

Introducción a JSP

JavaServer Pages (JSP)

- Es una tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web. Los archivos JSP combinan HTML con etiquetas especiales y fragmentos de código Java.

JSP

- El código fuente de una página JSP puede contener:
- - Directivas: Indican información general de la página, como puede ser importación de clases, página a invocar ante errores, si la página forma parte de una sesión, etc.
- - Declaraciones: Sirven para declarar métodos o variables.
- - Scriptlets: Código Java embebido.
- - Expresiones: Expresiones Java que se evalúan y se envían a la salida.
- - Tags JSP: Etiquetas especiales que interpreta el servidor.

Un ejemplo simple de archivo JSP

- `<%@page import="java.util.*"%> //Directiva`
- `<%! String cadena="Bienvenidos a JSP"; %>`
`//declaración`
- `<html>`
- `<body>`
- `<%= cadena %> //Expresión`
- `
`
- `<% out.println(new Date()); %> //scriptlet`
- `</body>`
- `</html>`

Elementos de JSP

- **1.1. Código Java**
- Podemos insertar código Java dentro de JSP de tres formas:
 - Expresiones,
 - *scriptlets* y
 - *declaraciones*.

JSP : Expresiones

- Son fragmentos de código Java, con la forma `<%= expresión %>` que se evalúan y se muestran en la salida del navegador. En general, dentro de una expresión podemos usar cualquier cosa que usaríamos dentro de un `System.out.print(expr);`
- Ejemplos:
- `<%= "Tamaño de cadena: "+cadena.length() %>`
- `<%= new java.util.Date() %>`
- `<%= Math.PI*2 %>`

Ejemplos : Scriptlets

- `<table>`
- `<% for (int i=0;i<10;i++)`
- `{`
- `%>`
- `<tr><td> <%=i%> </td></tr>`
- `<% }`
- `%>`
- `</table>`
- `//-----//`
- `<%`
- `out.println("<table>");`
- `for (int i=0;i<10;i++)`
- `out.println("<tr><td>"+i+"</td></tr>");`
- `out.println("</table>");`
- `%>`

Scriptlets

```
<%
```

```
String parametro1=request.getParameter("parametro1");
```

```
String parametro2=request.getParameter("parametro2");
```

```
MiClase miClase=new MiClase();
```

```
String salida=miClase.procesa(parametro1, parametro2);
```

```
%>
```

```
<%= salida %>
```


Comentarios en JSP

- dentro de un *scriptlet* o *declaración*

<%-- Comentario JSP --%>

<!-- Comentario HTML --> //declaración

<%

// Comentario

/* Comentario */

%>

Declaraciones

- Contienen declaraciones de variables o métodos, con la forma **<%! declaración %>**. Estas variables o métodos serán accesibles desde cualquier lugar de la página JSP. Hay que tener en cuenta que el servidor transforma la página JSP en un *servlet*, y éste *es usado por múltiples peticiones, lo que provoca que las variables conserven su valor entre sucesivas ejecuciones.*

Declaraciones

```
<%! int numeroAccesos=0; %>
```

```
<html>
```

```
<body>
```

```
<%=
```

```
"La página ha sido accedida "+(++numeroAccesos)+  
" veces desde el arranque del servidor"
```

```
%>
```

```
</body>
```

```
</html>
```

Declaraciones

```
<%! java.util.Date primerAcceso=new  
    java.util.Date(); %>
```

```
<html>
```

```
<body>
```

El primer acceso a la página se realizo en:

```
<%= primerAcceso %>
```

```
</body>
```

```
</html>
```

Declaraciones

```
<%!  
private String ahora()  
{  
return ""+new java.util.Date();  
}  
%>  
<html>  
<body>  
<%= ahora() %>  
</body>  
</html>
```

Directivas

- Las directivas son elementos que proporcionan información al motor JSP, e influirán en la estructura del *servlet generado*. *Hay tres tipos de directivas: page, taglib e include.*
- page: Se indica con la forma **<%@ page atributo="valor">**

Directivas :page

- Importar clases. Importar código, de la misma forma que se realiza en un programa en Java, se indica con el atributo import.
- Ejemplo:

```
<%@page import="java.io.*,  
miPackage.miClase"%>
```

Directivas : page

- Indicar si la página tendrá acceso a la sesión. Se especifica con el atributo **session**.
- Ejemplo:

```
<%@page session="true"  
import="java.util.ArrayList"%>
```


Directivas : page

- **Gestión de errores.** Permite redireccionar a una página cuando se produzca un error, se indica con los atributos `errorPage` y `isErrorPage`. Ejemplos:

```
<%@page errorPage="error.jsp">
```

```
[...]
```

```
<%@page isErrorPage="yes">
```

```
<html>
```

```
<body>
```

```
Error, contacte con el administrador [...]
```

```
</body>
```

```
</html>
```

Directivas:include

- Permite incluir un archivo en el lugar donde se especifique, al contrario que con la acción `<jsp:include>` que veremos más adelante, la directiva `include` simplemente copia el contenido del archivo byte a byte, siendo el resultado similar a si copiáramos el texto del archivo incluido y lo pegáramos en el JSP.

Directivas:include

```
<html>
```

```
<head>
```

```
<%@ include file="titulo.txt"%>
```

```
</head>
```

```
<body>
```

```
<%@ include file="cuerpoPagina.jsp"%>
```

```
</body>
```

```
</html>
```

Directiva : taglib

- taglib: Se emplea para indicar que se van a emplear librerías de etiquetas.
- Ejemplo:

```
<%@ taglib prefix="c"  
    uri="http://java.sun.com/jsp/jstl/core" %>
```

Acciones

Las acciones tienen la forma

<jsp:accion [parámetros]/>

, y tienen diversos usos, entre los que destacan la inclusión de páginas y transferencia de control.

Acciones:Inclusión de páginas

- **<jsp:include page="pagina.jsp">**
- Incluye la salida de otra página JSP en la actual, al contrario que con la directiva `<%@include file="fichero.ext"%>` la página incluida se *ejecuta y su salida se inserta en la página que la incluye, con la directiva se incluye el contenido del archivo (no su salida) y se ejecuta conjuntamente con la página principal.*

Acciones : inclusión de páginas

- La página incluida tiene acceso a los parámetros enviados a la principal, y podemos enviarle nuevos parámetros con la subetiqueta
- **`<jsp:param name="nombre" value="valor"/>`**

Acciones: inclusión de páginas

```
<html>  
<head>  
<jsp:include page="cabecera.jsp"/>  
</head>  
<body>  
<jsp:include page="cuerpo.jsp">  
<jsp:param name="tipo" value="paginaPrincipal"/>  
</jsp:include>  
</body>  
</html>
```


Acciones : transferencia de control

- **<jsp:forward page="pagina.jsp"/>**
- La petición es redirigida a otra página, y la salida de la actual se descarta. Al igual que con la inclusión, la página a la que se redirige tiene acceso a los parámetros pasados a la actual, y es posible el envío de nuevos parámetros.
- Ejemplo:

<jsp:forward page="principal.jsp">

<jsp:param name="titulo" value="Principal"/>

</jsp:forward>

Objetos implícitos

- request
- Response
- out
- session
- Application
- Config
- pageContext
- page

Objetos implícitos

- Podemos acceder al *JavaDoc* de estas clases para ver los atributos y funciones disponibles en las direcciones:
- <http://java.sun.com/products/servlet/reference/api/index.html>
- <http://java.sun.com/products/jsp/reference/api/index.html>

Objeto Request

- Es un objeto de la clase **HttpServletRequest**, su uso principal es el acceso a los parámetros de la petición.
- Funciones/métodos principales :
- **String getParameter(String name)**
Devuelve el valor de un parámetro.
- **Enumeration getParameterNames()**
Devuelve una enumeración con los nombres de todos los parámetros de la petición.

Objeto Request

- **String[] getParameterValues(String name)**

Los parámetros pueden tener valor múltiple, con esta función recuperamos un array con todos los valores para un nombre dado.

- **String getRemoteAddr()**

Devuelve la IP del *host desde el que se realiza la petición*

- **String getRemoteHost()**

Devuelve el nombre del *host desde el que se realiza la petición*.

Objeto Request

```
<html>
<body>
<form>
<input type="text" name="parametro"/>
<input type="submit"/>
</form>
<br>
<br>
Su IP: <%=request.getRemoteAddr()%>
<br>
Su nombre de host: <%= request.getRemoteHost() %>
<br>
Valor del parámetro:
<%= request.getParameter("parametro") %>
</body>
</html>
```

Objeto Response

- Es un objeto de la clase HttpServletResponse, que asiste al *servlet en su generación de la respuesta para el cliente, contiene funciones para manejo de cabeceras, códigos de estado, cookies y transferencia de control.*