Índice

1. Creación del JavaBean

Se crea un proyecto `Dynamic Web Project` llamado `JSPtaller1`

En el directorio `src` creamos una clase llamada `Libro` que pertenece al paquete `edu.uv.dawts.jsp.taller1` y ponemos el siguiente código:

```
package edu.uv.dawts.jsp.taller1;

public class Libro {
        private String titulo;
        private String editorial;
        private String autor;
        private double precio;

        public Libro() {
        }

   // Setters y getters
}
```

Como vemos se trata de una clase que tiene `setters` y `getters` para los atributos que define y que tiene un constructor público sin argumentos (por defecto).

2. JSP que usa el JavaBean mediante scriplets

Vamos a realizar un JSP desde se crea una instancia del tipo anterior y se muestran los atributos en forma de tabla.

Creamos un JSP llamado `Scriptlets.jsp` y ponemos el siguiente contenido:

```
<%@ page import="edu.uv.dawts.taller3.Libro" language="java" contentType="text/html; charset=
    UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
    loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Elementos de script</title>
</head>
<body>

<%
    Libro l = new Libro();
    l.setAutor("Liskova");
    l.setEditorial("Pertnice Jall");
```

```
    l.setTitulo("Java programming");
    l.setPrecio(20.6);
%>

<h1> Libro </h1>

<table>
<tr><th>Autor </th> <th>Titulo</th> <th> Editorial </th> <th>Precio </th> </tr>
<tr>
    <td><%= l.getAutor() %> </td>
    <td><%= l.getTitulo() %> </td>
    <td><%= l.getEditorial() %> </td>
    <td><%= l.getPrecio() %> </td>
</tr>

 </table>

</body>
</html>
```

3. JSP que usa el JavaBean mediante elementos de acción

Creamos un nuevo JSP en el proyecto con el nombre `Acciones.jsp`

```
<%@ page import="edu.uv.dawts.taller3.Libro" language="java" contentType="text/html; charset
    =UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
    loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Elementos de accion</title>
</head>
<body>

<jsp:useBean id="libro" class="edu.uv.dawts.taller3.Libro" scope="page"></jsp:useBean>

<jsp:setProperty name="libro" property="autor" value="Liskova" />
<jsp:setProperty name="libro" property="editorial" value="Pertnice Jall" />
<jsp:setProperty name="libro" property="titulo" value="Java programming" />
<jsp:setProperty name="libro" property="precio" value="20.6" />


<h1> Libro </h1>

<table>
<tr><th>Autor </th> <th>Titulo</th> <th> Editorial </th> <th>Precio </th> </tr>
<tr>
    <td> <jsp:getProperty property="autor" name="libro"/> </td>
    <td> <jsp:getProperty property="titulo" name="libro"/> </td>
    <td> <jsp:getProperty property="editorial" name="libro"/> </td>
    <td> <jsp:getProperty property="precio" name="libro"/> </td>
</tr>

 </table>

</body>
</html>
```

4. JSP que usa el JavaBean mediante Expression Language

Creamos un nuevo JSP en el proyecto con el nombre `EL.jsp`:

```
<%@ page import="edu.uv.dawts.taller3.Libro" language="java" contentType="text/html; charset=
    UTF-8"
```

```
    pageEncoding="UTF-8" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
    loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Expression Language</title>
</head>
<body>

<%
    Libro l = new Libro();
    l.setAutor("Liskova");
    l.setEditorial("Pertnice Jall");
    l.setTitulo("Java programming");
    l.setPrecio(20.6);
    pageContext.setAttribute("libro", l);
 %>

<h1> Libro </h1>

<table>
<tr><th>Autor </th> <th>Titulo</th> <th> Editorial </th> <th>Precio </th> </tr>
<tr>
    <td> ${libro.autor} </td>
    <td> ${libro.titulo} </td>
    <td> ${libro.editorial} </td>
    <td> ${libro.precio} </td>
</tr>

 </table>

</body>
</html>
```

5. Configuración y despliegue para mantener el fuente del servlet generado

Depurar los JSPs no es fácil ya que los errores hacen referencia al número de línea del servlet generado (y no al número de línea del JSP). Si no detectamos el error necesitamos tener el código fuente del servlet para ver qué sentencia es la que provoca el error. El siguiente procedimiento explica cómo conseguir el fuente del servlet generado a partir del JSP.

Editar el fichero `default-web.xml` en el directorio `config` del dominio y donde aparece:

```
   <servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
    <init-param>
    </init-param>
    <load-on-startup>3</load-on-startup>
  </servlet>
```

Añadir el parámetro inicial `keepgenerated` y asignarle el valor `true`:

```
   <servlet>
    <servlet-name>jsp</servlet-name>
    <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
    <init-param>
      <param-name>keepgenerated</param-name>
      <param-value>true</param-value>
    </init-param>
    <load-on-startup>3</load-on-startup>
  </servlet>
```

Además, al realizar el despliegue en el servidor con `asadmin` se indica que debe precompilar el jps:

```
asadmin deploy --precompilejsp=true fichero.war
```

El código fuente del servlet generado se encuentra en:

```
opt/glassfish3/glassfish/domains/domain1/generated/jsp/NOMBRE-MODULO/org/apache/jsp
```

donde `NOMBRE-MODULO` es el nombre de la aplicación.