



Universidad de
Oviedo



JAVA SERVER PAGES (JSP)

Enol García González
Universidad de Oviedo
10 de noviembre de 2025

CONTENIDOS

- 1 Introducción
- 2 Ventajas de JSP
- 3 Elementos de JSP
 - Scripting
 - Directivas
 - Acciones

¿QUÉ ES JSP?

Java Server Pages

Una tecnología construida sobre servlets que facilita la creación de páginas webs dinámicas al combinar código HTML con elementos especiales.

Su principal función es resolver el problema de salida de los servlets.

¿CUÁL ES EL PROBLEMA DE SALIDA?

```
public class SaludoServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println(<html><head><title>Saludo personalizado</title></head><body>);
        Date hora = new Date();
        if (hora.getHours() < 12) {
            out.println("<h1>Buenos_días</h1>");
        } else if (hora.getHours() < 21){
            out.println("<h1>Buenas_tardes</h1>");
        } else {
            out.println("<h1>Buenas_noches</h1>");
        }
        out.println("<body></html>")
    }
}
```

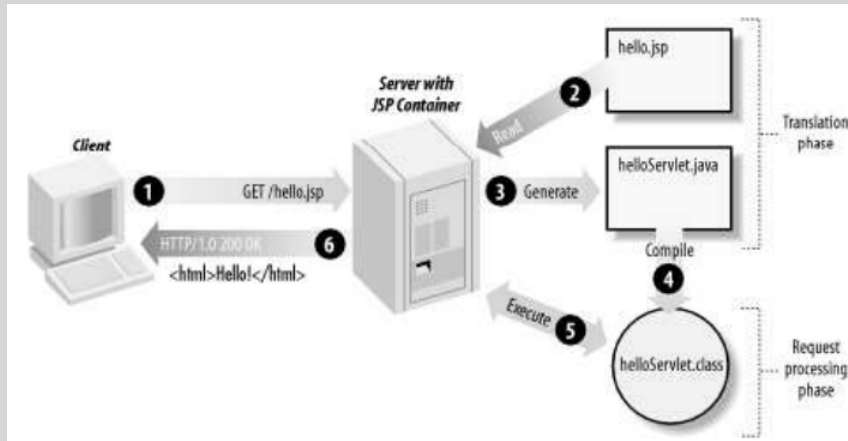
¿CÓMO SE HACE CON JSP?

```
<html>
  <head>
    <title>Saludo personalizado con JSP</title>
  </head>
  <body>
    <% java.util.Date hora = new java.util.Date(); %>
    <% if (hora.getHours() < 12) { %>
      <h1>Buenos días</h1>
    <% } else if (hora.getHours() < 21) { %>
      <h1>Buenas tardes</h1>
    <% } else { %>
      <h1>Buenas noches</h1>
    <% } %>
  </body>
</html>
```

VENTAJAS

- Aumenta la legibilidad en paginas grandes.
- Permiten la inclusión de otros ficheros pudiendo crear también código HTML reutilizable.
- Bien utilizado es una herramienta muy potente para separar lógica y presentación.

¿CÓMO FUNCIONA?



ELEMENTOS DE JSP

- **Scripting.** Permiten insertar código que será ejecutado en el momento de la petición
- **Directivas.** Permiten especificar información acerca de la página que permanece constantes para todas las peticiones.
- **Acciones.** Determinan acciones que ejecutar sobre la información que se requiere en la petición.

Y además, tienen acceso a los mismos objetos predefinidos que tenían los servlets: *request*, *response*, *pageContext*, *session*, *application*, *out*, *config* y *page*.

SCRIPTING

- `<% ... %>`. **Scriptlet**. Encierra código Java.
- `<%= ... %>`. **Expresión**. Accede a valores y los imprime en la web.
- `<%! ... %>`. **Declaración**. Declara variables y métodos.
- `<%-- ... -- %>`. **Comentario**.

EJEMPLO SCRIPTING

```
<%!  
private java.util.Date hora = new java.util.Date();
```

```
String getSaludo(){  
    if (hora.getHours() < 13)  
        return "Buenos_días";  
    if (hora.getHours() < 20)  
        return "Buenas_tardes";  
    return "Buenas_noches";  
}
```

```
<html>  
<head><title>Saludo con JSP</head>  
<body>  
<%= getSaludo() %>  
</body>  
</html>
```

DIRECTIVAS

- `<%@ page ... %>`. Permite importar clases de java, especificar el tipo de respuesta (por defecto es "text/html"), etc.
- `<%@ include ... %>`. Permite incluir otros ficheros antes de que la página sea traducida a servlet.
- `<%@ taglib ... %>`. Declara una biblioteca de etiquetas con acciones personalizadas.

Ejemplos

```
<%@ taglib uri = "http://www.example.com/custlib" prefix = "mytag" %>
```

Puede permitir tener una etiqueta como `<mytag:hello/>` que estaría desarrollada con anterioridad

ACCIONES

- `<jsp:useBean>`. Permite utilizar un Bean desde la página
- `<jsp:getProperty>`. Obtiene la propiedad de un Bean
- `<jsp:setProperty>`. Establece la propiedad de un Bean
- `<jsp:include>`. Incluye la respuesta de un servlet o página JSP
- `<jsp:forward>`. Redirige a otro servlet o página JSP
- `<jsp:param>`. Envía un parámetro a una petición redirigida.

EJEMPLO DE UN BEAN

```
public class DateBean {  
    private Date hora = new Date();  
  
    public int getHour() {  
        return date.getHours();  
    }  
  
    public String getSaludo(){  
        if (hora.getHours() < 13)  
            return "Buenos_días";  
        if (hora.getHours() < 20)  
            return "Buenas_tardes";  
        return "Buenas_noches";  
    }  
}
```

EJEMPLO DE USO DE UN BEAN

```
<jsp:usebean id="date" class="uo.tew.beans.DateBean" />

<html>
<head>
  <title>Saludos con JSP</title>
</head>
<body>
  Son las <%= date.getHour() %>. Así que saludo con: <%=date.getSaludo() %>.
</body>
</html>
```