Programación Básica en Internet

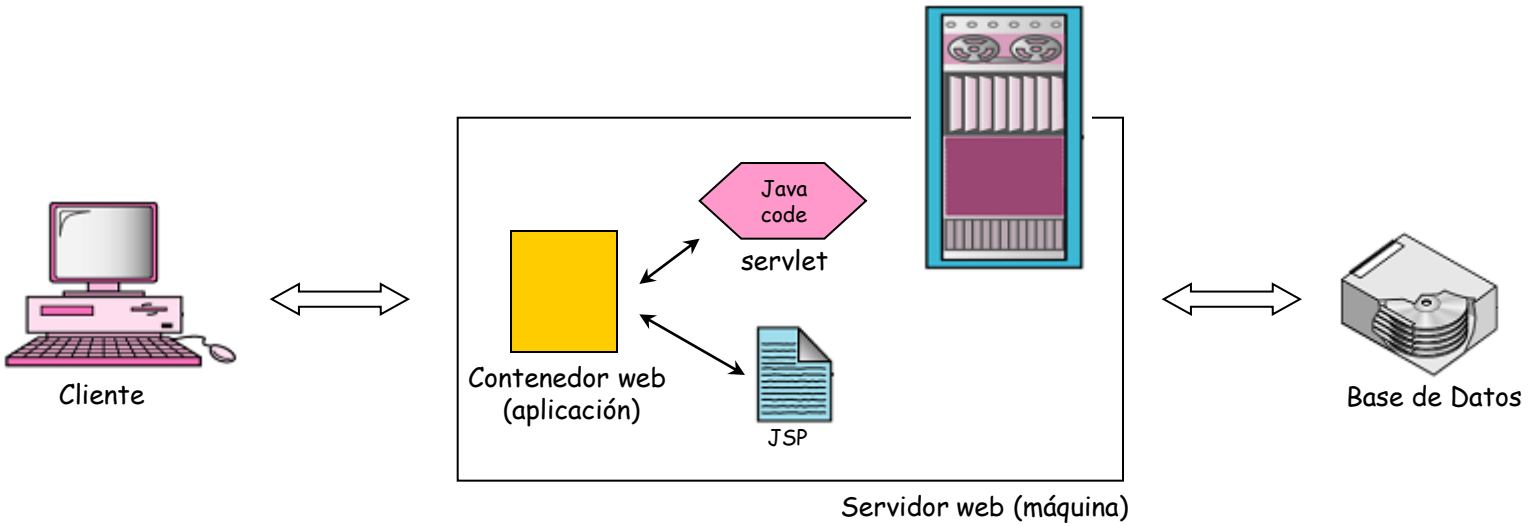
JSP. Introducción.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

Java Server Pages (JSP)

MVC: Modelo Vista Controlador



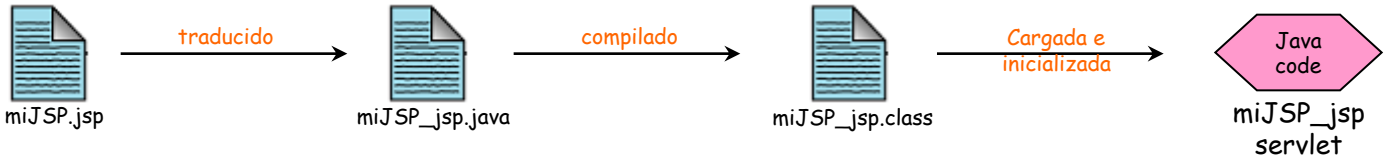
Java Server Pages (JSP)

JSP.- son páginas HTML que contienen líneas de código adicional que sirven para ejecutar aplicaciones que dan lugar a contenido dinámico. Dichas aplicaciones corren en el servidor.

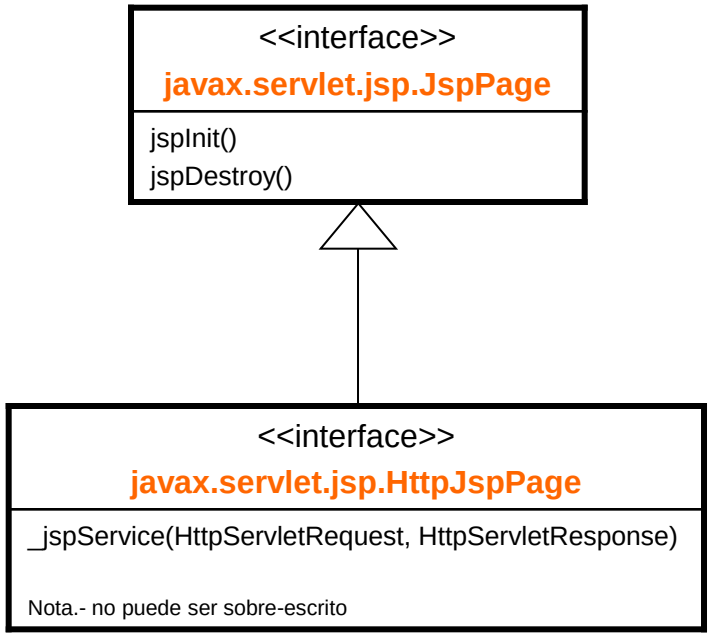
Las JSP permiten separar fácilmente la lógica de negocio de la presentación de resultados (desacoplar), encapsulando la primera → Los desarrolladores de la lógica de negocio no tienen que ser expertos en HTML y viceversa.

Las JSP son traducidas de forma automática a *servlets*, siendo a partir de ese momento ejecutadas como tales. En caso de modificaciones de la página, se realizará de nuevo una traducción automática.

Java Server Pages (JSP)



Java Server Pages (JSP)



Java Server Pages (JSP)

JSP

```
<html>
  <body>
    <% int suma=0; %>
    El valor actual es:
    <%= ++suma %>
  </body>
</html>
```

Servlet

```
public class contador_jsp extends HttpServlet {
  public void _jspService(HttpServletRequest request,
                HttpServletResponse response)
                throws java.io.IOException, ServletException {

    PrintWriter out= response.getWriter();
    response.setContentType("text/html");
    out.write("<html><body>");
    int suma= 0;
    out.write("El valor actual es:");
    out.print(++suma);
    out.write("<body><html>");
  }
}
```

TomcatHomeDir/work/Catalina/yourServerName/yourWebAppName/org/apache/jsp

Java Server Pages (JSP)

Una página JSP tiene una apariencia similar a una página HTML, donde se han introducido etiquetas encapsuladas entre < ….. >.

Dentro de dichas etiquetas estarían los diferentes componentes que se pueden emplear:

- Comentarios.- <%--.....--%>
- Declaraciones.- <%!.....%>
- Expresiones.- <%=.....%>
- Scriplets.- <%.....%>
- Directivas.- <%@.....%>

Java Server Pages (JSP)

Comentarios.- sirven para hacer más comprensible el código escrito. Su contenido no se transfiere al cliente.

Ej.-

```
<%-- Este es un comentario JSP --%>
```

Declaraciones.- utilizadas para definir variables y métodos que se utilizarán posteriormente.

Ej.-

```
<%! int contador= 0; %>
<%! int suma(int a, int b){
        _____
        _____
    }
%>
```

En la traducción, aparecerán fuera del cuerpo del método *_jspService*

Java Server Pages (JSP)

Expresiones.- utilizadas para evaluar una expresión escrita en Java.

Ej.-

```
<%= clock.getYear() %>
<%= Math.random() %>
```

Scriplets.- permiten la inclusión de cualquier código Java entre los delimitadores.

Ej.-

```
<% if (time==10) %>
<% int contador= 0; %>
```

Java Server Pages (JSP)

Directivas.- instrucciones que son procesadas por el motor JSP cuando la página es compilada en un servlet. No devuelven nada, aunque afectan a la estructura del servlet resultante de la traducción.

Ej.-

i) Definiciones a nivel de página:

```
<%@ page languaje== “java” imports== “java.util.*” %>
```

ii) Inserciones de datos desde ficheros externos:

```
<%@ include file== “pbi.html”%>
```

iii) Utilización de elementos definidos en librería propia:

```
<%@ taglib uri= http://www.usc.es/pbi/jsp_lib prefix=“fun”%>
        ……
        <fun:acción_1 ……>
```

Java Server Pages (JSP)

Directiva page.- define atributos que se aplican a toda la página JSP. Típicamente se localizan al comienzo de la página.

<%@ page *lista de atributos* %>

donde lista de atributos incluye:

- language= *"scriptingLanguage"* → java por defecto
- extends= *"className"* → paquete de la clase extendida
- import= *"importList"* → permite especificar lista de paquetes java importados
- session= *"**true**/false"* → permite o no sesiones
- buffer= *"none/**default**/size kb"* → tamaño del buffer de out.
- autoFlush= *"**true**/false"* → si true buffer salida debe descargarse automáticamente.
- isThreadSafe= *"**true**/false"* → si true, contenedor puede despachar varias hebras simultáneamente
- info= *"info_text"* → texto de información de lo que hace la página
- errorPage= *"error_url"* → url de la página a la que se accederá en caso de error
- iserrorPage= *"true/**false**"* → si true página de error tiene acceso al objeto Exception
- contentType= *"ctinfo"* → indica tipo MIME utilizado en página de respuesta (text/html por defecto)
- pageEncoding= *"peinfo"* → indica tipo de codificación de página de respuesta, por defecto
- isELIgnored= *"true/false"* → si true, expresiones de EL ignoradas

Java Server Pages (JSP)

Directiva include.- permite la inclusión de ficheros (.html, .jsp, .xml, o de texto) en una página JSP. Esto permite dividir la página en trozos más manejables.

<%@ include file= “*filename*” %>

Java Server Pages (JSP)

Directiva taglib.- Indica la biblioteca de acciones personalizadas que se puede utilizar. Dos formas de invocación:

i) Mediante clase *Java* y descriptor de fichero *.tld*

```
<%@ taglib tagdir= "/WEB-INF/tags" prefix= "func"%>

……..
<func: accion_1 …….>
```

ii) Mediante fichero *.jar*

```
<%@ taglib uri= "http://...../mibiblioteca" prefix= "func"%>

……..
<func: accion_1 …….>
```

Java Server Pages (JSP)

Objetos implícitos.- Conjunto de objetos declarados e inicializados que pueden ser utilizados por el contenedor en el método *_jspService()* y que serán utilizados para su ejecución.

El acceso a los objetos implícitos se hace a través del objeto implícito *pageContext* de la clase *PageContext.* Se obtiene, a través del método *getPageContext() de la clase JspFactory*.

pageContext encapsula el resto de objetos implícitos, que serán accedidos a través de los métodos correspondientes.

Java Server Pages (JSP)

Objetos implícitos.- Conjunto de objetos declarados e inicializados que pueden ser utilizados por el contenedor en el método _jspService() y que serán utilizados para su ejecución.

Objeto	Clase	Método	Uso
out	JspWriter	getOut()	Almacenamiento de la respuesta que será enviada al cliente que solicito la página
request	HttpServletRequest	getRequest()	Encapsulamiento de los datos enviados por el cliente al servidor
response	HttpServletResponse	getResponse()	Encapsulamiento de los datos enviados por el servidor al cliente
session	HttpSession	getSession()	Identificación de usuarios que visitan las páginas (seguimiento de sesiones)
application	ServletContext	getServletContext()	Definición del conjunto de métodos que todo *servlet* utiliza para su comunicación con el contenedor
config	ServletConfig	getServletConfig()	Pasar información al *servlet* durante la inicialización
exception	JspException	getException()	Contener información acerca de cualquier excepción lanzada desde la página JSP
page	Object	getPage()	Representación del *servlet* obtenido de la página JSP

Java Server Pages (JSP)

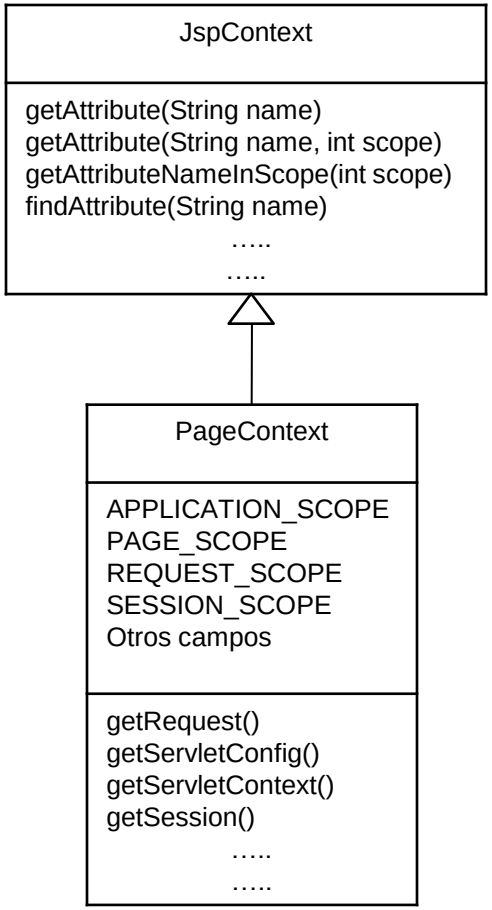
Atributos.- Su papel es análogo al visto en los *servlets*. En JSP se definirá un nuevo ámbito de uso (*page*), no presente en los *servlets*.

Se manipulan mediante los métodos correspondientes: getAttribute() y setAttribute() del objeto correspondiente:

Ámbito	Servlet	JSP (mediante objetos implícitos)
Application	getServletContext.getAttribute()	Application.getAttribute()
Request	request.getAttribute()	request.getAttribute()
Session	Request.getSession().getAttribute()	session.getAttribute()
Page	No existe	pageContext.getAttribute()

Java Server Pages (JSP)

Otra forma alternativa de manipulación es mediante el uso de los métodos correspondientes del objeto implícito *pageContext*.



Java Server Pages (JSP)

Otra forma alternativa de manipulación es mediante el uso de los métodos correspondientes del objeto implícito *pageContext.*

Ej.-

```
<%= session.getAttribute("email") %>

<%= pageContext.getAttribute("email", PageContext.SESSION_SCOPE) %>
```

Ej.-

```
<%= pageContext.findAttribute("email") %>
```

MVC: Modelo Vista Controlador

HTML

```
<form action="fijaAtr">
        username:  <input type="text" name="userName">
        ....
</form>
```

servlet

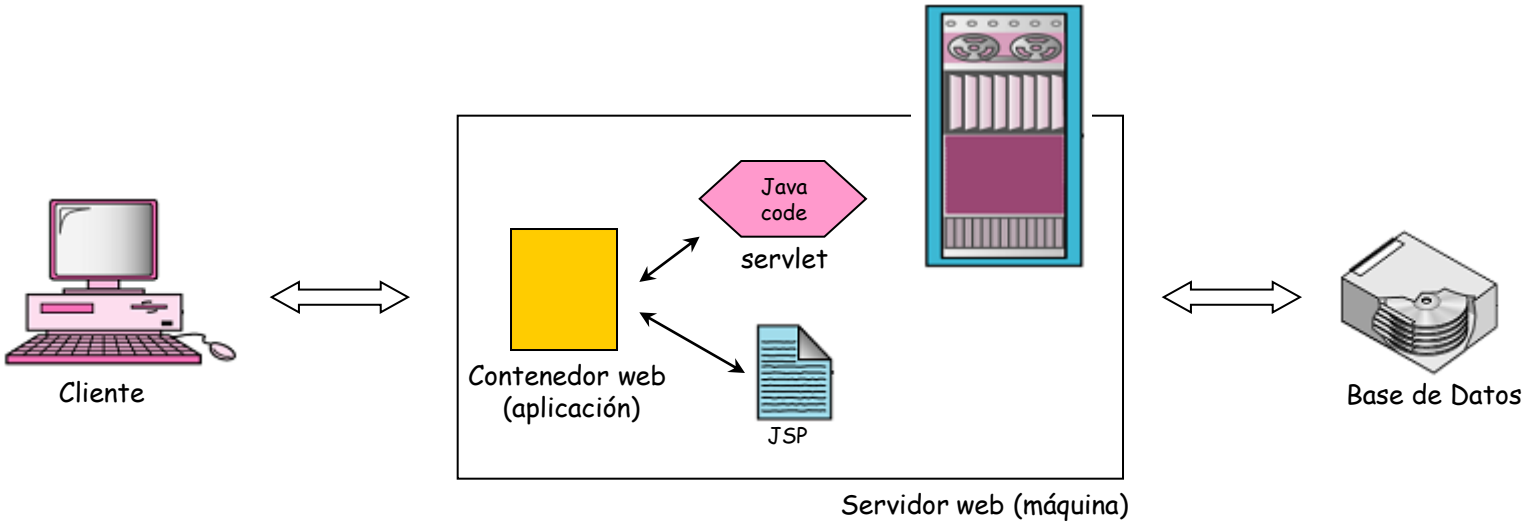
```
String name= request.getParameter("userName");

request.setAttribute("nombre", name);

RequestDispatcher view= request.getRequestDispatcher("vista_1.jsp");
view.forward(request, response);
```

JSP

```
El nombre del usuario es:
        <%= request.getAttribute("nombre") %>
```

JDBC

MVC: Modelo Vista Controlador



RequestDispatcher view= request.getRequestDispatcher(“mi_jspPage.jsp”);

view.forward(request, response);

<%= request.getAttribute("salida")%>