JSP: Java Server Pages

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

28 de abril de 2008

Java Server Pages

Java empotrado dentro de HTML.

```
<html>
<body>
<%
String nombre = request.getParameter("nombre");
if (nombre==null) {
    nombre = "Mundo";
}
%>
<p>Hola <%=nombre%></p>
</body>
</html>
```

Marcas espciales:

- `<% 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 que realiza el servicio */
  }
}
```

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, 819
    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 {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
  }
```

Traducción a servlet III

Traducción de la página

```
      out.write("<!-- $Id: transparencias.tex,v 1.2 2008-04-26 14:53:07 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>");
```

```
<html>
<!-- $Id: transparencias.tex,v 1.2 2008-04-26 14:53:07 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>
```

```
<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>
```

Directivas: <%@ directiva atributos %> I

```
<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>
```

```
<%@ 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" %>
```

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</opcion>
     </select>
   <button type="submit">Enviar</button>
</p>
</form>
<%  } else { %>
<jsp:getProperty name="valor" property="datos"/>
<%  } %>
</body>
```

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%>"><br>
   <label>Ibex ID:</label><input name="ibex_id" value="<%=ibex_id==null?"":ibex_id%>"><b
   <label>url:</label><input name="url" value="<%=url==null?"":url%>"><br>
   <label>Tipo:</label>
     <select name="tipo">
       <option value="<%=Valor.ACCIONES%>">Acciones</option>
       <option value="<%=Valor.FONDOS%>">Fondos</opcion>
     </select>
   <br>
   <button type="submit">Enviar</button>
</p>
</form>
<% } else { %>
```

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="<%=
   <% } %>
</valores:repite>
<%  }  %>
  </body>
```

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>
```

Directivas page

Atributos:

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

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

contentType Especifica el tipo MIME:

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

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" %>
```

Pueden incluir HTML estático y marcas JSP.

Acciones JSP

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

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

```
      out.write("<!-- $Id: transparencias.tex,v 1.2 2008-04-26 14:53:07 luis Exp $ -->\n
      JspRuntimeLibrary.include(request, response, "html/cabecera.html", out, false);
      out.write("\n");
      out.write("<h1>Capítulo 1");
      out.write("</h1>\n  ");
      JspRuntimeLibrary.include(request, response, "html/capitulo1.html", out, false);
      out.write("\n");
      out.write("<h1>Capítulo 2");
      out.write("</h1>\n  ");
      JspRuntimeLibrary.include(request, response,
      "html/capitulo2.html", out, false);

    /* Resto de la traducción */
```

Acción jsp:include II

```
Date date = new Date();
DateFormat df = new SimpleDateFormat("dd' de 'MMMM' de 'yyyy");

out.write("\n   ");
out.write("<p>");
out.print(df.format(date));
out.write("</p>\n  ");
out.write("</body>\n");
out.write("</html>\n");
out.write("\n");
```

Las habichuelas Java I

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

```
public class Valor  extends TagSupport {

    public final static int ACCIONES=1;
    public final static int FONDOS=2;
    public final static String strAcciones="acciones";
    public final static String strFondos="fondos";

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

    public Valor() {
        super();
    }

    public void setNombre(String _nombre) {
        nombre = _nombre;
    }
    public String getNombre() {
        return nombre;
    }
```

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;
 }
```

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 val
        }

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

Las habichuelas Java

Indicar que se va a usar un JavaBean

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

- `id` El nombre que le damos al objeto
- `scope` El ámbido 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" ... />
...
<jsp:setProperty name="nombre"
                 property="propiedad" ... />
```

Sólo da el valor si construye el objeto.

```
<jsp:useBean id="myName" ... >
  ...
  <jsp:setProperty name="myName"
                   property="someProperty" ... />
</jsp:useBean>
```

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"/>
```

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

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

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

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

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

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

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>
```

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</opcion>
     </select>
   <button type="submit">Enviar</button>
</p>
</form>
<%  } else { %>
<jsp:getProperty name="valor" property="datos"/>
<%  } %>
</body>
```

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="*"/>
```

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%>"/>
```

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"?>
<!DOCTYPE taglib
      PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
      "http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>Valor</short-name>
  <tag>
    <name>entrada</name>
    <tag-class>valores.Valor</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>nombre</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
    </attribute>
    <!-- más atributos -->
  </tag>
  <!-- más marcas tag -->
</taglib>
```

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` I

`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.

Ficheros `tld`. Marca `tag` I

`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

`javax.servlet.jsp.tagext.Tag`
`javax.servlet.jsp.tagext.TagSupport`

Tags sin cuerpo

`javax.servlet.jsp.tagext.Tag`
`javax.servlet.jsp.tagext.TagSupport`

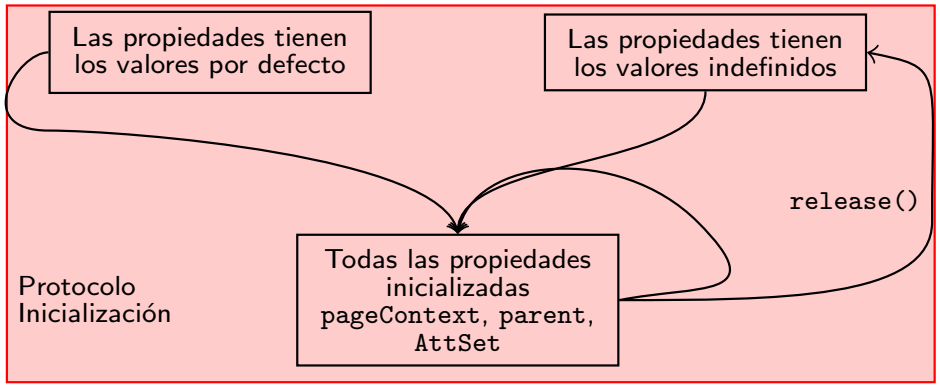
Las propiedades tienen los valores por defecto

Las propiedades tienen los valores indefinidos

1. Inicialización.

Tags sin cuerpo

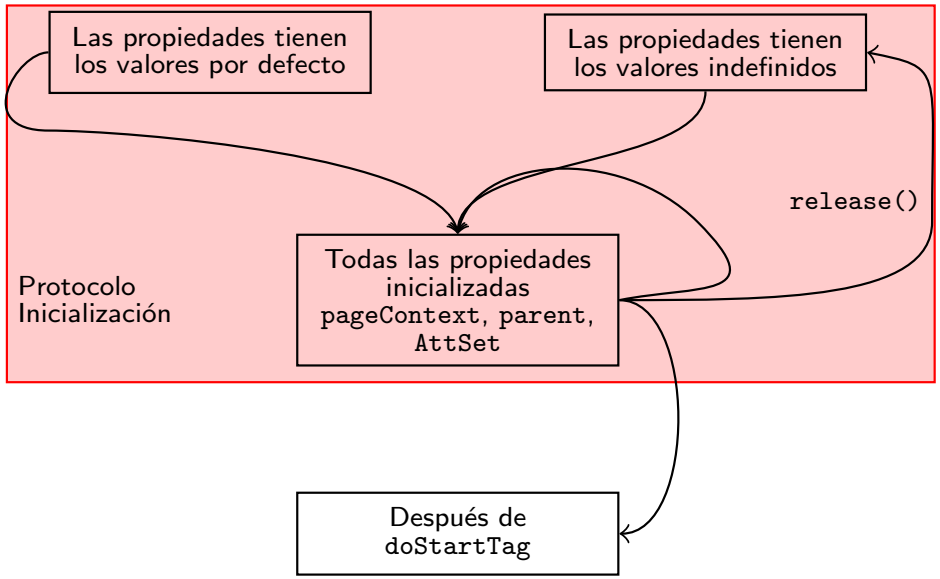
`javax.servlet.jsp.tagext.Tag`
`javax.servlet.jsp.tagext.TagSupport`



1. Inicialización.
2. `setPageContext()`, `setParent()`, los `setXXXX`.

Tags sin cuerpo

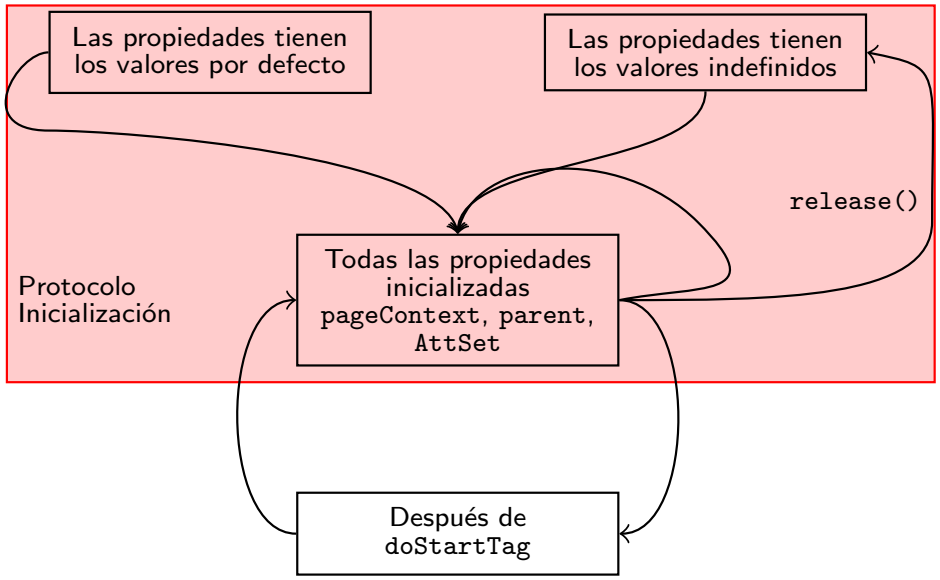
`javax.servlet.jsp.tagext.Tag`
`javax.servlet.jsp.tagext.TagSupport`



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

Tags sin cuerpo

`javax.servlet.jsp.tagext.Tag`
`javax.servlet.jsp.tagext.TagSupport`



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

```java
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();
    }
```

```
    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;
    }
}
```

Tags con cuerpo

`javax.servlet.jsp.tagext.BodyTag`

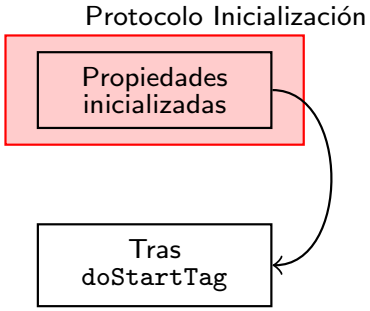
`javax.servlet.jsp.tagext.BodyTagSupport`

Protocolo Inicialización

Tags con cuerpo

`javax.servlet.jsp.tagext.BodyTag`

`javax.servlet.jsp.tagext.BodyTagSupport`

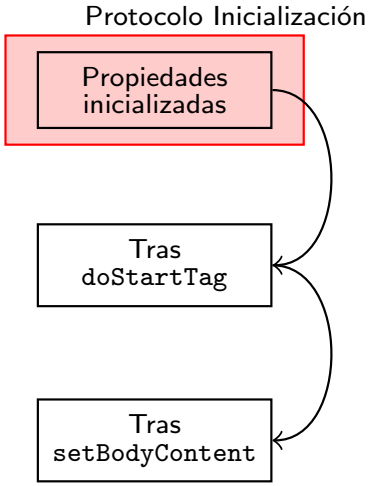


1. Inicialización `setPageContext()`, `setParent()`, los `setXXXX`.

Tags con cuerpo

`javax.servlet.jsp.tagext.BodyTag`

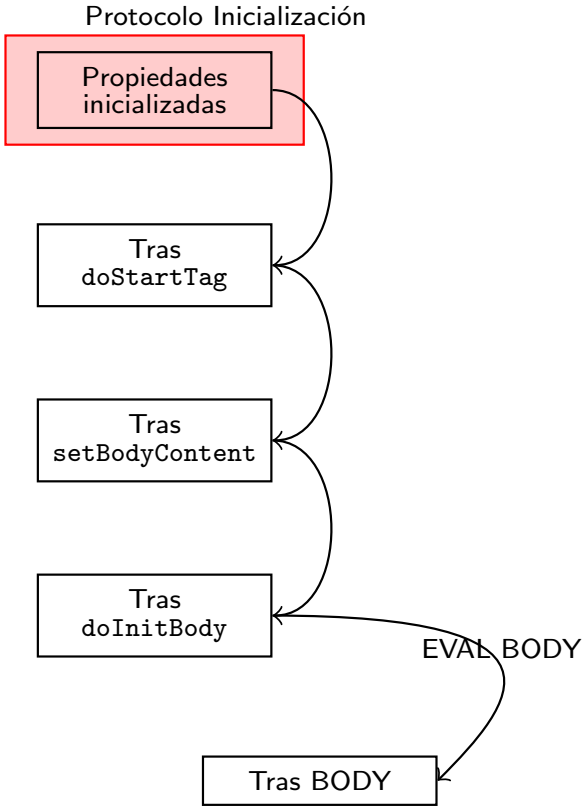
`javax.servlet.jsp.tagext.BodyTagSupport`



1. Inicialización `setPageContext()`, `setParent()`, los `setXXXX`.
2. `doStartTag()` que devuelve `SKIP_BODY` (si no hay cuerpo) o `EVAL_BODY_INCLUDE`, `EVAL_BODY_BUFFERED`,

Tags con cuerpo

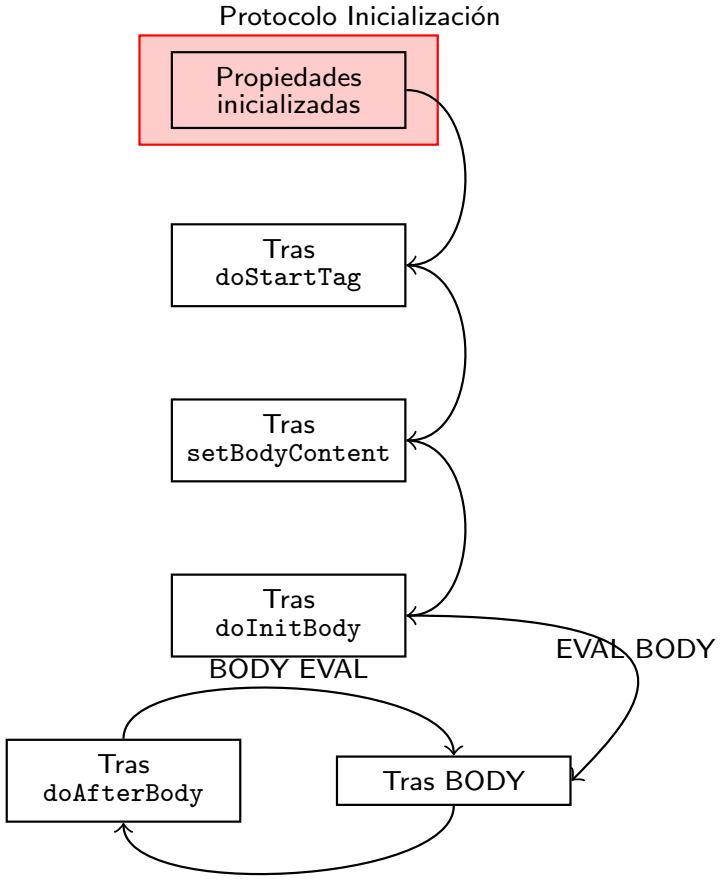
`javax.servlet.jsp.tagext.BodyTag`
`javax.servlet.jsp.tagext.BodyTagSupport`



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()`.

Tags con cuerpo

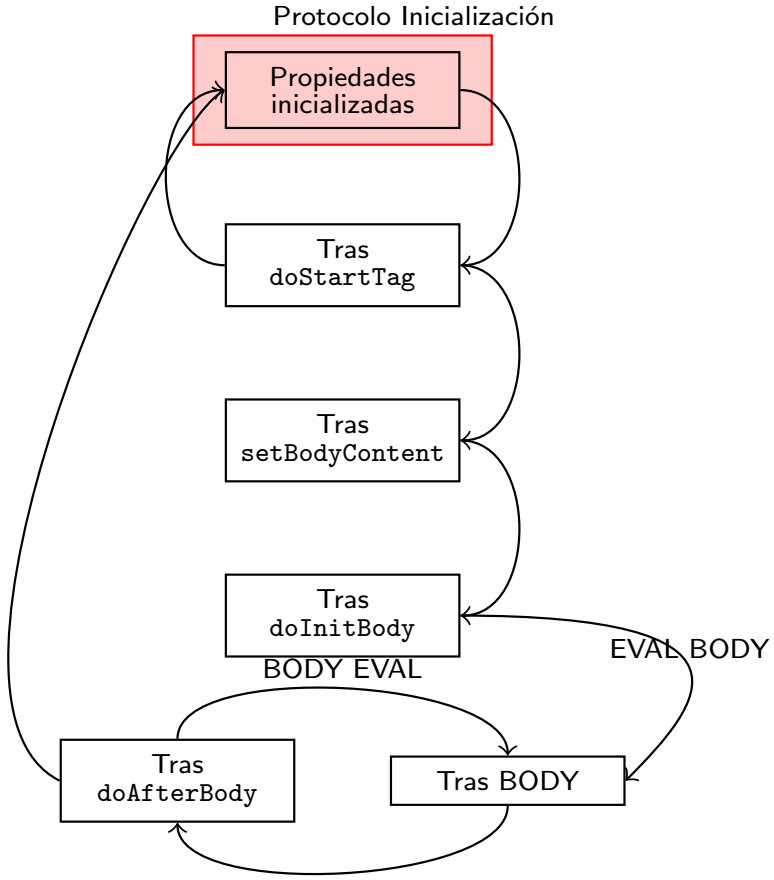
`javax.servlet.jsp.tagext.BodyTag`
`javax.servlet.jsp.tagext.BodyTagSupport`



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.

Tags con cuerpo

`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`

```
<%@ 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="<%=
   <% } %>
</valores:repite>
```

```
  <tag>
    <name>repite</name>
    <tag-class>valores.Repite</tag-class>
    <body-content>JSP</body-content>
    <attribute>
      <name>ini</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
    </attribute>
    <attribute>
      <name>fin</name>
      <required>true</required>
      <rtexprvalue>true</rtexprvalue>
   </attribute>
  </tag>
```

```java
package valores;

import javax.servlet.jsp.tagext.BodyTagSupport;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.BodyContent;
import java.io.IOException;

public class Repite extends BodyTagSupport {
    private int ini;
    private int fin;
    private int act;

    /* getters and setters */

    public final void doInitBody() throws JspException {
        act = ini;
    }
```

```java
    public final int doAfterBody() throws JspException {
        act++;
        if (act<=fin) {
            return EVAL_BODY_BUFFERED;
        } else {
            return SKIP_BODY;
        }
    }
    public final int doEndTag() throws JspException {
        BodyContent body = getBodyContent();
        try {
            body.writeOut(pageContext.getOut());
```

```
        } catch (IOException e) {
            throw new JspException("IOException: "+e.getMessage(),e);
        }

        return EVAL_PAGE;
    }
}
```

¿Funciona bien con la siguiente?

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

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

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;
    }
```

```
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;
}
```

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;
}
```

Jakarta Taglibs
http://jakarta.apache.org/taglibs/index.html

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

- Librería estándar de etuquetas 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}">
  ...
</c:forEach>
```

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