

Bootcamp Java Developer

Fase 1 - Java Analyst
Módulo 11

Java Server Pages (JSP)

Java Server Pages (JSP)

Es la solución de Java para el desarrollo Web dinámico. Son documentos que poseen elementos (*tags*) HTML y, además, la capacidad de incluir fragmentos de **código Java** a través de *tags* propios de esta tecnología. Estos fragmentos de códigos también son llamados *Scriptlets*.

Un JSP se convierte en una clase **Servlet** y se ejecuta en el servidor. El código de Java incluido en el documento JSP pasa a formar parte de esta nueva clase y genera una **respuesta HTML del lado del cliente**.

Recordemos que los clientes Web (navegadores) no ejecutan otro tipo de código que no sea HTML, CSS y JavaScript

Objetos Implícitos en JSP

JSP proporciona objetos implícitos en cada documento para ser utilizados como sea necesario. Adicionalmente, se pueden crear objetos propios o variables de forma explícita.



Declaración	Descripción
request	Objeto de la clase <code>HttpServletRequest</code> .
response	Objeto de la clase <code>HttpServletResponse</code> .
session	Objeto de la clase <code>HttpSession</code> que está asociada al <code>request</code> , si es que trabajamos con <code>sessions</code> .
out	Objeto de la clase <code>PrintWriter</code> usada para imprimir directamente en la página resultante.

Elementos JSP

Para poder incorporar código Java en un documento HTML, JSP y/o XML se debe **encerrar entre llaves angulares y los símbolos de porcentaje “<% %>”**.

Los elementos JSP que veremos en los siguientes slides, son:

- Comentarios.
- Expresiones.
- Scriptlets.
- Declaraciones.
- Directivas.



<% %>

Comentarios

Ayudan a saber más sobre el código.

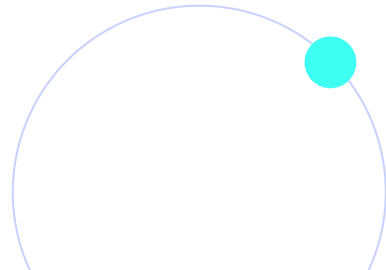


```
<!-- Comentario de una linea en JSP --%>  
<%  
    // Comentarios de una linea en Java  
  
    /*  
    Comentarios de varias lineas en Java  
    */  
%>
```

Expresiones

Se usan para **enviar valores directamente a la página web** resultante del JSP.

```
<%-- Podemos utilizar explicitas o implícitas --%>  
<%=request.getParameter("nombre") %>  
  
<%=i %>
```



Scriptlets

Sirven para usar cualquier **bloque** o **sentencia de Java**.



```
<%  
    for (int i = 0; i < 10; i++){  
        out.print(i);  
    }  
%>  
  
<!-- Podemos intercalar codigo Java y HTML --%>  
  
<%  
    for (int i = 0; i < 10; i++){  
%>  
        <h1><%=i %></h1>  
%>  
    }  
%>
```


Declaraciones

Sirven para **definir variables y métodos** que se utilizarán en *scriptles* y expresiones.



```
<%!  
    private int contadorIngresos=0;  
  
    public String breakHTML(){  
        return "<br>";  
    }  
%>
```

Directivas

Son **etiquetas** a partir de las cuales se genera información que puede usar el motor JSP.

No producen una salida visible al usuario sino que configuran cómo se ejecutará la página JSP.

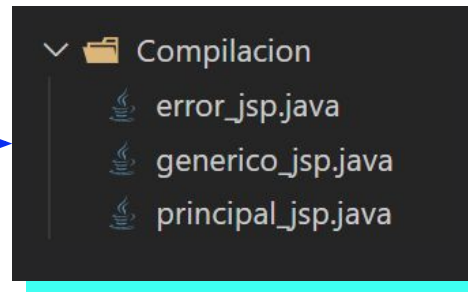
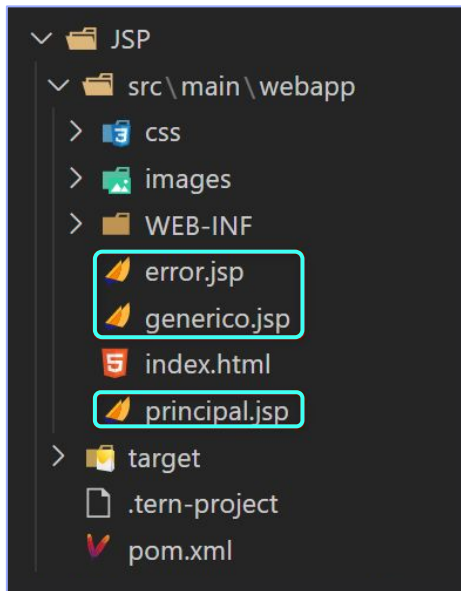
Tipo	Método	Descripción
include	<code><%@include file="cabecera.html" %></code>	Incluye el contenido de un fichero en la página mediante el atributo <code>file</code> .
taglib	<code><%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%></code>	Importa bibliotecas de etiquetas (<i>Tag Libraries</i>).
page import	<code><%@page import="java.util.Map"%></code>	Importa clases y paquetes Java para ser utilizadas dentro del fichero JSP.
page session	<code><%@page session="true"%></code>	Especifica si utiliza los datos contenidos en sesión; por defecto <code>"true"</code> .
page errorPage	<code><%@page errorPage="error.jsp"%></code>	Especifica la ruta de la página de error que será invocada en caso de producirse una excepción durante la ejecución de este fichero JSP.

JSP Servlets

Como habíamos comentado al principio, al invocar todos los documentos JSP, el servidor o contenedor Web creará **un Servlet por cada documento que generará la lógica en Java** y que terminará mostrando el **resultado en HTML**.

Veamos el ejemplo de la próxima pantalla.





RequestDispatcher

RequestDispatcher

Esta interfaz proporciona la posibilidad de **enviar o incluir objetos a otro recurso** (HTML, *Servlet* o JSP).

Tipo	Método	Descripción
void	<code>forward(ServletRequest request, ServletResponse response)</code>	Reenvía una solicitud de un <i>Servlet</i> a otro recurso.
void	<code>include(ServletRequest request, ServletResponse response)</code>	Incluye el contenido de un recurso.

```
request.setAttribute("nombreObjeto", objeto);  
  
RequestDispatcher dispatcher = request.getRequestDispatcher("documento.ext");  
dispatcher.forward(request, response);
```



**¡Sigamos
trabajando!**