

7.9. Java Server Pages

Las *Java Server Pages (JSP)* constituyen una tecnología complementaria a los *Servlets* que acabamos de ver. Surgieron como respuesta a las *Active Server Pages* de *Microsoft* y permiten separar los bloques de *HTML* estáticos de los dinámicos generados por código *Java* que se ejecuta en el servidor. En resumen, permiten separar el código de presentación de la lógica de programación.

Simplificando, podemos afirmar que los *Servlet* son “*HTML* dentro de *Java*” y los *JSP* son “*Java* dentro de *HTML*”. Son tecnologías complementarias, en los modelos MVC (Modelo Vista Controlador) los *Servlets* se encargarán de controlar las peticiones, mientras que los *JSP* se encargarán de generar la vista de la aplicación.

Cuando un cliente solicita una página *.jsp*, se ejecuta en el servidor el código *JSP* de la página, dando como resultado una página *HTML* que se fusiona con el *HTML* original, generando una página *HTML* de respuesta que será enviada al cliente.

1. Primera página *JSP*.

- 1.1. Crea un nuevo *Dynamic Web Project* llamado **JavaServerPages**.
- 1.2. Pulsando con el botón derecho del ratón sobre la carpeta *WebContent* añade un nuevo fichero *JSP* llamado **expresiones.jsp**, Figura 7.56.

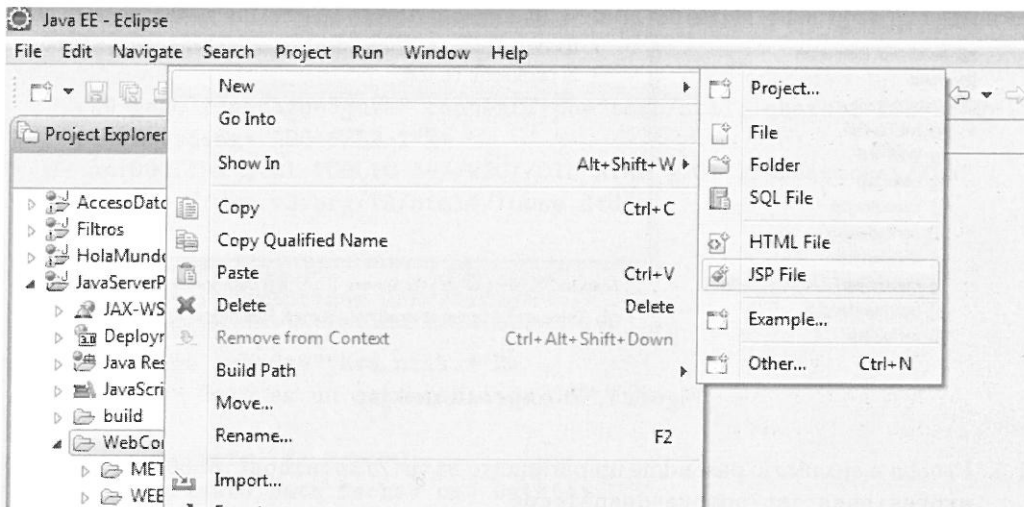


Figura 7.56: Nuevo fichero JSP

- 1.3. Para probar el uso de expresiones en *JSP*, añade el siguiente código al fichero **expresiones.jsp**:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
```

```

<title>Expresiones JSP</title>
</head>
<body>
  <h1>Ejemplo de expresiones JSP</h1>
  <ul>
    <li>Fecha actual: <%=new java.util.Date()%>
    <li>Nombre del host: <%=request.getRemoteHost()%>
    <li>ID de la sesión: <%=session.getId()%>
    <li>El parámetro es: <%=request.getParameter("nombre")%>
  </ul>
</body>
</html>

```

1.4. Pulsando con el botón derecho del ratón sobre el fichero **expresiones.jsp**, selecciona la acción **Run As, Run on Server**, Figura 7.57.

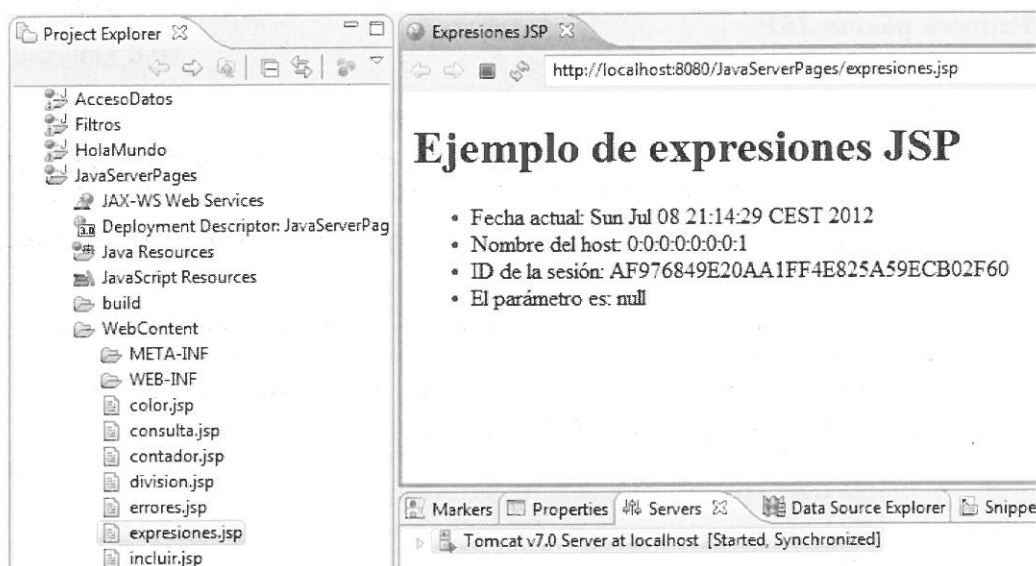


Figura 7.57: **expresiones.jsp**

1.5. Prueba a ejecutarlo pasándole un parámetro `http://localhost:8080/JavaServerPages/expresiones.jsp?nombre=despliegue`

2. Más ejemplos de *JSPs*.

2.1. Para probar el uso de *scriptlets* añade la página **color.jsp** con el siguiente código:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>

```

```

<title>Scriptlets JSP</title>
<%
    String bgColor=request.getParameter("bgColor");
    boolean hayColor;
    if (bgColor!=null) hayColor=true; else
    {
        hayColor=false;
        bgColor="WHITE";
    }
%>
</head>
<body BGCOLOR="<%=bgColor%>">
<h1>Ejemplo de scriptlets JSP</h1>
<%
    if (hayColor) out.println("Se ha utilizado el color: " + bgColor);
    else out.println("Se ha utilizado el color por defecto: WHITE");
%>
</body>
</html>

```

- 2.2. Prueba a ejecutarlo con las siguientes *URLs*: <http://localhost:8080/JavaServerPages/color.jsp> y <http://localhost:8080/JavaServerPages/color.jsp?bgColor=RED>.
- 2.3. Para probar el empleo de declaraciones y de la directiva *page* crea la página **contador.jsp** con el siguiente código:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Scriptlets JSP</title>
</head>
<%@page import="java.util.*"%>
<%-- Esto en un comentario de JSP --%>
<%!
    private int cont=0;
    private Date fecha= new Date();
%>
<body>
<p>Esta página ha sido accedida <b><%= ++cont%></b> veces.</p>
<p>El último acceso ha sido con fecha <b><%=fecha%></b></p>
<%
    fecha=new Date();
%>
</body>
</html>

```

- 2.4. Ejecútalo para comprobar su funcionamiento.

- 2.5. Ahora, para probar la directiva *include* crea un fichero **incluir.jsp** con el siguiente código:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo de uso de un contador incluido en un JSP</title>
  </head>
  <body>
    <h1>Ejemplo de uso de un contador incluido en un JSP</h1>
    <%include file="contador.jsp"%>
  </body>
</html>
```

- 2.6. Finalmente vamos a utilizar la directiva *page* para que ante cualquier error se ejecute una página *JSP* de tratamiento de errores.
- 2.7. Crea una página **division.jsp** que se encargue de dividir dos enteros que recibe como parámetros:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Manejo de errores de JSP</title>
  </head>
  <body >
    <%@page errorPage="errores.jsp"%>
    <h1>Ejemplo de manejo de errores en JSP</h1>
    <%!
      private double toDouble(String value)
      {
        return(Double.valueOf(value).doubleValue());
      }
    %>
    <%
      double op1 = toDouble(request.getParameter("op1"));
      double op2= toDouble(request.getParameter("op2"));
      double res = op1/op2;
    %>
    <table border=1>
    <tr><th></th><th>División</th></tr>
    <tr><th>Operando 1: </th><td><%=op1%></td></tr>
    <tr><th>Operando 2: </th><td><%=op2%></td></tr>
    <tr><th>Resultado: </th><td><%=res%></td></tr>
    </table> <-A partir de aqui el código está duplicado.
```

```

</body>
</html>

```

Observa cómo con la directiva *page* indicamos que ante un error se ejecute la página **errores.jsp**.

2.8. Ahora crea la página **errores.jsp** con el siguiente código:

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Página de errores en JSP</title>
  </head>
  <body>
    <%@page isErrorPage="true"%>
    <h1>Página de errores en JSP</h1>
    <p>division.jsp ha reportado el siguiente error:
    <b><%=exception%></b>
  </p>
    <p>El error que ha ocurrido es: <pre>
    <% exception.printStackTrace(new java.io.PrintWriter(out)); %>
    </pre>
    </p>
  </body>
</html>

```

Observa cómo ahora se emplea la directiva *page* para indicar que esta es una página de errores, *isErrorPage*.

2.9. Ejecuta estas dos URLs `http://localhost:8080/JavaServerPages/division.jsp?op1=12&op2=6` y `http://localhost:8080/JavaServerPages/division.jsp?op1=12&op2=a` para comprobar cómo funciona si hay o no error, Figura 7.58.

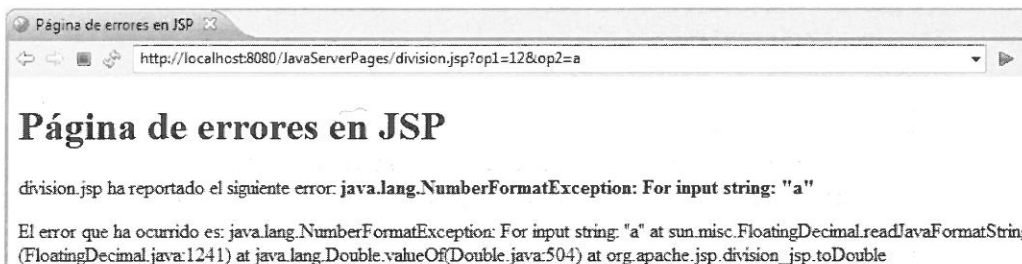


Figura 7.58: Página de errores

3. Ejemplo de JSP con acceso a datos.

- 3.1. Para finalizar vamos a realizar un ejemplo de JSP que accede a nuestra base de datos *TiendaLibros* con la misma funcionalidad que el *Servlet* que realizamos en la práctica de Acceso a Datos con JDBC.
- 3.2. Añade un fichero `consulta.jsp` a nuestro proyecto **Java Server Pages** con el siguiente código:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Consulta Libros</title>
</head>
<body>
    <h1 align="center">Tienda Libros</h1>
    <h3>Elige autor(s):</h3>
    <form method="get">
        <input type="checkbox" name="autor" value="Alvaro Garcia">Alvaro Garcia
        <input type="checkbox" name="autor" value="Aleksa Vukotic">Aleksa Vukotic
        <input type="checkbox" name="autor" value="Giulio Zambon">Giulio Zambon
        <input type="submit" value="consulta">
    </form>
<%
    String[] autores = request.getParameterValues("autor");
    if (autores != null) {
%>
<%@ page import = "java.sql.*" %>
<%
    //Paso 1: Cargar el driver JDBC.
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    //Paso 2: Conectarse a la Base de Datos utilizando la clase Connection
    String userName="root";
    String password="despliegue";
    String url="jdbc:mysql://localhost/TiendaLibros";
    Connection conn = DriverManager.getConnection(url, userName, password);
    //Paso 3: Crear sentencias SQL, utilizando objetos de tipo Statement
    Statement stmt = conn.createStatement();
    String sqlStr = "SELECT * FROM libros WHERE ";
    for (int i = 0; i < autores.length; i++) {
        sqlStr = sqlStr + "autor = '" + autores[i] + "' ";
        if (i != autores.length - 1) {
            sqlStr += "OR ";
        }
    }
    sqlStr += "AND cantidad > 0 ORDER BY precio DESC";
```

```

// para depuración
System.out.println("La consulta sql es " + sqlStr);
//Paso 4: Ejecutar las sentencias SQL a través de los objetos Statement
ResultSet rset = stmt.executeQuery(sqlStr);
%>
<hr>
<table border=1 cellpadding=5>
  <tr>
    <th>Autor</th>
    <th>Titulo</th>
    <th>Precio</th>
    <th>Cantidad</th>
  </tr>
<%
//Paso 5: Procesar el conjunto de registros resultante utilizando ResultSet
while (rset.next()) {
%>
  <tr>
    <td><%= rset.getString("autor") %></td>
    <td><%= rset.getString("titulo") %></td>
    <td><%= rset.getInt("precio") %></td>
    <td><%= rset.getInt("cantidad") %></td>
  </tr>
<%
}
%>
</table>
<%
//Cierre de recursos
rset.close();
stmt.close();
conn.close();
}
%>

</body>
</html>

```

Observa cómo el código es muy similar al que utilizamos en el *Servlet* para acceder a datos, la diferencia es que ahora el código *Java* está embebido dentro de código *HTML*.

- 3.3. Ejecuta la página **consulta.jsp** desde *Eclipse* y comprueba su correcto funcionamiento, Figura 7.59.

Consulta Libros

http://localhost:8080/JavaServerPages/consulta.jsp?autor=Alvaro+Garcia&autor=Aleksa+Vukotic

Tienda Libros

Elige autor(s):

☐ Alvaro Garcia ☐ Aleksa Vukotic ☐ Giulio Zambon

Autor	Titulo	Precio	Cantidad
Alvaro Garcia	Servicios de Red e Internet	25	100
Aleksa Vukotic	Apache Tomcat 7	22	22

Figura 7.59: JSP con acceso a datos