo de peticiones concurrentes, programación distribuida, etc. Por ejemplo, un servlet podría ser responsable de procesar los datos desde un formulario en HTML como registrar la transacción, actualizar una base de datos, contactar algún sistema remoto y retornar un documento dinámico o redirigir a otro servlet u alguna otra cosa.

## Características de un servlet

* **Son independientes** del servidor utilizado y de su sistema operativo
* **Consumen pocos recursos** ya que solo son cargados la primera vez que se solicitan sus servicios (las siguientes peticiones crearan hilos de ejecución)
* **Seguros y portables** al ser ejecutados bajo la misma máquina virtual de java
* **No requieren soporte para java** en el explorador del cliente, ya que operan en el dominio en el servidor y envían los resultados en HTML
* **Manejo de Sesiones**: Permite realizar un seguimiento de usuarios a través de distintos servlets mediante la creación de sesiones.
* **Utilización de Cookies**: Las cookies son pequeños datos en texto plano que pueden ser guardados en el cliente. La API de servlets permite un manejo fácil y limpio de ellas.
* **Multi-thread**: Los servlets soportan el acceso concurrente de los clientes, aunque hay que tener especial cuidado con las variables compartidas a menos que se utilize la interfaz SingleThreadModel.
* **Programación en Java**: Se obtienen las características de multiplataforma o acceso a APIs como JDBC, RMI, etc.
* **Los servlets pueden llamar a otros servlets**, e incluso a métodos concretos de otros servlets (en la misma máquina o en una máquina remota). De esta forma se puede distribuir de forma más eficiente el trabajo a realizar.
* **Los servlets pueden obtener fácilmente información** acerca del cliente (la permitida por el protocolo HTTP), tal como su dirección IP, el puerto que se utiliza en la llamada, el método utilizado (GET, POST), etc
* **Pueden actuar como enlace** entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor.
* **Pueden realizar tareas de proxy para un applet**. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero sí podría hacerlo a través de un servlet.
* **Permiten la generación dinámica de código HTML**, lo que se puede utilizar para la creación de contadores, banners, etc.

# JSP

## ¿Qué es JSP?

Es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas externas e incluso personalizadas.

Con JSP, los diseñadores web pueden incorporar elementos dinámicos dentro de la página utilizando tanto porciones de código Java incrustadas, como unas cuantas etiquetas. De esta forma, cuando una página es solicitada por un usuario y procesada por un servidor HTTP, el código HTML pasará directamente al usuario, mientras que las porciones de código Java serán ejecutadas en el servidor cuando la solicitud haya sido recibida, para generar el contenido dinámico de la página. Cuando el usuario acceda al código de la página que le llega sólo verá HTML, sin poder acceder al código JSP subyacente.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

## Características JSP

* Utilización de procesos ligeros (hilos java) para el manejo de la peticiones
* Manejo de múltiples peticiones sobre una página .jsp en un instante dado
* El contenedor servlet puede ser ejecutado como parte del servidor web
* Facilidad para compartir recursos entre peticiones (hilos con el mismo padre: servlet container)
* Creación, utilización y modificaciones de JavaBeans del servidor.
* Los JavaBeans utilizados en páginas .jsp pueden ser utilizados en servlets, applets o aplicaciones Java.
* Cambios realizados en el código HTML relativos a cómo son mostrados los datos, no interfieren en la lógica de programación y viceversa.
* Los diseñadores de páginas pueden centrarse en el código HTML y los programadores en la lógica del programa.
* Los desarrollos pueden hacerse independientemente.
* Las frecuentes modificaciones de una página se realizan más eficientemente.
* Conjunta el poder de Java en el servidor y la flexibilidad de HTML en el navegador.
* No sólo se puede utilizar HTML, sino también XML o WML.
* Capaz de instanciar cualquier clase de Java

# EJB

## ¿Qué es?

Es una arquitectura de componentes de servidor que simplifica el proceso de construcción de aplicaciones de componentes empresariales distribuidos en Java.

Su uso permite realizar aplicaciones escalables, fiables y seguras sin escribir código de infraestructura.

Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor. Este modelo permite al programador abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, etc.) para 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.

Existen tres tipos de EJBs:

* **EJB de Entidad**: encapsulan los objetos del lado del servidor que almacena los datos. Los EJB de entidad presentan la característica fundamental de la persistencia:
  + **Persistencia gestionada por el contenedor (CMP)**: el contenedor se encarga de almacenar y recuperar los datos del objeto de entidad mediante el mapeo de una tabla de la base de datos.
  + **Persistencia gestionada por el bean (BMP)**: el propio objeto entidad se encarga, mediante una base de datos u otro mecanismo, de almacenar y recuperar los datos.
* **EJB de Sesión**: gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:
  + **Con estado**. El contenedor garantiza el mantenimiento del estado del componente entre distintas invocaciones.
  + **Sin estado**. Los beans de sesión carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. No se garantiza que los contenidos de las variables de instancia se conserven entre llamadas al método.
* **EJB dirigidos por mensajes**: son los únicos beans con funcionamiento asíncrono. Usando el Java Messaging System (JMS), se suscriben a un tema o a una cola y se activan al recibir un mensaje dirigido a dicho tema o cola. No requieren de su instanciación por parte del cliente.

## Características

* **Manejo de transacciones**: apertura y cierre de transacciones asociadas a las llamadas a los métodos del bean.
* **Seguridad**: comprobación de permisos de acceso a los métodos del bean.
* **Concurrencia**: llamada simultánea a un mismo bean desde múltiples clientes.
* **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 en la empresa.
* **Persistencia**: sincronización entre los datos del bean y tablas de una base de datos.
* **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.

# Diferencias entre ellos

## Diferencia entre Servlet y JSP

* Los servlets son extensiones del servidor y proveen una funcionalidad extra al servidor web, son utilizados como controladores de componentes, en cambio JSP son páginas web, con una estructura similar a las páginas HTML. Una aplicación web consiste en una combinación de servlets y JSP’s:
  + JSP: se utilizan para la presentación visual.
  + Servlet: controlan la lógica de negocio.
* JSP es un lenguaje de scripting web que pueden generar contenido dinámico mientras servlets son programas Java que ya están compilados que también crea contenido web dinámico.
* Los servlets se ejecutan más rápido en comparación con JSP.
* JSP puede ser compilado en Java Servlets.
* Es más fácil el código JSP que en Java.

## Diferencia entre EJB y Servlet

* Los EJBs no pueden aceptar peticiones HTTP. En otras palabras, no pueden servir peticiones que vienen directamente desde un navegador Web, y los servlets si pueden.
* Servlets y JSPs se pueden usar para implementar presentación y control web, pero al contrario que los EJBs, no pueden manejar transacciones distribuidas. Los EJBs pueden ser llamados desde cualquier cliente basado en Java.

# Cuándo usar Servlets

Se utilizan para generar páginas HTML para cada una de las siguientes peticiones:

* Página Web basada en datos enviados por el cliente
  + Motores de búsqueda, confirmación de pedidos
* Página Web derivada de datos que cambian con frecuencia
  + Informe del tiempo o noticias de última hora
* Página Web que usa información de bases de datos corporativas u otras fuentes de la parte del servidor
  + Comercio electrónico: precios y disponibilidades

Como JSP es en realidad una forma alternativa de crear servlets se utiliza para el mismo tipo de aplicaciones web que los servlets.

# Cuándo usar EJBs

Los EJBs son buenos para las aplicaciones que tienen alguno de estos requerimientos:

* **Escalabilidad**: si tenemos un número creciente de usuarios, los EJBs nos permitirán distribuir los componentes de nuestra aplicación entre varias máquinas con su localización transparente para los usuarios.
* **Integridad de Datos**: los EJBs nos facilitan el uso de transacciones distribuidas.
* **Variedad de clientes**: si nuestra aplicación va a ser accedida por una variedad de clientes, se pueden usar los EJBs para almacenar el modelo del negocio, y se puede usar una variedad de clientes para acceder a la misma información.

# Bibliografía

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

<https://users.dcc.uchile.cl/~jbarrios/servlets/general.html>

<https://prezi.com/ojbrawecgrs0/descripcion-de-las-caracteristicas-basicas-del-uso-de-servlets-y-jsp-para-html-del-lado-del-cliente/>

<http://www.diferenciaentre.net/jsp-vs-servlet/>

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

<https://prezi.com/p0fjnovlwx9q/descripcion-de-las-caracteristicas-principales-de-las-siguie/>

<http://di002.edv.uniovi.es/~dflanvin/docencia/dasdi/teoria/Transparencias/13.%20Introducci%C3%B3n%20EJBs.pdf>

<http://javagratis.net63.net/programacion-java/ejb/61-ique-son-los-ejb.html>

<http://www.inf-cr.uclm.es/www/mpolo/asig/0708/tutorJavaWeb.pdf>

<http://onj2ee.blogspot.com.es/2009/12/conceptos-basicos-servlets-y-jsp.html>

<https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0ahUKEwiP0NCauPvPAhWLXBoKHYzUDGIQFghBMAY&url=http%3A%2F%2Fprofesores.elo.utfsm.cl%2F~agv%2Felo329%2F1s05%2Fprojects%2FValdes.Aracena%2FpresentacionDpoo.ppt&usg=AFQjCNEAKAr8pWgR_H0u8mUglW-FpHsCEg&cad=rja>

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