Parte VI

Java Server Page (JSP)

Java Server Page (JSP)

- Introducción.
  - Tecnología creada por SUN Microsystems.
  - Objetivo: Marco de referencia para la creación de contenido.
  - Es una especificación, no un producto:
    - Encima de TomCat (Jasper).
    - Parte de J2EE.

Java Server Page

- Conceptos.
  - Permitir la separación de los contenidos dinámico y estático.
  - Permitir la creación de contenido dinámico en forma fácil y flexible.
  - Embeber pequeños “servlets” dentro de una página que utiliza JSP (no siempre correcto)

Java Server Page

- Otras formas de flexibilizar:
  - JavaBeans (EJB) embebidos dentro de JSP.
  - Creación de elementos que parecen HTML, pero invocan a Java.
  - Solicitud a Servlet para trabajo back-end, y reenvío de información/contenido al cliente, para su presentación.
    - Separación de Lógica del Negocio y Presentación.

Java Server Page

- Sin embargo:
  - Se pierde control de la operación del servlet.
  - Recomendado:
    - Codificación a la medida mediante Servlets reales.
    - Programas que requieren únicamente manejo de lógica de presentación de contenido usando JSP.

Java Server Page

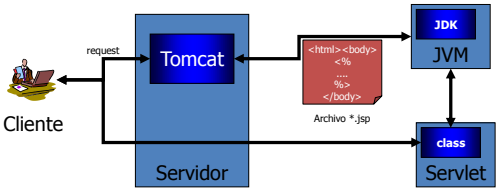
- Es un documento xHTML con código java incrustado
- Formalmente, una Página JSP es una implementación de la clase javax.servlet.Servlet que indica como crear un objeto respuesta (response) HttpServletResponse a partir de un objeto petición (request) HttpServletRequest

Java Server Page

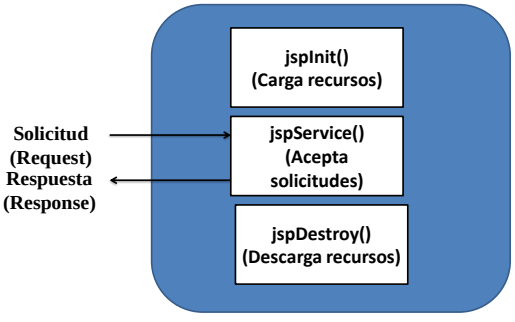
- Una JSP es convertido a un Servlet por el contenedor de Servlets

Modo de operación de JSP

- Cada “scriptlet” funciona como un servlet oculto, generado a partir de una especificación JSP.



Ciclo de vida



Uso de JSP

- Por medio de scriptlets.
  - <% ... %>
- Pueden contener variables predefinidas.
  - HttpServletRequest (petición)
  - HttpServletResponse (respuesta)
  - Javax.servlet.jsp.JspWriter.out
  - HttpSession (sesion)
  - ServletContext (aplicación)
  - Javax.Servlet.jsp.PageContext.pageContext

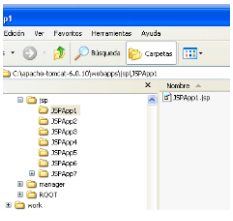
JSP en Acción

- Primer ejemplo (JSPApp1.jsp)

```
<body>
<h1>
<%
        if (request.getParameter("nombre") == null) {
                out.println("Hola, Mundo");
        } else {
                out.println("Hola, "+request.getParameter("nombre"));
        }
%>
</h1>
</body>
```

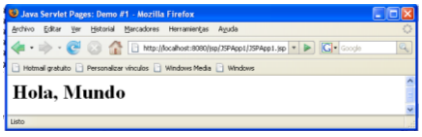
JSP en acción

- Copiar el archivo dentro del directorio webapps (jsp/JSPApp1)



JSP en acción

http://localhost:8080/jsp/JSPApp1/JSPApp1.jsp



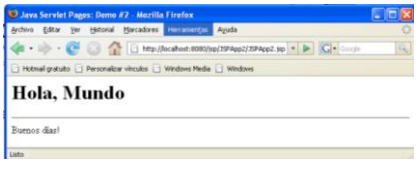
Estructura de un JSP

- Cada pieza de código propio de un servlet se denomina “scriptlet”.
  - Los scriptlets están rodeados de <% ...%>
    - Ejemplo (JSPApp2.jsp):

```
<% java.util.Date reloj = new java.util.Date(); %>
<% if (reloj.getHours() < 12) {    %>
      Buenos días!
<% } else if (reljo.getHours() < 17 {  %>
      Buenas tardes!
<% } else {   %>
      Buenas noches!
<% } %>
```

Ejecución JSPApp2

http://localhost:8080/jsp/JSPApp2/JSPApp2.jsp



Estructura de una JSP

- Adicionalmente, se puede agregar código usando:
  - Expresiones
  - Declaraciones

Estructura de una JSP

- Una expresión JSP se estructura:

  <%= ... %>
  - Ejemplo:

    <%= foo %>
    - Se evalúa la variable foo, y el resultado se incluye directamente en la página:
    - Usando: out.println(foo);

Estructura de una JSP

- Una declaración se estructura:

  <%! ... %>
  - Ejemplo:

    <%! int contador = 0; %>

    Ejemplos prácticos más adelante

Estructura de una JSP

- También se pueden aplicar Directivas, que tienen la estructura:
  <%@ ... %>
  - Ejemplo:
    <%@ page contentType="text/plain" %>
    <%@ page import="java.util.HashTable" %>
    <%@ page buffer="16kb" %>
    <%@ page session="false" %>
    <%@ page errorPage="/basurero.jsp" %>
  - ... entre otras...

Estructura de una JSP

- Una directiva es una orden que se ejecuta antes de que se comience a procesar la JSP. Ejemplo:

<%@ page language="java" contentType = "text/html" %>

- En este caso se indica que la página que se va a ejecutar está en lenguaje Java y que el contenido que se va a generar es de tipo text/html
  - Se pueden generar paginas en WML o XML

Estructura de un JSP

- Ejemplo JSPApp3.jsp

```
<!-- Ejemplos de scriptlets                                    -->

<%
    if (request.getParameter("nombre")== null) {
            out.println("Hola, Mundo");
    } else {
            out.println("Hola, "+request.getParameter("nombre"));
    }
%>
</h1>
<hr>

<!-- mezcla de html, HTML4.0 y JSP                    -->
<font face="Arial" color="blue" size="3">
Hoy es: <%= new java.util.Date().toString() %><br>
</font>
```

JSPApp3.jsp en acción

http://localhost:8080/jsp/JSPApp3/

Directivas (ErrorPages)

- Se puede *customizar* la apariencia de la porción del contenedor que manipule errores (JSPApp4.jsp):

```
<%@ page errorPage="/JSPApp4/JSPApp4Basurero.jsp" %>
…
…
<!-- cualquier condicion de error que se pueda generar -->
<%
        if (System.currentTimeMillis() > 0) {
                throw new Exception("oops");
        } // fin if
%>
```

Directivas (ErrorPages)

- Luego se hace referencia a la página (JSPApp4Basurero.jsp) que muestra el error (amigablemente).

```
<%@ page isErrorPage="true" %>
<%  response.setStatus(500); %>
<html><head>
<title>OOPS! <%= exception.getClass().getName() %></title>
</head>
<body>
<h1>¿Que crees?</h1><br>
<h3>¡Te equivocaste!</h3>
<br>
</body>
</html>
```

Sintaxis de manejo delicado

- Si escribimos un scriptlet en vez de una expresión (olvidamos el “=“), o si...
- Escribimos una directiva en vez de una expresión (colocamos “@” en vez de “=“)...
- OJO: El compilador está actuando sobre un servlet generado, no sobre el archivo JSP!
- Puede causar muchos errores por efectos secundarios.

Client-Side vs. Server-Side

- JSP está hecho para funcionar del lado del servidor.
  - No habrá manera que el cliente ejecute directamente una pieza de código dentro de un JSP.
  - Hacerlo mediante un truco de hacking puede abrir agujeros de seguridad.
    - Se recomienda no hacerlo.

Uso de Java Beans en JSP

- Java Beans reutilizables dentro de una página JSP por medio del elemento:

```
<jsp:useBean id="reloj" class="java.util.Date" />
```

  - Ejemplo JSPApp5.jsp

JSPApp6.jsp en acción

http://localhost:8080/jsp/JSPApp6/JSPApp6.jsp

Información de la sesión

- Podemos obtener información en forma análoga a como la obtuvimos programando Servlets.
  - Ver JSPApp6.jsp

```
<li>Método HTTP:        <%= request.getMethod() %>
<li>URI Procedencia:    <%= request.getRequestURI() %>
<li>Protocolo:          <%= request.getProtocol() %>
<li>Ruta:               <%= request.getServletPath() %>
<li>Query:              <%= request.getQueryString() %>
```