

# JSP: Java Server Pages

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación  
Universidad Complutense de Madrid

14 de mayo de 2007

## Java Server Pages

Java empotrado dentro de HTML.

```
<html> 1
<body> 2
<% 3
String nombre = request.getParameter("nombre"); 4
if (nombre==null) { 5
    nombre = "Mundo"; 6
} 7
%> 8
<p>Hola <%=nombre %></p> 9
</body> 10
</html> 11
```

Marcas especiales:

- <% código Java %>
- <%=expresión %>

Variable predefinida `request`.

# Traducción a servlet I

```
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;

public class pr1_jsp extends HttpJspBase {

    private static java.util.Vector _jspx_includes;

    public java.util.List getIncludes() {
        return _jspx_includes;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
        /* Cuerpo del método que realiza el servicio */
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18

# Traducción a servlet II

```
public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws java.io.IOException, ServletException {
    JspFactory _jspxFactory = null;
    javax.servlet.jsp.PageContext pageContext = null;
    HttpSession session = null;
    ServletContext application = null;
    ServletConfig config = null;
    JspWriter out = null;
    Object page = this;
    JspWriter _jspx_out = null;
    try {
        _jspxFactory = JspFactory.getDefaultFactory();
        response.setContentType("text/html;charset=ISO-8859-1");
        pageContext = _jspxFactory.getPageContext(this, request, response, null, true, 8192, true);
        application = pageContext.getServletContext();
        config = pageContext.getServletConfig();
        session = pageContext.getSession();
        out = pageContext.getOut();
        _jspx_out = out;
        /* Traducción de la página */
    } catch (Throwable t) {
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0) out.clearBuffer();
        if (pageContext != null) pageContext.handlePageException(t);
    } finally {
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

# Traducción a servlet III

```
    if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
}

```

26  
27

## Traducción de la página

```
    out.write("<!-- $Id: transparencias.tex,v 1.3 2007-05-14 17:14:33 luis Exp $ -->\n");
    out.write("<html>\n");
    out.write("<body>\n");

    String nombre = request.getParameter("nombre");
    if (nombre==null) {
        nombre = "Mundo";
    }

    out.write("\n");
    out.write("<p>Hola ");
    out.print(nombre);
    out.write("</p>\n");
    out.write("</body>\n");
    out.write("</html>");

```

2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

```
<html>
<!-- $Id: transparencias.tex,v 1.3 2007-05-14 17:14:33 luis Exp $ -->
<body>
<%
int numero;
try {
    numero = Integer.parseInt(request.getParameter("numero"));
} catch (NumberFormatException e) {
    numero=1;
}
%>

<table>
<%
    for (int i = 1; i <= 10 ; i++) {
    %>
<tr><td><%=i %></td><td><%=numero %></td><td><%=i*numero %></td></tr>
<%
    }
    %>
</table>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

```

<table>
  <%
    for (int i = 1; i <= 10; i++) {
      out.print("<tr>");
      out.print("<td>"+i+"</td>");
      out.print("<td>"+numero+"</td>");
      out.print("<td>"+(i*numero)+"</td>");
      out.println("</tr>");
    }
  %>
</table>

</body>
</html>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

## Directivas: `<%@ directiva atributos %>` |

```

<html>
<body>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.DateFormat" %>
<%@ page import="java.text.SimpleDateFormat" %>

<%
  Date date = new Date();
  DateFormat df = new SimpleDateFormat("dd' de 'MMMM' de 'yyyy");
  %>
<p>Hoy es <%=df.format(date)%></p>
</body>
</html>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

```

<%@ include file="jsp/cabecera.jsp" %>
<h1>Capítulo 1</h1>
  <%@ include file="html/capitulo1.html" %>
<h1>Capítulo 2</h1>
  <%@ include file="html/capitulo2.html" %>
<h1>Capítulo 3</h1>
  <%@ include file="html/capitulo3.html" %>
<h1>Capítulo 4</h1>
  <%@ include file="html/capitulo4.html" %>
<%@ include file="html/pie.html" %>

```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

## Directivas: <%@ directiva atributos %> ||

## Habichuelas Java: *JavaBeans*

```
<body>
<jsp:useBean id="valor" scope="request" class="valores.Valor"/>
<jsp:setProperty name="valor" property="*/>
<%
    if (valor.getNombre()==null || valor.getIbex_id()==null ||
        valor.getUrl()==null || valor.getTipo()==0 ) {
        String nombre = valor.getNombre()==null?"":valor.getNombre();
        String ibex_id = valor.getIbex_id()==null?"":valor.getIbex_id();
        String url = valor.getUrl()==null?"":valor.getUrl();
        int tipo = valor.getTipo()==0?0:valor.getTipo();
    }
    <form action="pr5.jsp" method="get">
    <p>
        <label>Nombre: </label><input name="nombre" value="<%=nombre %>"><br>
        <label>ibex_id: </label><input name="ibex_id" value="<%=ibex_id %>"><br>
        <label>URL: </label><input name="url" value="<%=url %>"><br>
        <label>Tipo </label>
        <select name="tipo">
            <option value="<%=Valor.ACCIONES %>">Acciones </option>
            <option value="<%=Valor.FONDOS %>">Fondos </option>
        </select>
        <button type="submit">Enviar </button>
    </p>
    </form>
    <% } else { %>
    <jsp:getProperty name="valor" property="datos"/>
    <% } %>
</body>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

# Etiquetas personalizadas I

```
<body>
<%
String nombre = request.getParameter("nombre");
String ibex_id = request.getParameter("ibex_id");
String url = request.getParameter("url");
String strTipo = request.getParameter("tipo");

int tipo = strTipo==null? 0:Integer.parseInt(strTipo);
if (nombre==null || url==null || tipo==0) {
%>
<form action="pr6.jsp" method="get">
<p>
<label>Nombre:</label><input name="nombre" value="<%=nombre==null?" : nombre %>"></input>
<label>Ibex ID:</label><input name="ibex_id" value="<%=ibex_id==null?" : ibex_id %>"></input><br>
<label>url:</label><input name="url" value="<%=url==null?" : url %>"></input><br>
<label>Tipo:</label>
<select name="tipo">
<option value="<%=Valor.ACCIONES %>">Acciones</option>
<option value="<%=Valor.FONDOS %>">Fondos</option>
</select>
<br>
<button type="submit">Enviar</button>
</p>
</form>
<% } else { %>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

# Etiquetas personalizadas II

```
<% } else { %>
<%@ taglib uri="xml/valores.tld" prefix="valores" %>
<valores:repite ini="5" fin="8">
<% if (ibex_id==null || ibex_id.equals("")) { %>
<valores:entrada nombre="<%=nombre %>" url="<%=url %>" tipo="<%=tipo %>" />
<% } else { %>
<valores:entrada nombre="<%=nombre %>" ibex_id="<%=ibex_id %>" url="<%=url %>" tipo="<%=tipo %>" />
<% } %>
</valores:repite>
<% } %>
</body>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11

## Variables Predefinidas

- `request` El `HttpServletRequest` asociada con esta petición.
- `response` El `HttpServletResponse` asociada con esta petición.
- `out` El `PrintWriter` usado para mandar la salida al usuario.
- `session` El `HttpSession` asociada con esta petición. Las sesiones están activadas por defecto. Recordemos el método `response.encodeURL`.
- `application` El `ServletContext` del servlet.
- `config` El `ServletConfig` del servlet.
- `pageContext` Un objeto de clase `PageContext` para encapsular ciertas características de los JSP.
- `page` El objeto que respresenta al sevlet: `this`.

## Sesión

```
<body>
<%
if (request.getParameter("borrarSesion")!=null) {
    session.removeAttribute("num");
}
Integer num=(Integer)session.getAttribute("num");
if (num==null) {
    num = new Integer(0);
}
num = new Integer(num.intValue()+1);
session.setAttribute("num",num);
%>
<p>Has accedido <%=num.intValue()%> veces a la página</p>
<form action="pr7.jsp">
  <p><button name="borrarSesion" type="submit">Borrar Sesión</button></p>
  <p><button name="continuarSesion" type="submit">Continuar Sesión</button></p>
</form>
</body>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18

## Directivas `page`

Atributos:

**import** Las cláusulas **import** debe haber en el servlet.

```
<%@ page import="java.util.*" %> 1
<%@ page import="java.text.DateFormat" %> 2
<%@ page import="java.text.SimpleDateFormat" %> 3
```

**contentType** Especifica el tipo MIME:

```
<%@ page contentType="text/html; charset=iso-8859-15" %> 1
```

**session** Valor **true** (por defecto) usa sesiones, valor **false** no.

**extends** Qué debe extender el servlet.

**info** La cadena que se obtiene a través del método **getServletInfo**.

## Directivas `page`

Atributos:

**isThreadSafe** Valor **true** (por defecto) es un servlet normal, el valor **false** indica que el servlet implementa **SingleThreadModel**.

**buffer** Especifica el tamaño del **JspWriter**. Por defecto toma el valor que se especifica en el servidor.

**autoflush** Valor **true** (por defecto) indica que el **buffer** es **flushed** cuando está lleno, si tiene el valor **false** se provoca una excepción.

**errorPage** Página JSP que procesa las exceptions.

**isErrorPage** Indica si la página es de error o no.

**language** El único valor posible el **Java**

## Directivas `include`

Incluye una página en *tiempo de compilación*

```
<%@ include file="jsp/pie.jsp" %>
```

1

Pueden incluir HTML estático y marcas JSP.

## Acciones JSP

Las acciones JSP usan sintaxis XML para controlar el comportamiento del servlet:

`jsp:include` Incluye una página en tiempo de ejecución.

`jsp:useBean` Usa una *habichuela* de Java.

`jsp:setProperty` Pone una propiedad de una *habichuela* de Java.

`jsp:getProperty` Consulta una propiedad de una *habichuela* de Java.

`jsp:forward` Dirige la petición a otra página.

`jsp:plugin` Genera las etiquetas para el plugin de JAVA para el navegador.

## Acción `jsp:include` I

```
<jsp:include page="jsp/cabecera.jsp" /> 1
<h1>Capítulo 1</h1> 2
<jsp:include page="html/capitulo1.html" /> 3
<h1>Capítulo 2</h1> 4
<jsp:include page="html/capitulo2.html" /> 5
<h1>Capítulo 3</h1> 6
<jsp:include page="html/capitulo3.html" /> 7
<h1>Capítulo 4</h1> 8
<jsp:include page="html/capitulo4.html" /> 9
<%@ include file="jsp/pie.jsp" %> 10
```

```
out.write("<!-- $Id: transparencias.tex,v 1.3 2007-05-14 17:14:33 luis Exp $ I->\n"); 1
JspRuntimeLibrary.include(request, response, "html/cabecera.html", out, false); 2
out.write("\n"); 3
out.write("<h1>Capítulo 1"); 4
out.write("</h1>\n "); 5
JspRuntimeLibrary.include(request, response, "html/capitulo1.html", out, false); 6
out.write("\n"); 7
out.write("<h1>Capítulo 2"); 8
out.write("</h1>\n "); 9
JspRuntimeLibrary.include(request, response, 10
"html/capitulo2.html", out, false); 11
/* Resto de la traducción */ 12
13
```

## Acción `jsp:include` II

```
Date date = new Date(); 1
DateFormat df = new SimpleDateFormat("dd' de 'MMMM' de 'yyyy"); 2
3
out.write("\n "); 4
out.write("<p>"); 5
out.print(df.format(date)); 6
out.write("</p>\n "); 7
out.write("</body>\n"); 8
out.write("</html>\n"); 9
out.write("\n"); 10
```

# Las habichuelas Java

Un *JavaBean* es un objeto de una clase cuyos métodos públicos son de la forma `getXXXX` o `setXXXX`.

# Las habichuelas Java I

```
public class Valor extends TagSupport {
    1
    2
    public final static int ACCIONES=1;
    3
    public final static int FONDOS=2;
    4
    public final static String strAcciones="acciones";
    5
    public final static String strFondos="fondos";
    6
    7
    private String nombre=null;
    8
    private String ibex_id=null;
    9
    private int tipo=0;
    10
    private String url=null;
    11
    12
    public Valor() {
    13
        super();
    14
    }
    15
    16
    public void setNombre(String _nombre) {
    17
        nombre = _nombre;
    18
    }
    19
    public String getNombre() {
    20
        return nombre;
    21
    }
    22
```

## Las habichuelas Java II

```
public void setIbex_id(String _ibex_id) {
    ibex_id = _ibex_id;
}
public String getIbex_id() {
    return ibex_id;
}

public void setUrl(String _url) {
    url = _url;
}
public String getUrl() {
    return url;
}

public void setTipo(int _tipo) {
    tipo = _tipo;
}

public int getTipo() {
    return tipo;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

## Las habichuelas Java III

```
public String getDatos() {
    StringBuffer out = new StringBuffer();
    out.append("<dl>\n");
    out.append(" <dt>Nombre<dd><a href=\""+url+"\">"+nombre+"</a>");
    out.append(" <dt>Apellidos<dd>"+ibex_id);
    out.append(" <dt>Tipo<dd>"+(tipo==1?"acciones":"fondos"));
    out.append("</dl>\n");
    return out.toString();
}

public static int parse(String s) {
    if (s.equals(strAcciones)) {
        return ACCIONES;
    } else if (s.equals(strFondos)) {
        return FONDOS;
    } else {
        throw new TipoValorException("La cadena '"+s+"' no representa un tipo de valor");
    }
}

public final int doStartTag() throws JspException {
    .....
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25

## Las habichuelas Java

Indicar que se va a usar un JavaBean

```
<jsp:useBean id="valor" scope="request" class="valores.Valor"/>
```

1

**id** El nombre que le damos al objeto

**scope** El ámbito donde vive el objeto

**request** Válido para el request. Valor por defecto

**session** Válido para la sesión.

**application** Válido para todas páginas que comparten el **ServletContext**.

**class** La clase que define el JavaBean

**type** El tipo estático (la clase o una superclase suya).

## Las habichuelas Java

Dar valores a los *campos* del JavaBean. El normal, pone el valor que se indique

```
<jsp:useBean id="nombre" ... /> 1
... 2
<jsp:setProperty name="nombre" 3
    property="propiedad" ... /> 4
```

Sólo da el valor si construye el objeto.

```
<jsp:useBean id="myName" ... > 1
... 2
<jsp:setProperty name="myName" 3
    property="someProperty" ... /> 4
</jsp:useBean> 5
```

## Las habichuelas Java

El `jsp:setProperty` puede tener los siguientes atributos:

**name** Usa el JavaBean cuyo `id` se indique.

**property** Indica el nombre de la propiedad que se quiera poner. El valor `*` indica los parámetros del request que case con algún `setXXXX` del JavaBean.

**value** (Opcional) el valor que se da la propiedad.

**param** (Opcional) el valor se toma del parámetro de request que se indica. Si no existe tal parámetro *no se hace nada*.

Se pueden omitir tanto **value** como **param**, entonces el valor se toma del parámetro del **request** cuyo nombre es **property**.

## Las habichuelas Java

Invoca al método `elAgenda.setNombre("Patata")`.

```
<jsp:setProperty name="valor" property="nombre" value="Patata"/>
```

 1

Si `request.getParameter("patata")` no es `null`, llama a `valor.setNombre(request.getParameter("patata"))`.

```
<jsp:setProperty name="valor" property="nombre" value="patata"/>
```

 1

Si `request.getParameter("nombre")` no es `null`, llama a `valor.setNombre(request.getParameter("nombre"))`.

```
<jsp:setProperty name="valor" property="nombre"/>
```

 1

Rellena los el JavaBean con los parámetros del request, si no hay no hace nada.

```
<jsp:setProperty name="valor" property="*/>
```

 1

## Las habichuelas Java

Acceder valores a los *campos* del JavaBean.

```
<dl>
  <dt>Nombre</dt>
  <dd><a href='<jsp:getProperty name="valor" property="url"/>'>
    <jsp:getProperty name="valor" property="nombre"/></a></dd>
  <dt>Ibex ID</dt>
  <dd><jsp:getProperty name="valor" property="ibex_id"/></dd>
  <dt>Tipo</dt><dd><jsp:getProperty name="valor" property="strTipo"/></dd>
</dl>
```

 1  
2  
3  
4  
5  
6  
7  
8

# Las habichuelas Java

```
<body>
<jsp:useBean id="valor" scope="request" class="valores.Valor"/>
<jsp:setProperty name="valor" property="*" />
<%
    if (valor.getNombre()==null || valor.getIbex_id()==null ||
        valor.getUrl()==null || valor.getTipo()==0 ) {
        String nombre = valor.getNombre()==null?"":valor.getNombre();
        String ibex_id = valor.getIbex_id()==null?"":valor.getIbex_id();
        String url = valor.getUrl()==null?"":valor.getUrl();
        int tipo = valor.getTipo()==0?0:valor.getTipo();
    }
    <form action="pr5.jsp" method="get">
    <p>
        <label>Nombre:</label><input name="nombre" value="<%=nombre%>"><br>
        <label>ibex_id:</label><input name="ibex_id" value="<%=ibex_id%>"><br>
        <label>URL: </label><input name="url" value="<%=url%>"><br>
        <label>Tipo </label>
        <select name="tipo">
            <option value="<%=Valor.ACCIONES%>">Acciones</option>
            <option value="<%=Valor.FONDOS%>">Fondos</option>
        </select>
        <button type="submit">Enviar</button>
    </p>
    </form>
    <% } else { %>
    <jsp:getProperty name="valor" property="datos" />
    <% } %>
</body>
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid

JSP: Java Server Pages

# Taglibs definidas por el usuario

## 1 La sintaxis de las acciones jsp es XML

```
<jsp:useBean id="valor" scope="request" class="valores.Valor"/>
<jsp:setProperty name="valor" property="*" />
```

1  
2

## 2 El XML *mola mucho*

## 3 La implementación de los servlets es Software Libre

¿Por qué no puedo hacerlo yo para mis propias aplicaciones?

¿Por qué no puedo definir algo como?

```
<valores:entrada nombre="<%=nombre%>"
    idex_id="<%=apellidos%>" tipo="<%=tipo%>" />
```

1  
2

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid

JSP: Java Server Pages

## Taglibs definidas por el usuario

¿Qué hace falta?

- Fichero de descripción de las etiquetas `.tld`.
- Un *manejador* que implementa lo que quiero hacer con mis etiquetas.

## Ficheros `tld`

Es un fichero XML

```
<?xml version="1.0" encoding="iso-8859-1"?> 1
<!DOCTYPE taglib 2
  PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN" 3
  "http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd"> 4
<taglib> 5
  <tlib-version>1.0</tlib-version> 6
  <jsp-version>1.2</jsp-version> 7
  <short-name>Valor</short-name> 8
  <tag> 9
    <name>entrada</name> 10
    <tag-class>valores.Valor</tag-class> 11
    <body-content>empty</body-content> 12
    <attribute> 13
      <name>nombre</name> 14
      <required>true</required> 15
      <rtexprvalue>true</rtexprvalue> 16
    </attribute> 17
    <!-- más atributos --> 18
  </tag> 19
  <!-- más marcas tag --> 20
</taglib> 21
```

## Ficheros `tld`

Las etiquetas de la cabecera son todas opcionales, menos:

`tlib-version` Versión de la librería.

`jsp-version` Versión de JSP de la que depende.

`short-name` Nombre corto. Puede ser usado por las aplicaciones como prefijo preferido.

El resto de las marcas son `uri`, `display-name`, `small-icon`, `large-icon`, `description`, `validator`, `listener`. Luego puede haber una o más marcas `tag`.

## Ficheros `tld`. Marca `tag`

`name` nombre de la etiqueta

`tag-class` la clase (*manejador*) que implementa la etiqueta.

`body-content` Puede tomar los valores

`JSP` El contenido es JSP que se debe interpretar.

`tagdependent` El contenido está en otro lenguaje, por ejemplo SQL.

`empty` No tiene cuerpo.

**attribute** Los atributos que puede tener, Contiene las siguientes marcas:

**name** el nombre del atributo

**required** el el atributo es opcional o no (**true** o **false**).

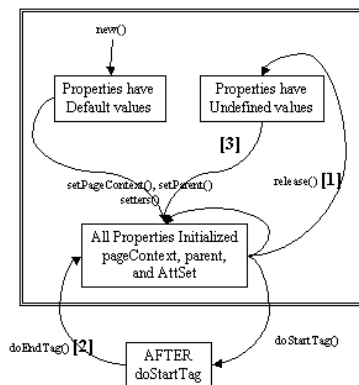
**rtexprvalue** si el contenido es estático o no (**true** o **false**).

## Tags sin cuerpo I

`javax.servlet.jsp.tagext.Tag`

`javax.servlet.jsp.tagext.TagSupport`

INIT PROTOCOL



- 1 Inicialización  
`setPageContext()`,  
`setParent()`, los `setXXXX`.
- 2 `doStartTag()` que devuelve `SKIP_BODY` (si no hay cuerpo) o `EVAL_BODY_INCLUDE`.
- 3 `doEndTag()` que devuelve `EVAL_PAGE` o `SKIP_PAGE`

## Tags sin cuerpo II

```
package valores;
import javax.servlet.jsp.tagext.TagSupport;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import java.io.IOException;
public class Valor extends TagSupport {

    private String nombre=null;
    private String ibex_id=null;
    private int tipo=0;
    private String url=null;

    public final String getNombre() {
        return nombre;
    }
    public final void setNombre(final String newNombre) {
        this.nombre = newNombre;
    }
    /* más getters y setters */
    public Agenda() {
        super();
    }
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

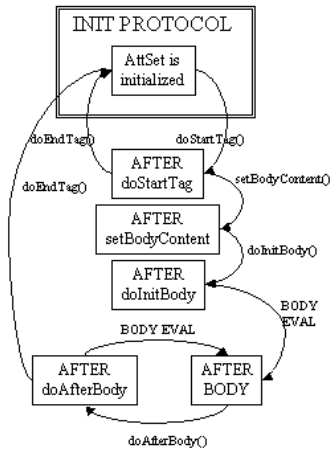
## Tags sin cuerpo III

```
public final int doStartTag() throws JspException {
    JspWriter out = pageContext.getOut();
    try {
        out.write("<dl>\n");
        out.write(" <dt>Nombre <dd><a href=\""+url+"\">"+nombre+"</a>");
        if (ibex_id!=null) {
            out.write(" <dt>Ibex ID<dd>"+ibex_id);
        }
        out.write(" <dt>Tipo<dd>"+(tipo==ACCIONES?strAcciones:strFondos));
        out.write("</dl>\n");
    } catch (IOException e) {
        throw new JspException("IO exception "+e.getMessage(),e);
    }
    return super.EVAL_BODY_INCLUDE;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18

# Tags con cuerpo I

javax.servlet.jsp.tagext.BodyTag  
javax.servlet.jsp.tagext.BodyTagSupport



- 1 Inicialización `setPageContext()`, `setParent()`, los `setXXXX`.
- 2 `doStartTag()` que devuelve `SKIP_BODY` (si no hay cuerpo) o `EVAL_BODY_INCLUDE`, `EVAL_BODY_BUFFERED`,
- 3 `doInitBody()`, `setBodyContent()`.
- 4 `doAfterBody` una o más veces. Devuelve `EVAL_BODY_AGAIN` para evaluar otra vez o `SKIP_BODY` para finalizar.
- 5 `doEndTag()` que devuelve `EVAL_PAGE` o `SKIP_PAGE`

# Tags con cuerpo II

```
<%@ taglib uri="xml/valores.tld" prefix="valores" %> 1
<valores:repite ini="5" fin="8"> 2
  <% if (ibex_id==null || ibex_id.equals("")) { %> 3
  <valores:entrada nombre="<%=nombre%>" url="<%=url%>" tipo="<%=tipo%>" /> 4
  <% } else { %> 5
  <valores:entrada nombre="<%=nombre%>" ibex_id="<%=ibex_id%>" url="<%=url%>" tipo="<%=tipo%>" /> 6
  <% } %> 7
</valores:repite> 8
```

```
<tag> 1
  <name>repite</name> 2
  <tag-class>valores.Repite</tag-class> 3
  <body-content>JSP</body-content> 4
  <attribute> 5
    <name>ini</name> 6
    <required>true</required> 7
    <rtexprvalue>true</rtexprvalue> 8
  </attribute> 9
  <attribute> 10
    <name>fin</name> 11
    <required>true</required> 12
    <rtexprvalue>true</rtexprvalue> 13
  </attribute> 14
</tag> 15
```

## Tags con cuerpo III

```
package valores;
1
2
import javax.servlet.jsp.tagext.BodyTagSupport;
3
import javax.servlet.jsp.JspException;
4
import javax.servlet.jsp.tagext.BodyContent;
5
import java.io.IOException;
6
7
public class Repite extends BodyTagSupport {
8
    private int ini;
9
    private int fin;
10
    private int act;
11
12
    /* getters and setters */
13
14
    public final void doInitBody() throws JspException {
15
        act = ini;
16
    }
17
```

## Tags con cuerpo IV

```
    public final int doAfterBody() throws JspException {
1
        act++;
2
        if (act<=fin) {
3
            return EVAL_BODY_BUFFERED;
4
        } else {
5
            return SKIP_BODY;
6
        }
7
    }
8
    public final int doEndTag() throws JspException {
9
        BodyContent body = getBodyContent();
10
        try {
11
            body.writeOut(pageContext.getOut());
12
        } catch (IOException e) {
13
            throw new JspException("IOException: "+e.getMessage(),e);
14
        }
15
        return EVAL_PAGE;
16
    }
17
18
19
```

¿Funciona bien con la siguiente?

## Tags con cuerpo V

```
<valores:repite ini="8" fin="5">
  <valores:entrada nombre="<%=nombre%>" ibex_id="<%=ibex_id%>"
    tipo="<%=tipo%>" />
</valores:repite>
```

1  
2  
3  
4

## Tags con cuerpo VI

```
<valores:mysql-conexion database="valores" host="localhost"
  user="luis" password="patata">
  <valores:busca ibex_id="<%=ibex_id%>" />
</valores:mysql-conexion>
```

1  
2  
3  
4

La etiqueta `valores:mysql-conexion` construye la conexión `connection`.

```
public final int doStartTag() throws JspException {
  try {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    connection = DriverManager.getConnection("jdbc:mysql://" + host +
      "/" + database + "?user=" + user +
      "&password=" + password);
  } catch (Exception e) {
    throw new JspException("Excepción" + e.getClass().getName() +
      " " + e.getMessage(), e);
  }
  return EVAL_BODY_BUFFERED;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

## Tags con cuerpo VII

```
public final int doEndTag() throws JspException {
    try {
        BodyContent body = getBodyContent();
        body.writeOut(pageContext.getOut());
        connection.close();
    } catch (Exception e) {
        throw new JspException("Excepción
                               "+e.getClass().getName()+
                               ": "+e.getMessage(),e);
    }

    return EVAL_PAGE;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

## Tags con cuerpo VIII

La clase que implementa `valores:mysql-conexion` tiene el `getConnection()`. La clase que implementa la búsqueda debe acceder a ella.

```
public final int doEndTag() throws JspException {
    try {
        Conexion conexion = (Conexion)findAncestorWithClass(
            this,
            Class.forName("valores.Conexion"));
        Connection con = conexion.getConnection();
        String sql = "select * from valores "+
            "where ibex_id like '%" + ibex_id + "%'";
        PreparedStatement psmt = con.prepareStatement(sql);
        ResultSet rs = psmt.executeQuery();
        JspWriter out = pageContext.getOut();
        out.write("<p>Resultados</p>");
        /* Se muestran los resultado */
    } catch (Exception e) {
        throw new JspException("Excepción "+e.getClass().getName()+
                               ": "+e.getMessage(),e);
    }

    return EVAL_PAGE;
}
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

Proyecto de Software libre en la que se implementan diversas taglibs:

- Librería estándar de etiquetas JSLT.

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>

<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSTL.html>

```
<c:forEach var="item" items="${sessionScope.cart.items}"> 1
  ... 2
</c:forEach> 3
```

```
<c:set var="bookId" value="${param.Remove}"/> 1
<jsp:useBean id="bookId" type="java.lang.String" /> 2
<% cart.remove(bookId); %> 3
<sql:query var="books" 4
  dataSource="${applicationScope.bookDS}"> 5
  select * from PUBLIC.books where id = ? 6
  <sql:param value="${bookId}" /> 7
</sql:query> 8
```