



Objetivos de la sesión:

- *Ampliar la información que se muestra de los alumnos*
- **Formatear el listado y el alta de alumnos** utilizando Bootstrap
- **Utilizar Fragmentos JSPF** para mejorar el mantenimiento y posteriores modificaciones de la webapp.



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - joseramon.profesor@gmail.com

Ampliando y formateando datos de alumnos:

Vamos a ampliar los datos de los alumnos y formatearlos para conseguir la siguiente apariencia:

Alumnos x +

localhost:8080/list-alumno.do

Simarro Home Alumnos DWES Logout

Listado de alumnos:

Dni	Nombre	Edad	Ciclo	Curso	Acción
11111111A	Jose	21	DAM	1	Borrar
22222222B	Pedro	32	DAW	2	Borrar
33333333C	Juan	23	ASIR	1	Borrar
44444444D	Maria	18	ASIR	1	Borrar

Añadir alumno

Alumnos x +

localhost:8080/add-alumno.do

Simarro Home Alumnos DWES Logout

Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:

Nombre:

55555555F

Fernando

Edad:

Ciclo:

Curso:

55

DAM

1

Añadir

DWES: Desarrollo Web en Entorno Servidor - profesor: joseramon.profesor@gmail.com



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com

Ampliando y formateando datos de alumnos:

- Para poder ampliar y formatear los datos que guardamos de los alumnos vamos a seguir los siguientes pasos:

- 1º Independizar el alta de los alumnos del listado
- 2º Añadir atributos nuevos a "Alumno"
- 3º Formatear listado con Bootstrap
- 4º Formatear el formulario con Bootstrap



UD 1: Introducción a los lenguajes de servidor

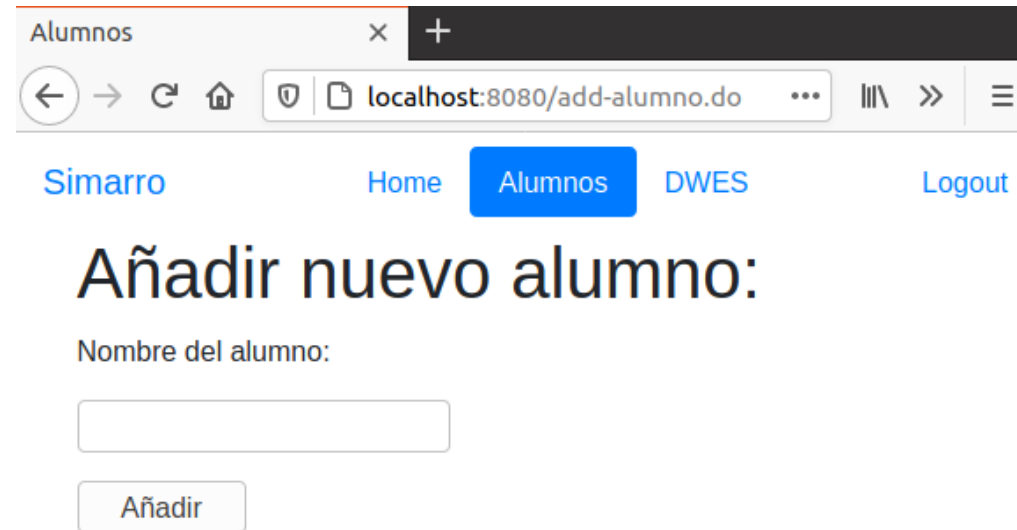
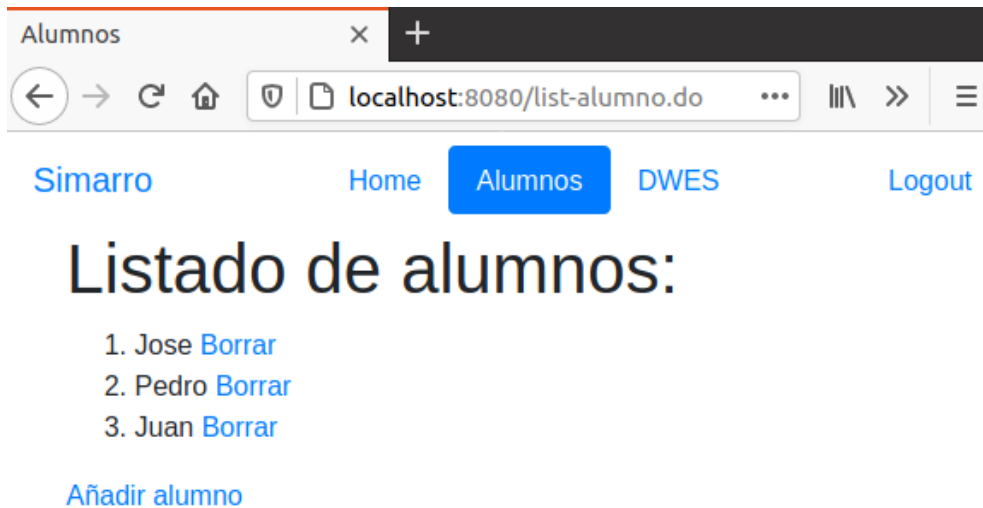
4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - joseramon.profesor@gmail.com

... continuación **Ampliando y formateando datos de alumnos:**

1º Independizar el alta de los alumnos del listado: En la práctica anterior creamos el servlet AddAlumnoServlet, ahora vamos a crear una página add-alumno.jsp que pedirá los datos del nuevo alumno y reenviará la solicitud a AddAlumnoServlet. Hay que modificar alumnos.jsp para que pueda llamar a la nueva página para añadir alumnos en vez de añadirlos directamente. Sería conveniente renombrar el listado de alumnos en el jsp como list-alumno.jsp . Antes de pasar a la siguiente hoja comprueba que todo funciona después de realizar los cambios anteriores:





UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com

... continuación **Ampliando y formateando datos de alumnos:**

2º Añadir atributos nuevos a Alumno:

Vamos a añadir a la clase “Alumno” nuevos atributos. Debemos añadir los atributos “dni” (String), “edad” (numérico), “ciclo” (String) y curso (numérico).

Este cambio arrastrará muchos otros cambios que en la página siguiente nombraremos.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/add-alumno.do'. The page has a navigation bar with links: 'Simarro', 'Home', 'Alumnos' (highlighted in blue), 'DWES', and 'Logout'. The main heading is 'Nuevo alumno:'. Below it, a subheading says 'Introduzca los datos del nuevo alumno:'. The form contains the following fields: 'Dni:' with an empty text input field; 'Nombre:' with a text input field containing 'Maria'; 'Edad:' with a numeric input field containing '23'; 'Módulo:' with a text input field containing 'DAW'; and 'Curso:' with a numeric input field containing '2'. At the bottom of the form is a button labeled 'Añadir'. The footer of the page contains the text 'DWES: Desarrollo Web en Entorno Servidor - profesor: joseramon.profesor@gmail.com'.



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - joseramon.profesor@gmail.com

... continuación **Ampliando y formateando datos de alumnos:**

... continuación **2º Añadir atributos nuevos a Alumno:**

Deberemos de realizar los cambios necesarios para que equals se indentifique por el “dni” y ya no se identifique por el nombre. Si cambiamos la clase “Alumno” deberemos de actualizar la clase “AlumnoService”. También hay que modificar add-alumno.jsp, AddAlumnoServlet y list-alumno.jsp para que se incorporen los nuevos datos y en el caso de list-alumno.jsp, para borrar un alumno se pasará el “dni” , no el nombre. Este cambio también implica modificar DelAlumnoServlet .

Para mejorar la introducción de datos todos los campos del alumno en “add-alumno.jsp” serán necesarios que estén llenos para que se pueda enviar el formulario (atributo “required”). Este cambio lo realizaremos más adelante.

ID	Nombre	Edad	Curso	Acción
1. 11111111A	Jose	21	DAM 1	Borrar
2. 22222222B	Pedro	32	DAW 2	Borrar
3. 33333333C	Juan	23	ASIR 1	Borrar
4. 44444444D	Maria	23	2	Borrar

[Añadir alumno](#)



... continuación **Ampliando y formateando datos de alumnos:**

3º Formatear listado con Bootstrap:

Si utilizamos el siguiente código podemos ver como cambia la presentación del listado:

```
<div class="container">
  <h1> Listado de alumnos:</h1>
  <table class="table table-striped">
    <thead>
      <th> Nombre</th>
      <th> Edad</th>
      <th> Acción</th>
    </thead>
    <tbody>
      <tr>
        <td> Nombre 1</td>
        <td> edad 1 </td>
        <td> <a class="btn btn-danger" href="#">Borrar</a></td>
      </tr>
      <tr>
        <td> Nombre 2</td>
        <td> edad 2 </td>
        <td> <a class="btn btn-danger" href="#">Borrar</a></td>
      </tr>
    </tbody>
  </table>
  <p> <a class="btn btn-success" href="#">Añadir alumno</a></p>
```



Simarro

Home

Alumnos

DWES

Logout

Listado de alumnos:

Nombre	edad	Acción
Nombre 1	edad 1	Borrar
Nombre 2	edad 2	Borrar

[Añadir alumno](#)

Modifica list-alumnos.jsp para que tenga la misma apariencia, pero con todos los datos de los alumnos.



... continuación **Ampliando y formateando datos de alumnos:**

3º Formatear formulario con Bootstrap:

Si **modificas** el formulario de `add-alumno.jsp` con el siguiente código podemos ver como cambia la presentación del alta de un nuevo alumno:

```
<form action="add-alumno.do" method="post">
  <div class="form-row">
    <div class="col">
      <label>Dni:</label>
      <input type="text" id="dni" name="dni" required
        class="form-control" minlength="9"/>
    </div>
    <div class="col">
      <label>Nombre:</label>
      <input type="text" id="nombre" name="nombre" required
        class="form-control" minlength="5"/>
    </div>
  </div>
  <div class="form-row">
    <div class="col">
      <label>Edad:</label>
      <input type="number" id="edad" name="edad" required
        class="form-control" min="18" max="100"/>
    </div>
    <div class="col">
      <label>Ciclo:</label>
      <input type="text" id="ciclo" name="ciclo" required
        class="form-control" minlength="3"/>
    </div>
    <div class="col">
      <label>Curso:</label>
      <input type="number" id="curso" name="curso"
        class="form-control" required min="1" max="2"/>
    </div>
  </div>
  <br><input type="submit" value="Añadir" class="btn btn-success">
</form>
```



Simarro

Home

Alumnos

DWES

Logout

Nuevo alumno:

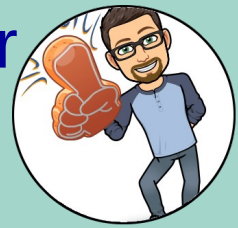
Introduzca los datos del nuevo alumno:

Dni:	Nombre:	
<input type="text"/>	<input type="text"/>	
Edad:	Ciclo:	Curso:
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Añadir"/>		



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com

Ahora que ya tenemos una funcionalidad básica (listar, añadir y borrar) en nuestra aplicación con una estética decente y controlamos que el usuario este logeado podemos ir pensando en **optimizaciones y mejoras**.

Una de las posibles mejoras que podemos hacer es **quitar la redundancia existente en las páginas JSP**. Todas las páginas JSP tienen la misma cabecera, barra de navegación y pie de página. La redundancia implica una dificultad en el mantenimiento. Cuando queramos realizar alguna modificación, deberemos acordarnos de hacerlo en todas las páginas y eso es un fallo de diseño.



¿Como podemos quitar la cabecera, barras de navegación y pie de todas las páginas de nuestra web para centralizarlo y facilitar las futuras modificaciones?



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com

JSPF:

- Las páginas **JSPF (Java Server Pages Fragment)** son fragmentos de **páginas JSP**.
- Son un paso más en la **refactorización de las páginas JSP** para conseguir código no redundante. Si una **porción de código se utiliza mucho en diferentes páginas y es siempre el mismo** código podemos **extraerlo a un fichero JSPF** e incluirse dicho fichero en las páginas JSP que lo utilizan.
- Las páginas JSPF por convención suelen guardarse en **/WEB-INF/jspf**. A diferencia de las páginas jsp, no puedes contruir una URL para llamar directamente a ese fichero, pero si que puedes utilizarlo para incluirlo en una página JSP.

Crea la carpeta "jspf" dentro de /WEB-INF. En la siguiente página tendrás que crear header.jspf, footer.jspf y menuSuperior.jspf



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Jose Ramon.profesor@gmail.com

... continuación **JSPF**:

A continuación vamos a crear una página JSPF para guardar la cabecera de la página JSP (header.jspf), otra página JSPF para guardar el pie (footer.jspf) y otra página para guardar la barra de navegación superior(menuPrincipal.jspf).

Para conseguirlo abre add-alumno.jsp y copia cada fragmento de código a la página JSPF correspondiente. A continuación puede ver un ejemplo de JSPF y como llamar a los fragmentos. Como se puede observar pueden aparecer warnings en las páginas jsp que llaman a jspf's o en las propias páginas jspf:

```
header.jspf
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <link href="webjars/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
6 <title>${pagina.getTitulo()}</title>
7 <style>
8     .footer {
9         position: absolute;
10        bottom: 0;
11        width: 100%;
12        height: 60px;
13        background-color: #f5f5f5;
14    }
15 </style>
16 </head>
17 <body>
```

```
add-alumno.jsp
1 <%@ include file="../jspf/header.jspf"%>
2 <%@ include file="../jspf/menuSuperior.jspf"%>
3
4 <div class="container">
5     <h1>Nuevo alumno:</h1>
6     <p> Introduzca los datos del nuevo alumno:</p>
7
8     <form action="add-alumno.do" method="post">
9
10 </div>
11
12 <%@ include file="../jspf/footer.jspf"%>
```

Modifica list-alumno.jsp y add-alumno.jsp para que utilicen header.jspf, footer.jspf y menuSuperior.jspf (la barra de navegación nav).



... continuación **JSPF**:

Modifica login.jsp para utilizar los 3 ficheros JSPF anteriores.

Alumnos: x +

← → ↻ 🏠 🛡️ 📄 localhost:8080 ... 📑 📄 👤 ☰

[Simarro](#) [Home](#) [Alumnos](#) [DWES](#) [Logout](#)

Introduzca su nombre:

Introduzca su contraseña:

Login

DWES: Desarrollo Web en Entorno Servidor - profesor: joseramon.profesor@gmail.com



¿Con que problema nos encontramos? ¿Como podemos solucionarlo?



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - joseramon.profesor@gmail.com

... continuación **JSPF**:

En el menú superior no nos indica que estamos en “Home”, sino que marca que estamos en “Alumnos”. Adicionalmente tenemos que el título (title) de la página web no es “Login”, sino “Alumnos”.

Deberás añadir una clase “Pagina” :

```
Pagina.java
1 package org.profesor.joseramon.primer_jee.pagina;
2
3 public class Pagina {
4     String titulo;
5     String paginaActiva;
6
7     public Pagina(String titulo, String paginaActiva) {}
12    public String getTitulo() {}
15    public void setTitulo(String titulo) {}
18    public String getPaginaActiva() {}
21    public void setPaginaActiva(String paginaActiva) {}
24
25    public String getStrBootstrapActiva(String pagina) {
26        if (paginaActiva.equals(pagina))
27            return "active";
28        else
29            return "";
30    }
31 }
```




... continuación **JSPF**:

Deberás crear una pagina con los datos adecuados y añadirlo a la sesión en cada método de los Servlets que hagan un `request.getRequestDispatcher()` para que cuando visualice el jsp se marque correctamente el menu en azul.

Por ejemplo en el listado de alumnos:

```
1 package org.profesor.joseramon.primer_je.e.alumno;
2
3+ import java.io.IOException;
12
13 @WebServlet(urlPatterns = "/list-alumno.do")
14 public class ListAlumnoServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16     AlumnoService alumnoServicio= new AlumnoService();
17
18- @Override
19     protected void doGet(HttpServletRequest request,
20         HttpServletResponse response) throws IOException, ServletException {
21         request.getSession().setAttribute("pagina", new Pagina("Alumnos","list-alumno.do"));
22
23         request.setAttribute("alumnos",alumnoServicio.listaAlumnos());
24         request.getRequestDispatcher("/WEB-INF/views/list-alumno.jsp").forward(request, response);
25     }
26 }
```



... continuación **JSPF**:

Modifica header.jspf y menuPrincipal.jspf para que estos valores sean dinámicos en base al atributo nuevo “pagina”.

AYUDA: En cada servlet deberemos de crear un objeto de tipo “pagina” y añadirsele a la sesión para que la página jsp pueda mostrar el título en la pagina y pueda remarcar en azul el apartado en el que estamos poniendo “active” o “ “

según coincida la url con la página en la que estamos. Tener en cuenta que la página de alta add-alumno.do tendrá remarcado “Alumno” aunque estemos en el listado.

```
header.jspf
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <link href="webjars/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
6 <title>${pagina.getTitulo()}</title>
7 <style>
8     .footer {
9         position: absolute;
10        bottom: 0;
11        width: 100%;<link href="webjars/bootstrap/4.5.2/css/bootstrap.min.css"
12 rel="stylesheet">
13
14        height: 60px;
15        background-color: #f5f5f5;
16    }
17 </style>
18 </head>
19 <body>
```

```
menuSuperior.jspf
1 <nav class="navbar navbar-default">
2     <a href="http://ieslluissimarro.org/" class="navbar-brand">Simarro</a>
3     <nav class="nav nav-pills flex-column flex-sm-row">
4         <a class="nav-link ${pagina.getStrBootstrapActiva("login.do")}" href="/login.do">Home</a>
5         <a class="nav-link ${pagina.getStrBootstrapActiva("list-alumno.do")}" href="/list-alumno.do">
6         <a class="nav-link" href="http://www.ieslluissimarro.org/moodle/course/view.php?id=508">DWES
7     </nav>
8     <ul class="nav navbar-nav navbar-right">
9         <li><a href="/logout.do">Logout</a></li>
10    </ul>
11 </nav>
```



UD 1: Introducción a los lenguajes de servidor

4.- JEE: Servlets, JSP y JSTL



Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com

EJERCICIO:

Sigue todos los pasos de los PDF y sube la aplicación final al moodle.

Para ello:

1º Haz un “Run As \Maven Clean” para dejar solo los fichero fuentes y quitar momentaneamente los necesarios para ejecutar la aplicación (dependencias).

2º Comprime la carpeta de tu aplicación y ponle como nombre al fichero comprimido UD1_practica7_nombreAlumno.tar.gz donde nombreAlumno es el nombre del alumno que entrega la práctica.

3º Súbela al moodle.

IMPORTANTE: No comprimir en RAR, porque Ubuntu no lo lee bien y en clase tenemos Ubuntu. Si tuviesemos Windows, podemos comprimir en ZIP.