

2.6.- JSP.

2.6.1. Introducción.

JSP es una tecnología Java para el desarrollo de aplicaciones Web basada en los servlet. Se puede decir que los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java.

La principal diferencia entre servlet y JSP es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java, mientras que un servlet es un programa que recibe peticiones y, a partir de ellas, genera una página web. Los JSP son mejores para generar páginas con gran parte de contenido estático y los servlets en tareas que generen poca salida, datos binarios o páginas con gran parte de contenido variable.

Una página JSP es simplemente una página web con “elementos JSP”. Todo en la página no son elementos JSP, el resto se denomina “plantilla de texto”.

Esta plantilla de texto puede ser cualquier tipo de texto: HTML, WML, XML. Esta anatomía JSP se muestra en la siguiente ilustración:

```
<%@ page language="java" contentType="text/html" %>
<html>
<body bgcolor="white">

<jsp:useBean
  id="userInfo"
  class="com.ora.jsp.beans.userInfo.UserInfoBean">
<jsp:setProperty name="userInfo" property="*" />
</jsp:useBean>

The following information was saved:
<ul>
<li>User Name:

<jsp:getProperty name="userInfo"
  property="userName" />

<li>Email Address:

<jsp:getProperty name="userInfo"
  property="emailAddr" />

</ul>
</body>
</html>
```

Ilustración 23.- Anatomía de una página JSP.

Para procesar páginas JSP es necesario el uso de un contenedor JSP (JSP Container).

Se puede observar que JSP es una tecnología que permite incluir código Java en páginas web, y el contenedor JSP es el encargado de tomar la página, sustituir el código Java que contiene por el resultado de su ejecución y enviarla al cliente. De esta forma se consigue una web dinámica.

2.6.2. Procesamiento de una página JSP.

El contenedor JSP se encarga de interceptar las peticiones de la página JSP. El ejecutar todos los elementos en la página requiere antes de la conversión de la página JSP en servlet. Esta conversión traduce todas las plantillas de texto a declaraciones similares a `println()` y todos los elementos JSP a código Java. Y posteriormente, el contenedor lo compila dando lugar una clase que se ejecuta como un servlet. Todo este procedimiento de conversión de la página JSP y la compilación se denomina “fase de traducción” y tiene lugar únicamente la primera vez que se inicia una solicitud a la página JSP, posteriormente no es necesaria porque ya se dispone del servlet.^{[18] [19]}

Una vez se completa esta fase, el contenedor JSP es también responsable de invocar la clase generada, es decir, el servlet que se ha generado al compilar el código java en la fase de traducción. De esta forma puede procesar cada solicitud y generar la respuesta. Esta es una segunda fase denominada “fase de solicitud de transformación”. Se puede seguir el proceso de estas dos fases en la ilustración 24:

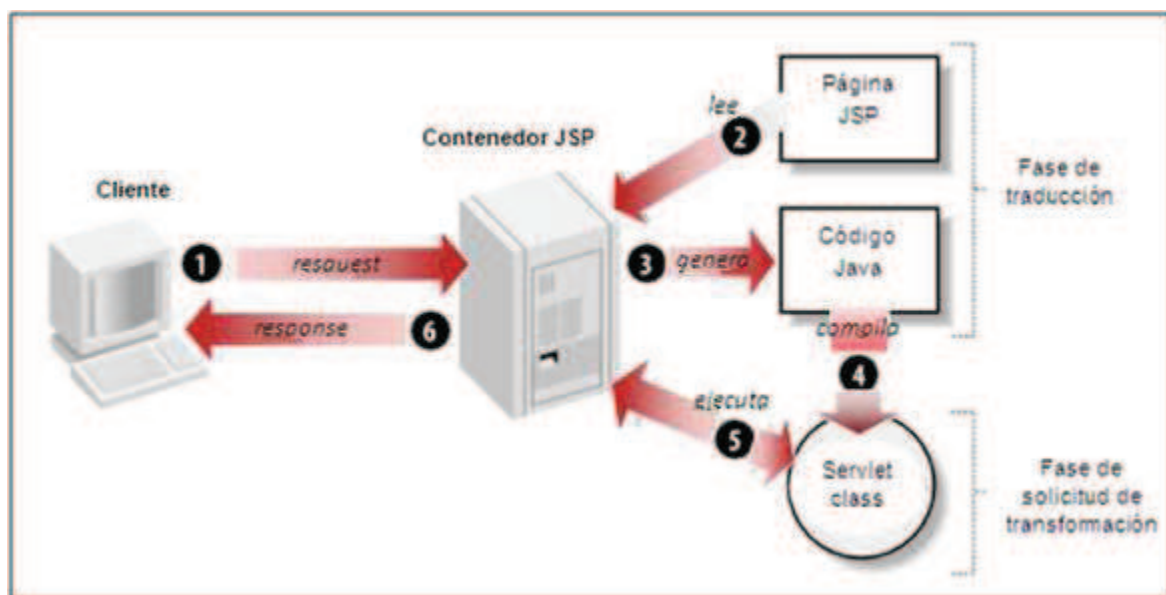


Ilustración 24.- Fases del procesamiento de una página JSP.

A excepción de la fase de traducción, una página JSP se maneja exactamente igual que un servlet: es cargada una vez y llamada en repetidas ocasiones, hasta que el servidor se apague.

En la Ilustración 25 se muestra el proceso de ejecución de un servlet donde se muestra el mismo funcionamiento que se utiliza en este proyecto:

- 1) El proceso comienza con la petición HTTP que llega en primer lugar al servidor web.

- 2) La petición se traslada al Contenedor de aplicaciones, en concreto a su motor del servicio Servlet/JSP (con su propia JVM).
- 3) El motor encapsula la información de la petición en un objeto del tipo `HttpServletRequest`, además encapsula en un objeto `HttpServletResponse` el flujo de respuesta.
- 4) El motor crea por cada petición un hilo, sobre el que se invoca a la función `service()` del servlet. En función del método de la petición (POST o GET), `service()` llamará al método correspondiente del servlet: `doPost()` o `doGet()`, pasándoles los objetos de `HttpServletRequest` y `HttpServletResponse`.
- 5) Cada clase del tipo servlet, tiene una única instancia, sobre la que corren los diferentes hilos (peticiones).

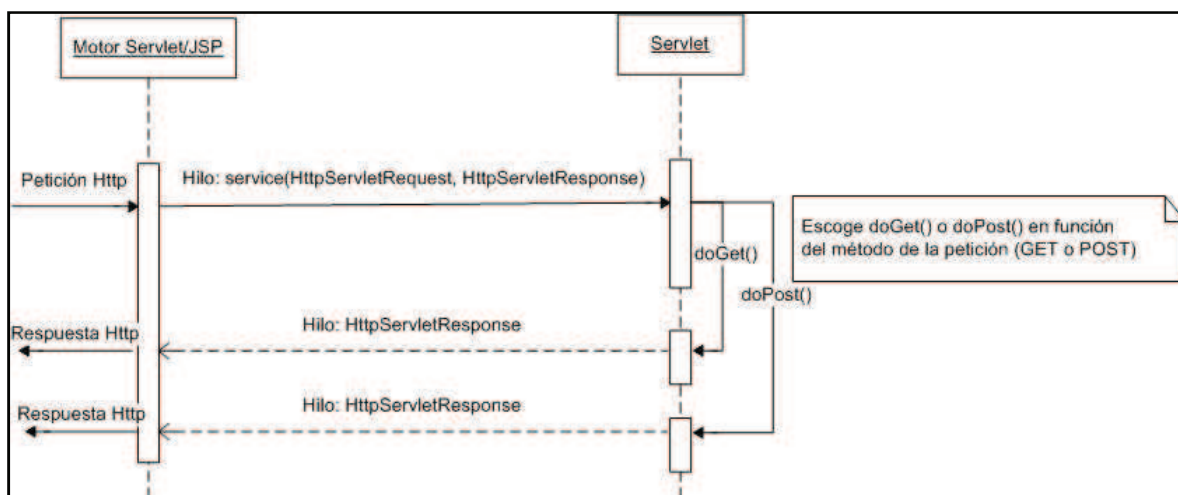


Ilustración 25.- Proceso de ejecución de un servlet.

Cada petición genera un hilo independiente. Pero sólo la primera petición genera una instancia de la clase servlet. Podremos tener N peticiones al servlet, por tanto N hilos, pero únicamente una instancia de la clase (un objeto).

2.6.3. Elementos de Script JSP.

ELEMENTOS DE SCRIPT JSP		
Nombre	Elemento	Descripción
Expresiones	<code><%= ... %></code>	Expresión en código Java cuyos valores se insertan directamente en la salida. La expresión Java es evaluada, convertida a string e insertada en la página.
Scriptlets	<code><% ... %></code>	Conjunto de instrucciones Java que se insertan tal cual dentro del método servlet que será construido al generar la página
Declaraciones	<code><%! ... %></code>	Código Java que permite definir variables de tipo simple, objetos y métodos que serán insertados dentro del cuerpo principal de la clase servlet.

Tabla 40.- Elementos de Script JSP.

Los elementos de script JSP permiten insertar trozos de código, normalmente Java, en una página JSP. Estos elementos son los que se muestran en la Tabla 40.

2.6.4. Directivas JSP.

Las directivas JSP son instrucciones que especifican información sobre la página en sí. Existen tres tipos de directivas: *page*, que permite importar clases, personalizar la superclase del servlet, etc.; *include*, que permite insertar un fichero dentro de la clase servlet en el momento que el fichero JSP es traducido a un servlet; y, *taglib* que permite que los autores de JSP definan sus propias etiquetas. Se identifican porque van entre las marcas:

<%@ ... %>

Las directivas JSP se muestran a continuación:

DIRECTIVAS JSP		
Nombre	Elemento	Descripción
<i>page</i>	<%@page ... %>	Define los atributos que dependen de la página:
	<%@page language = "...">	Especifica el lenguaje a utilizar.
	<%@page extends = "..."%>	Especifica la superclase del servlet que se va a generar.
	<%@page import = "..."%>	Importa los paquetes y clases Java para ser utilizados dentro del fichero JSP.
	<%@page session = "..."%>	Un valor de <i>true</i> indica que la variable <i>session</i> debería unirse a la sesión existente, si no existiera se debería crear una nueva sesión. Un valor de <i>false</i> indica que no se usarán sesiones, y los intentos de acceder a la variable <i>session</i> resultarán en errores.
	<%@page isThreadSafe = "..."%>	El valor <i>true</i> indica que múltiples peticiones pueden procesarse con un solo servlet. El valor <i>false</i> indica que habrá una instancia por cada petición.
	<%@page errorPage = "..."%>	Especifica la ruta de la página de error que será invocada en caso de producirse una excepción durante la ejecución de este fichero JSP.
	<%@page isErrorPage = "..."%>	Indica si la página actual actúa o no como página de error de otras páginas JSP.
	<@page contentType = "...">	Especifica el tipo <i>MIME</i> de la salida. El valor por defecto es <i>text/html</i> .
	<@page buffer = "...">	Especifica el tamaño del buffer de salida. Por defecto es 8Kb.
<i>include</i>	<@page autoFlush = "...">	El valor <i>true</i> indica que el buffer debería descargarse cuando esté lleno. El valor <i>false</i> indica que se debe lanzar una excepción cuando el buffer se sobrecargue.
	<@page info = "...">	Define un string que puede usarse para ser recuperado mediante el método <i>getServletInfo</i> .
<i>taglib</i>	<@page pageEncoding = "...">	Define el juego de caracteres que usa la página
<i>include</i>	<%@include ... %>	Incluye código en la página JSP en la fase de traducción.
<i>taglib</i>	<%@taglib ... %>	Permite el uso de etiquetas definidas por el usuario.

Tabla 41.- Directivas JSP.

2.6.5. Acciones JSP.

Las acciones JSP realizan algún tipo de acción basada en la información que es requerida en la solicitud de la página JSP. Las acciones disponibles son:

ACCIONES JSP	
Nombre	Descripción
<code><jsp:useBean></code>	<code><jsp:useBean id="..." class="..." scope="..." type="..." beanName="..." /></code> Permite cargar y utilizar un JavaBean en la página JSP. Sus atributos son: - <i>id</i>: nombre asignado a la variable que referencia al bean. - <i>class</i>: designa el nombre completo del paquete del bean. - <i>scope</i>: indica el contexto en el que el bean estará disponible. Valores posibles: "page request session application". (opcional) - <i>type</i>: especifica el tipo de variable a la que se referirá el objeto. (opcional) - <i>beanName</i>: asigna el nombre del bean. (opcional)
<code><jsp:getProperty></code>	<code><jsp:getProperty name = "..." property = "..." /></code> Recupera el valor de una propiedad del bean, lo convierte a un string, e inserta el valor en la salida. Sus atributos: - <i>name</i>: nombre del bean. - <i>property</i>: propiedad cuyo valor se desea obtener.
<code><jsp:setProperty></code>	<code><jsp:setProperty name="..." property="..." value = "..." param="..." /></code> Obtiene los valores de las propiedades (atributos) de los beans que se han referenciado anteriormente y los modifica. Sus atributos son: - <i>name</i>: designa el bean cuya propiedad va a ser seleccionada. - <i>property</i>: indica la propiedad que se quiere seleccionar. - <i>value</i>: especifica el valor que se quiere establecer en dicha propiedad. (opcional) - <i>param</i>: parámetro de la petición del que se debería derivar la propiedad. (opcional)
<code><jsp:include></code>	<code><jsp:include page="..." flush="..." /></code> Inserta el contenido generado por otra página (<i>page</i>) dentro de la salida de la página actual, en el lugar donde aparece la acción <code><jsp:include ... ></code> . Al contrario que la directiva <i>include</i> , que inserta el fichero en el momento de la conversión de la página JSP a un Servlet, esta acción inserta el fichero en el momento en que la página es solicitada.
<code><jsp:forward></code>	<code><jsp:forward page="..." /></code> Permite reenviar la petición a otra página. Tiene un sólo atributo, <i>page</i> , que es una URL relativa.
<code><jsp:param></code>	<code><jsp:param></code> Modificar la petición HTTP para incluir un nuevo parámetro (y su valor) que no venían en la petición original.
<code><jsp:plugin></code>	<code><jsp:plugin attribute="value"*/></code> Permite insertar un elemento OBJECT o EMBED del navegador para especificar que el navegador debería ejecutar un applet usando el Plug-in Java.
	<code><jsp:plugin attribute="value"*/></code> Se usa para la configuración de applets.

Tabla 42.- Acciones JSP.

2.6.6. Variables Predefinidas JSP.

Las variables predefinidas simplifican el código en expresiones y scriptlets JSP, ya que son objetos ya establecidos los cuales no hay que instanciar. Existen ocho variables definidas, algunas veces llamadas "*objetos implícitos*":

VARIABLES PREDEFINIDAS JSP	
Nombre	Descripción
<i>request</i>	Asociado a la petición del cliente, del tipo <i>HttpServletRequest</i> .
<i>response</i>	Asociado a la respuesta, del tipo <i>HttpServletResponse</i> .
<i>pageContext</i>	Representa el objeto <i>PageContext</i> para encapsular características de uso específicas del servidor.
<i>session</i>	Representa el objeto <i>HttpSession</i> asociado con la petición actual.
<i>application</i>	Representa el contexto del servlet, es del tipo <i>ServletContext</i> .
<i>out</i>	Representa el objeto de salida de texto por pantalla.
<i>config</i>	Representa el objeto del tipo <i>ServletConfig</i> para dicha página.
<i>page</i>	Referencia a la propia página. Se usa como alternativa al objeto <i>this</i> .

Tabla 43.- Variables Predefinidas JSP.

2.6.7. Convenciones de comentarios y Caracteres de escape.

Hay un pequeño número de construcciones especiales que se pueden usar en varios casos para insertar comentarios o caracteres que de otra forma serían tratados especialmente. Algunos de ellos son:

CONVENCIONES DE COMENTARIOS Y CARACTERES DE ESCAPE	
Sintaxis	Descripción
<code><%-- comment --%></code>	Comentario JSP. Ignorado por el traductor. Este comentario no es enviado al cliente
<code><!-- comment --></code>	Comentario HTML. Se pasa al HTML resultante. Este comentario sí es enviado al cliente.
<code><% //... %></code> <code><% /**... */ %></code>	Comentario Java. Este tipo de comentario no es enviado al cliente.
<code>\'</code>	Representa una comilla simple en un atributo.
<code>\"</code>	Representa una comilla doble en un atributo.

Tabla 44.- Convenciones de comentarios y Caracteres de escape JSP.

2.6.8. Sumario de Sintaxis.

Elemento JSP	Sintaxis	Interpretación
Expresión JSP	<code><%= expression %>;</code>	La Expresión es evaluada y situada en la salida.
Scriptlet JSP	<code><% code %>;</code>	El código se inserta en el método <i>service</i> .
Declaración JSP	<code><%! code %></code>	El código se inserta en el cuerpo de la clase del servlet, fuera del método <i>service</i> .
Directiva <i>page</i> JSP	<code><%@ page att="val" %></code>	Dirige al motor servlet sobre la configuración general.
Directiva <i>include</i> JSP	<code><%@ include file="url" %></code>	Un fichero del sistema local se incluirá cuando la página se traduzca a un Servlet.
Comentario JSP	<code><%-- comment --%></code>	Comentario ignorado cuando se traduce la página JSP en un servlet.
Acción <i>jsp:include</i>	<code><jsp:include page="relative URL" flush="true"/></code>	Incluye un fichero en el momento en que la página es solicitada.
Acción <i>jsp:useBean</i>	<code><jsp:useBean&nbsp;att=val*/></code> <code><jsp:useBean&nbsp;att=val*></code> <code>... </jsp:useBean></code>	Encuentra o construye un Java Bean.
Acción <i>jsp:setProperty</i>	<code><jsp:setProperty att=val*/></code>	Selecciona las propiedades del bean, bien directamente o designando el valor que viene desde un parámetro de la petición.
Acción <i>jsp:getProperty</i>	<code><jsp:getProperty name="propertyName" value="val"/></code>	Recupera y saca las propiedades del Bean.
Acción	<code><jsp:forward page="relative</code>	Reenvía la petición a otra página.

Elemento JSP	Sintaxis	Interpretación
jsp:forward	URL"/>	
Acción jsp:plugin	<jsp:plugin attribute="value"*> ... </jsp:plugin>	Genera etiquetas OBJECT o EMBED, apropiadas al tipo de navegador, pidiendo que se ejecute un applet usando el Java Plugin.

Tabla 45.- Sumario de sintaxis JSP.