

JAVA SERVER PAGES TECHNOLOGY (JSP)

Introducción

Definición

JSP significa Java Server Pages.

Es la propuesta más amigable de la empresa Oracle para programación Web.

Es un lenguaje sencillo y altamente productivo.

Se complementa con el uso de clases de negocios y servlets.

Se utiliza como la capa de presentación de una aplicación Web.

Relación con Servlets

Los servlets fueron la primer propuesta de Sun Microsystems (ahora Oracle) para programación Web.

Los servlets podían responder con HTML pero resultaba muy incómodo.

En java 1.2 surgen los JSP y pueden trabajar en conjunto con los servlets.

Traducción y compilación

Los archivos JSP se convierten en clases (servlets).

El Servlet Container se encarga de realizar la conversión del JSP a una clase, y su posterior compilación.

JSP Scripts

Scriptlets

Los bloques de código encerrados por `<% %>` se conocen como scriptlets.

El código JSP se construye a través de scriptlets.

Las páginas que contiene este tipo de código son denominadas "páginas dinámicas", ya que su contenido tiende a cambiar cada vez que se la llama.

Por ejemplo:

```
[code]
<HTML>
<BODY>
  Hola mundo!La fecha y hora es <%= new java.util.Date() %>
</BODY>
</HTML>
[/code]
```

Declaraciones

Es posible escribir los métodos que formarán parte explícitamente del Servlet que se construirá a partir del archivo JSP.

Ocurre lo mismo con los atributos.

Dichos métodos pueden verse también como "funciones" utilizables dentro del archivo JSP.

Para construir dichos métodos y atributos se utiliza el pseudo-tag `<%!`

Por ejemplo:

```
[code]
```

```

<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%!

    // Atributo
    Date fechaActual = new Date();

    // Método
    Date obtenerFechaActual()
    {
        return fechaActual;
    }

%>
La fecha actual es <%= obtenerFechaActual() %>
</BODY>
</HTML>
[/code]

```

Directivas de JSP

Definición

La directiva es código en forma de tags que afecta a la estructura general del servlet a construir.

En general es código que impacta directamente en el funcionamiento, con lo cual se requiere poner al principio.

Las directivas de JSP comienzan con <%@

Tienen la siguiente forma de uso:

```
<%@ directiva atributo="valor" %>
```

Las directivas que existen son page, include y taglib

Directiva page

Entre sus atributos más conocidos se encuentran: import, pageEncoding, isErrorPage, errorPage, contentType, buffer, autoFlush, isThreadSafe, session, etc.

La más comúnmente utilizada es la "import"

La forma de realizar una importación es la siguiente:

```
<%@ page import="java.util.*;java.text.*" %>
```

Directiva include

Utilizada para incluir físicamente el contenido de un archivo dentro de otro

El archivo incluido puede ser cualquier archivo: HTML, JSP u otros.

Por ejemplo:

```

[/code]
<HTML>
<BODY>
A continuación se incluye el archivo hola.jsp<BR>
<%@ include file="hola.jsp" %>
</BODY>
</HTML>
[/code]

```

EL PATRÓN MODELO - CONTROLADOR - VISTA

Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

El Modelo: es la representación del modelo de negocio, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). No debe verse afectado por cambios en la interfaz de usuario. Son todas las clases Java que representan el negocio, por ejemplo: Persona, Auto, Alumno, etc.

La Vista: representa la interfaz de usuario. Es lo que ve el usuario final cuando interactúa con nuestra aplicación. En Java, la vista es representada por las páginas JSP.

El Controlador: hace de intermediario entre la 'vista' y el 'modelo'. Recibe los pedidos de la vista (request), y, si bien no lo resuelve, conoce que clase del Modelo puede responder (response) a dicho pedido. Cuando obtiene una respuesta del Modelo, decide a que JSP debe dirigirse con dicho response.

