



### Objetivos de la sesión:

- *Aprender a utilizar los Spring MVC Form Tags más importantes para enlazar y validar datos:*
  - *CheckBox*
  - *Radiobutton*
  - *Password*
  - *Select*
  - *Options*
  - *TextArea*



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



Hemos utilizado **Spring MVC Form Tags en la vista (formulario JSP)** con la inclusión de la librería:

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="mvc"%>
```

La librería la suministra el fichero **spring-webmvc.jar** incluido en la dependencia spring-webmvc en Maven.

Gracias a esta librería podemos **enlazar y validar los datos de la vista con los parámetros con los que se invocan a los métodos en el controlador**.

Sin embargo, hasta ahora solo hemos visto los tags relacionados con:

- La declaración de formularios (mvc:form)
- Campos de texto/númericos (mvc:input)
- Etiquetas (mvc:label)
- Campos ocultos (mvc:hidden)
- Errores (mvc:errors)



¿Que otros tags existen?



***Spring MVC Form Tags*** también incorpora tags para tratar:

- **Checkbox**
- **Radiobutton**
- **Password**
- **Select**
- **Options**
- **TextArea**

Modifiquemos el formulario “Alumno” para incorporar estos tags y ver como funcionan y como enlazarlos en el controlador.



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### • Checkbox

Existen **3 tipos de checkbox** en la vista según con que tipo de objeto se enlacen en el controlador:

**1º** El valor a enlazar es de tipo **booleano (java.lang.Boolean)**

**2º** El valor a enlazar es una **lista (java.util.Collection)**

**3º** El valor a enlazar es de tipo **objeto (java.lang.Object)**

**Veamos con ejemplos como realizar la implementación de estos 3 tipos de checkbox!!**

# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag

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



### • Checkbox

**Modelo:** Modifica “Alumno.java” para incluir los siguientes campos resaltados (con sus getters y sus setters) :

```
Alumno.java
1 package org.profesor.joseramon.joseramon_primer_app_spring_mvc.model;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10
11
12
13
14
15
16 public class Alumno implements Modificable<Alumno>, Serializable, Comparable<Alumno>{
17     private static final long serialVersionUID = 1L;
18     @Pattern(regexp = "[0-9]{8}[A-Za-z]{1}", message = "El dni debe tener 8 números y una letra")
19     private String dni;
20     @Size(min=5,message="El nombre debe de tener un tamaño mínimo de 5 caracteres")
21     private String nombre;
22     @NotNull(message = "La edad no puede estar vacia")
23     @Range(min = 18, max = 99, message = "La edad debe ser igual o mayor a 18 y menor o igual a 99")
24     @Digits(integer = 2,fraction = 0, message = "La edad no puede tener decimales ni más de 2 dígitos")
25     private Integer edad;
26     @Size(min = 3, message = "El ciclo debe tener almenos 3 caracteres")
27     private String ciclo;
28     @NotNull(message = "El curso no puede estar vacio")
29     @Digits(fraction = 0, integer = 1,message = "El curso tiene un formato incorrecto")
30     @Range(min = 1, max = 2, message = "El curso solo admite los valores 1 o 2")
31     private Integer curso;
32
33     private boolean erasmus;
34     private String[] interesadoEn;
35     private String lenguajeFavorito;
36
37     private Date ts;
38     private String user;
39
40     public Alumno() {
41     }
42
43     public Alumno(String dni) {
44         super();
45         this.dni = dni;
46     }
47
48     public Alumno(String dni, String nombre, Integer edad, String ciclo, Integer curso) {
```





### • Checkbox

**Vista:** Modifica “add-alumno.jsp” para incluir los campos anteriores:

```
add-alumno.jsp
37 <mvc:label path="curso">Curso:</mvc:label>
38 <mvc:input type="number" id="curso" path="curso" class="form-control"/>
39 <mvc:errors path="curso" cssClass="text-warning"/>
40 </div>
41 <br/>
42
43 <div class="form-row">
44   <div class="col">
45     <mvc:checkbox path="erasmus"/>
46     <mvc:label path="erasmus">Alumno en Erasmus</mvc:label>
47   </div>
48   <div class="col">
49     <mvc:label path="interesadoEn">Interesado en:</mvc:label><br/>
50     <mvc:checkbox path="interesadoEn" value="Backend"/> Backend
51     <mvc:checkbox path="interesadoEn" value="Frontend"/>Frontend
52   </div>
53   <div class="col">
54     <mvc:label path="lenguajeFavorito">Lenguaje favorito:</mvc:label><br/>
55     <mvc:checkbox path="lenguajeFavorito" value="Java"/>&nbsp;&nbsp;&nbsp;Java
56   </div>
57 </div>
58 <br/><input type="submit" value="Añadir" class="btn btn-success"/>
59 </mvc:form>
60 </div>
```

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

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:	Nombre:	
<input type="text"/>	<input type="text" value="Nuevo Alumno"/>	
Edad:	Ciclo:	Curso:
<input type="text" value="18"/>	<input type="text" value="DAW"/>	<input type="text" value="2"/>
<input type="checkbox"/> Alumno en Erasmus		
Interesado en:		Lenguaje favorito:
<input type="checkbox"/> Backend <input type="checkbox"/> Frontend		<input type="checkbox"/> Java
<input type="button" value="Añadir"/>		



# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag



Desarrollo Web en Entorno Servidor - [joseramon.profesor@gmail.com](mailto:joseramon.profesor@gmail.com)

- **Checkbox**

**Controlador:** Sin cambiar nada puedes comprobar que ya podemos añadir correctamente alumnos. Crea un alumno de erasmus e incluye en el listado el campo erasmus de solo lectura:

[Simarro](#)

[Home](#)

[Alumnos](#)

[Modulos](#)

[Errores](#)

[DWES](#)

[Logout](#)

### Listado de alumnos:

Dni	Nombre	Edad	Ciclo	Curso	Erasmus	Acciones
11111111A	Jose	21	DAM	1	<input type="checkbox"/>	<a href="#">Modificar</a> <a href="#">Borrar</a>
22222222B	Pedro	32	DAW	2	<input type="checkbox"/>	<a href="#">Modificar</a> <a href="#">Borrar</a>
33333333C	Juan	23	ASIR	1	<input type="checkbox"/>	<a href="#">Modificar</a> <a href="#">Borrar</a>
44444444D	Nuevo Alumno	18	DAW	2	<input checked="" type="checkbox"/>	<a href="#">Modificar</a> <a href="#">Borrar</a>

[Añadir alumno](#)



### • Checkbox

No es muy elegante tener que crear la lista de elementos del checkbox desde la vista. Lo mejor sería que el controlador pudiera proporcionar esta lista

```
add-alumno.jsp
37      <mvc:label path="curso">Curso:</mvc:label>
38      <mvc:input type="number" id="curso" path="curso" class="form-control"/>
39      <mvc:errors path="curso" cssClass="text-warning"/>
40  </div>
41 </div>
42 <br/>
43 <div class="form-row">
44   <div class="col">
45     <mvc:checkbox path="erasmus"/>
46     <mvc:label path="erasmus">Alumno en Erasmus</mvc:label>
47   </div>
48   <div class="col">
49     <mvc:label path="interesadoEn">Interesado en:</mvc:label><br/>
50     <mvc:checkbox path="interesadoEn" value="Backend"/> Backend
51     <mvc:checkbox path="interesadoEn" value="Frontend"/>Frontend
52   </div>
53   <div class="col">
54     <mvc:label path="lenguajeFavorito">Lenguaje favorito:</mvc:label><br/>
55     <mvc:checkbox path="lenguajeFavorito" value="Java"/>&nbsp;Java
56   </div>
57 </div>
58 <br><input type="submit" value="Añadir" class="btn btn-success"/>
59 </mvc:form>
60 </div>
```



¿Que cambios tenemos que hacer si queremos que la lista “interesado en” se genere desde el controlador en vez de ponerlo manualmente en el “jsp”?



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### • Checkbox

Una de las soluciones pasa por que el controlador pase la lista en el modelo:  
Modifica la vista como en la imagen:

```
add-alumno.jsp
38         <mvc:input type="number" id="curso" path="curso" class="form-control"/>
39         <mvc:errors path="curso" cssClass="text-warning"/>
40     </div>
41 </div>
42 <br/>
43 <div class="form-row">
44     <div class="col">
45         <mvc:checkbox path="erasmus"/>
46         <mvc:label path="erasmus">Alumno en Erasmus</mvc:label>
47     </div>
48     <div class="col">
49         <mvc:label path="interesadoEn">Interesado en:</mvc:label><br/>
50         <mvc:checkboxes path="interesadoEn" items="${interesadoEnLista}"/>
51     </div>
52     <div class="col">
53         <mvc:label path="lenguajeFavorito">Lenguaje favorito:</mvc:label><br/>
54         <mvc:checkbox path="lenguajeFavorito" value="Java"/>&nbsp;Java
55     </div>
56 </div>
57 <br/><input type="submit" value="Añadir" class="btn btn-success"/>
58 </mvc:form>
59 </div>
60
61 <%@ include file="../../jspf/footer.jspf"%>
```



¿Que cambios tenemos que hacer en el controlador?



### • Checkbox

Para modificar el controlador tenemos que hacer primero que el servicio pueda devolver la lista que volcaremos en “InteresadoEnLista”.

Crea un nuevo método en “AlumnoService.java” que devuelva la lista:

```
public List<String> listaInteresadoEn()
```

A continuación tenemos 2 opciones para modificar el controlador:

**1º opción:** Añadir el atributo de la lista al modelo

```
AlumnoController.java
49 @RequestMapping(value="/add-alumno",method = RequestMethod.GET)
50 public String mostrarAlumno(ModelMap model) {
51     paginaServicio.setPagina(pagina);
52     model.addAttribute("pagina",paginaServicio.getPagina());
53     model.addAttribute("alumno",new Alumno("", "Nuevo Alumno",18,"DAW",2));
54     model.addAttribute("interesadoEnLista", alumnoService.listaInteresadoEn().toArray());
55     return "add-alumno";
56 }
```



### • Checkbox

**2º opción:** Crear un método nuevo que contenga datos del modelo. Esta opción es buena si esta lista se va a utilizar en más de un método y no es demasiado larga.

Al crear un “ModelAttribute” en el controller no hace falta indicarle cada vez la lista “interesadoEnLista” porque el “ModelAttribute” actúa como una especie de variable de sesión y se pasa con el modelo en todos métodos del controlador (de ahí que es buena si se utiliza en más de un método y no es demasiado costosa).

```
AlumnoController.java
49
50 @RequestMapping(value="/add-alumno",method = RequestMethod.GET)
51 public String mostrarAlumno(ModelMap model) {
52     paginaServicio.setPagina(pagina);
53     model.addAttribute("pagina",paginaServicio.getPagina());
54     model.addAttribute("alumno",new Alumno("", "Nuevo Alumno",18,"DAW",2));
55     //model.addAttribute("interesadoEnLista", alumnoService.listaInteresadoEn().toArray());
56     return "add-alumno";
57 }
58
59 @ModelAttribute("interesadoEnLista")
60 public Object[] getinteresadoEnLista() {
61     return alumnoService.listaInteresadoEn().toArray();
62 }
```

Comprueba que funcionan ambos métodos y deja el 2º método.



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### • Checkbox

Modifica “update-alumno.jsp” para incorporar a la página de modificación del alumno los datos de tipo checkbox añadidos a la página “add-alumno.jsp”.

La magia de Spring es que solo tendrás que modificar el JSP, porque si has añadido el ModelAttribute “interesadoEnLista” a nivel de controlador (para añadir alumnos) no hace falta hacer nada más para el apartado de modificar porque el controller recibe y envia “beans”.





## • Radiobutton

Spring también permite declarar un radiobutton (input HTML con el tipo “radio”):

add-alumno.jsp

```
36<div class="col">
37    <mvc:label path="curso">Curso:</mvc:label>
38    <mvc:input type="number" id="curso" path="curso" class="form-control">
39    <mvc:errors path="curso" cssClass="text-warning"/>
40</div>
41</div>
42<br/>
43<div class="form-row">
44    <div class="col">
45        <mvc:checkbox path="erasmus"/>
46        <mvc:label path="erasmus">Alumno en Erasmus</mvc:label>
47    </div>
48    <div class="col">
49        <mvc:label path="interesadoEn">Interesado en:</mvc:label><br/>
50        <mvc:checkboxes path="interesadoEn" items="${interesadoEnLista}" />
51    </div>
52    <div class="col">
53        <mvc:label path="lenguajeFavorito">Lenguaje favorito:</mvc:label>
54        <mvc:checkbox path="lenguajeFavorito" value="Java"/>&nbsp;&nbsp;&nbsp;Java
55    </div>
56</div>
57<div class="form-row">
58    <div class="col">
59        <mvc:label path="genero">Género:</mvc:label><br/>
60        <mvc:radiobutton path="genero" value="Hombre" /> Hombre<br/>
61        <mvc:radiobutton path="genero" value="Mujer" /> Mujer
62    </div>
63</div>
64<br/>
65<input type="submit" value="Añadir" class="btn btn-success">
66</mvc:form>
67</div>
68
69<%@ include file="../../jspf/footer.jspf"%>
```

localhost:8080/joseramon\_primer\_app\_spring

Simarro Home Alumnos Modulos Errores DWES Logout

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:

Nombre:

Edad:

Ciclo:

Curso:

☐ Alumno en Erasmus

Interesado en:

Lenguaje favorito:

☐ Backend☐ Frontend

☐ Java

Género:

☐ Hombre

☐ Mujer

Añadir

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





# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag

Desarrollo Web en Entorno Servidor - [joseramon.profesor@gmail.com](mailto:joseramon.profesor@gmail.com)



### • Radiobutton

Modifica el `jsp` para que sea el controlador el que proporcione la lista de items del radiobutton utilizando `mvc:radiobuttons` de igual manera que hemos hecho con `mvc:checkboxes`. Por defecto aparecen todos en horizontal, **investiga como puedes llegar a hacer que los radiobuttons esten en vertical** como en la imagen:

Simarro Home **Alumnos** Modulos Errores DWES Logout

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:  Nombre:

Edad:  Ciclo:  Curso:

☐ Alumno en Erasmus Interesado en: ☐ Backend ☐ Frontend Lenguaje favorito: ☐ Java

Género: ☐ Hombre ☐ Mujer

**Ayuda:** Solo tienes que tocar el `jsp` para indicarle a la vista que cada elemento del radiobutton se separa , por ejemplo con el tag “p”.



## • Password

Para gestionar un campo de tipo password desde Spring debemos utilizar el tag “password”. **Modifica el login** para que se puede utilizar **mvc:password** en el JSP.

```
login.jsp
1 <%@ include file="../../jspf/header.jspf"%>
2 <%@ include file="../../jspf/menuSuperior.jspf"%>
3 <div class="container">
4 <p>
5 <font color="red">${errores}</font>
6 </p>
7 <mvc:form method="post" action="login" modelAttribute="usuario">
8 <mvc:label path="nickname">Introduzca su usuario:</mvc:label>
9 <mvc:input path="nickname" type="text" class="form-control"/>
10 <mvc:errors path="nickname" cssClass="text-warning"/>
11 <mvc:label path="password">Introduzca su contraseña:</mvc:label>
12 <mvc:password path="password" class="form-control"/>
13 <mvc:errors path="password" cssClass="text-warning"/><br/>
14 <input type="submit" value="Login" class="btn btn-success">
15 </mvc:form>
16 </div>
17 <%@ include file="../../jspf/footer.jspf"%>
18
```



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### • Password

Este cambio implica además **realizar los siguientes pasos**:

- **Modelo:** Crear la clase Usuario.java que contiene el nickname, nombre y el password. 2 usuarios se considerarán iguales si tienen el mismo nickname (equals). Utilizando Spring y expresiones regulares valida que el nickname no puede tener menos de 5 caracteres y el password debe tener entre 8 y 16 caracteres, y al menos un dígito, una minúscula, una mayúscula y un carácter no alfanumérico.

- **Servicio:** LoginServicio creará 2 usuarios (joseramon/miPassword@1 y el usuario que ya tenias tu como alumno anteriormente en la aplicación) de manera similar a como se crean los alumnos precargados iniciales y tendrá 2 métodos:

```
Usuario encontrarUsuarioPorNickName(String nickname)  
boolean usuarioValido(Usuario usuarioLogin)
```

- **Controlador:** LoginController utilizará la clase Usuario en vez de los atributos sueltos como parámetros de los métodos. Al hacer login se deberá de validar el Usuario e informar de los posibles errores. A nivel de sesión se almacenara el usuario. Cuando se realice una modificación en Alumnos se guardará el nickname de quien lo modifiko, no su nombre completo.

- **Vista:** Modificar bienvenida.jsp y login.jsp para incorporar los cambios.



# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag

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



### • Password

localhost:8080/joseramon\_primer\_app\_...

Simarro Home Alumnos Modulos Errores DWES Logout

Usuario 'jose' o contraseña incorrecta

Introduzca su usuario:

El usuario debe de tener un tamaño mínimo de 5 caracteres

Introduzca su contraseña:

La contraseña debe tener entre 8 y 16 caracteres, y al menos un dígito, una minúscula, una mayúscula y un caracter no alfanumérico.

Login

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



localhost:8080/joseramon\_primer\_app\_...

Simarro Home Alumnos Modulos Errores DWES Logout

Usuario 'jose' o contraseña incorrecta

Introduzca su usuario:

El usuario debe de tener un tamaño mínimo de 5 caracteres

Introduzca su contraseña:

La contraseña debe tener entre 8 y 16 caracteres, y al menos un dígito, una minúscula, una mayúscula y un caracter no alfanumérico.

Login

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



localhost:8080/joseramon\_primer\_app\_...

Simarro Home Alumnos Modulos Errores DWES Logout

Bienvenid@ JoseRa





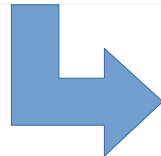
### Select y Option

Spring también permite crear combos HTML que contiene una serie de valores que se muestran al desplegar el combo:

```

53 <input type="checkbox" path="lenguaje favorito" value="java" /> java, java
54 </div>
55 </div>
56 <div class="form-row">
57 <div class="col">
58 <mvc:label path="genero">Género:</mvc:label>
59 <ul>
60 <mvc:radio buttons path="genero" items="{opc
61 </ul>
62 </div>
63 <div class="col">
64 <mvc:label path="horario">Horario:</mvc:label>
65 <mvc:select path="horario">
66 <mvc:option value="Mañana" />
67 <mvc:option value="Tarde" />
68 </mvc:select>
69 </div>
70 </div>
71 <br><input type="submit" value="Añadir" class="btn btn-
72 </mvc:form>

```



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

## Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:
Nombre:

Edad:
Ciclo:
Curso:

☐ Alumno en Erasmus
Interesado en:
Lenguaje favorito:

☐ Backend ☐ Frontend
☐ Java

Género:
Horario:

☐ Hombre
☐ Mujer





## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### • Select y Option

Modifica el `jsp` para que sea el controlador el que proporcione la lista de items del combo utilizando `mvc:options` de igual manera que hemos hecho con `mvc:checkboxes`:

Simarro   Home   Alumnos   Modulos   Errores   DWES   Logout

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:    Nombre:

Edad:    Ciclo:    Curso:

☐ Alumno en Erasmus   Interesado en: ☐ Backend ☐ Frontend   Lenguaje favorito: ☐ Java

Género: ☐ Hombre ☐ Mujer   Horario:

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

Ayuda: el `ModelAttribute` tendrá que devolver `List<String>`



### Select y Option

Modifica el jsp para almacenar el **país** y que por defecto te muestre “Ninguno”. Fijate que en vez de almacenar el texto que se muestra en el navegador se almacena la abreviatura ( en “Ninguno” se almacenará “-”, en “España” será “ES”).

```
add-alumno.jsp
60 <ul>
61 <li><mvc:radio buttons path="genero" items="${opcionesGenero}" element="p" />
62 </li>
63 </ul>
64 </div>
65 <div class="col">
66 <mvc:label path="horario">Horario:</mvc:label><br/>
67 <mvc:select path="horario">
68 <options items="${horarioLista}" />
69 </mvc:select>
70 </div>
71 <div class="col">
72 <mvc:label path="pais">País:</mvc:label><br/>
73 <mvc:select path="pais">
74 <option value="-" label="Ninguno" />
75 <options items="${paisLista}" />
76 </mvc:select>
77 </div>
78 <br><input type="submit" value="Añadir" class="btn btn-success" />
79 </mvc:form>
```

Simarro

Home

Alumnos

Modificar

Errores

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:  Nombre:

Edad:  Ciclo:  Curso:

☐ Alumno en Erasmus

Interesado en: ☐ Backend ☐ Frontend

Lenguaje favorito: ☐ Java

Género: ☐ Hombre ☐ Mujer

Horario:

País:

**Ayuda:** Piensa que una lista con 2 campos es realmente un Map < String, String >.

Esto significa que devuelves un HashMap al JSP ...

Añadir



# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag

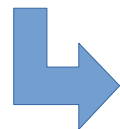
Desarrollo Web en Entorno Servidor - Joseramon.profesor@gmail.com



### • Select y Option

Modifica el jsp para almacenar el módulo en el que está matriculado el alumno. Habrá un “ModelAttribute” en el controlador que devolverá la lista de módulos.

```
add-alumno.jsp
69<div class="col">
70<mvc:label path="pais">País:</mvc:label><br/>
71<mvc:select path="pais">
72<mvc:option value="." label="Ninguno" />
73<mvc:options items="${paisLista}" />
74</mvc:select>
75</div>
76<div class="col">
77<mvc:label path="matriculadoEn">matriculado En:</mvc:label><br/>
78<mvc:select path="matriculadoEn">
79<mvc:options items="${moduloLista}" itemValue="id" itemLabel="nombre" />
80</mvc:select>
81</div>
82</div>
83<br><input type="submit" value="Añadir" class="btn btn-success">
84</mvc:form>
```



**Ayuda:** El jsp recibirá una lista de módulos y mostrará el itemLabel “nombre” (pero almacenará el itemValue “id”). En el bean “Alumno” se almacenará un Integer.

Simarro Home Alumnos Modulos Erasm...

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni: 66666666y Nombre: Alumno 6

Edad: 18 Ciclo: DAW Curso: 2

☐ Alumno en Erasmus Interesado en: ☐ Backend ☐ Frontend Lenguaje favorito: ☐ Java

Género: ☐ Hombre ☐ Mujer Horario: Mañana País: Ninguno matriculado En: Desarrollo Web en Entorno Servidor

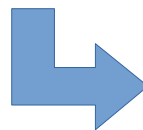
alumno	Alumno (id=125)
ciclo	"DAW" (id=142)
curso	Integer (id=143)
dni	"66666666y" (id=146)
edad	Integer (id=147)
erasmus	false
genero	null
horario	"Mañana" (id=148)
interesadoEn	String[0] (id=149)
lenguajeFavorito	null
matriculadoEn	Integer (id=143)
value	2
nombre	"Alumno 6" (id=150)
pais	"-" (id=151)
ts	null
user	null



### Select y Option

Modifica el jsp para poder seleccionar en el combo más de un módulo para así indicar los módulos matriculados:

```
add-alumno.jsp
64 <mvc:label path="horario">Horario:</mvc:label><br/>
65 <mvc:select path="horario">
66 <mvc:options items="${horarioLista}" />
67 </mvc:select>
68 </div>
69 <div class="col">
70 <mvc:label path="pais">País:</mvc:label><br/>
71 <mvc:select path="pais">
72 <mvc:option value="-" label="Ninguno" />
73 <mvc:options items="${paisLista}" />
74 </mvc:select>
75 </div>
76 <div class="col">
77 <mvc:label path="matriculadoEn">matriculado En:</mvc:label><br/>
78 <mvc:select path="matriculadoEn" items="${moduloLista}" itemValue="id" itemLabel="nombre">
79 </mvc:select>
80 </div>
81 </div>
82 <br><input type="submit" value="Añadir" class="btn btn-success">
83 </mvc:form>
```



Simarro Home Alumnos Modi

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni: 77777777G

Edad: 18 Ciclo: DAW Curso: 2

☐ Alumno en Erasmus Interesado en: ☐ Backend ☐ Frontend Lenguaje favorito: ☐ Java

Género: ☐ Hombre ☐ Mujer Horario: Mañana País: Ninguno

matriculado En: Programación Desarrollo Web en Entorno Servidor

Añadir

alumno	Alumno (id=82)
ciclo	"DAW" (id=124)
curso	Integer (id=125)
dni	"77777777G" (id=128)
edad	Integer (id=129)
erasmus	false
genero	null
horario	"Mañana" (id=130)
interesadoEn	String[0] (id=131)
lenguajeFavorito	null
matriculadoEn	ArrayList<E> (id=132)
elementData	Object[10] (id=143)
modCount	2
size	2
nombre	"Alumno 7" (id=141)
pais	"-" (id=142)
ts	null
user	null

**Ayuda:** Fijate en el cambio del jsp para poder seleccionar más de una opción. El modelo "Alumno" contendrá en matriculadoEn un ArrayList de enteros, pero en el jsp se recibirá una lista de módulos.

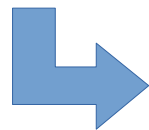




### · TextArea

Modifica el jsp para poder incorporar un campo donde explicar los hobbies del alumno:

```
add-alumno.jsp
74      </mvc:select>
75    </div>
76    <div class="col">
77      <mvc:label path="matriculadoEn">matriculado En:</mvc:label>
78      <mvc:select path="matriculadoEn" items="${moduloLista}" />
79    </div>
80  </div>
81
82  <div class="form-row">
83    <div class="col">
84      <mvc:label path="hobbies">Hobbies:</mvc:label><br/>
85      <mvc:textarea path="hobbies" rows="3" cols="70" />
86    </div>
87  </div>
88  <br><input type="submit" value="Añadir" class="btn btn-success">
89 </mvc:form>
90 </div>
91
92 <%@ include file="../jspf/footer.jspf"%>
```



**Ayuda:** El modelo “Alumno” contendrá un nuevo campo de tipo “String” llamado “hobbies”

Simarro

Home

Alumnos

Modulos

Errores

DWES

Logout

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

Dni:

Nombre:

Edad:

Ciclo:

Curso:

☐ Alumno en Erasmus

Interesado en:

☐ Backend ☐ Frontend

Lenguaje favorito:

☐ Java

Género:

☐ Hombre☐ Mujer

Horario:

País:

matriculado En:

Hobbies:

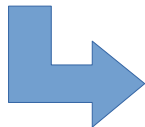




### errors

Modifica el jsp para que aparte de mostrar los errores debajo de cada campo se muestre un cuadro inicial con todos los errores:

```
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>
7     <font color="red">${errores}</font>
8   </p>
9   <p> Introduzca los datos del nuevo alumno:</p>
10
11   <mvc:form method="post" action="add-alumno" modelAttribute="alumno">
12     <mvc:errors path="*" cssClass="text-warning" />
13     <div class="form-row">
14       <div class="col">
15         <mvc:label path="dni">Dni:</mvc:label>
16         <mvc:input path="dni" type="text" class="form-control"/>
17         <mvc:errors path="dni" cssClass="text-warning" />
18       </div>
19       <div class="col">
20         <mvc:label path="nombre">Nombre:</mvc:label>
```



Simarro Home Alumnos Modulos Errores DWES Logout

### Nuevo alumno:

Introduzca los datos del nuevo alumno:

El curso solo admite los valores 1 o 2  
El nombre debe de tener un tamaño mínimo de 5 caracteres  
La edad debe ser igual o mayor a 18 y menor o igual a 99  
El dni debe tener 8 números y una letra  
El ciclo debe tener almenos 3 caracteres

Dni: 12 Nombre: AI  
El dni debe tener 8 números y una letra El nombre debe de tener un tamaño mínimo de 5 caracteres

Edad: 1 Ciclo: DA Curso: 7  
La edad debe ser igual o mayor a 18 y menor o igual a 99 El ciclo debe tener almenos 3 caracteres El curso solo admite los valores 1 o 2

☐ Alumno en Erasmus Interesado en: ☐ Backend ☐ Frontend Lenguaje favorito: ☐ Java  
Género: ☐ Hombre ☐ Mujer Horario: Mañana País: Ninguno matriculado En: Programación Desarrollo Web en Entorno Servidor



# UD 2: Modelo Vista Controlador

## 8.- Librería Spring Form Tag

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



Modifica el **JSP de modificación del alumno** para incorporar todos los campos que hemos incorporado al añadir un nuevo alumno

Simarro   Home   **Alumnos**   Modulos   Errores   DWES   Logout

### Alumno a modificar:

Dni:	Nombre:		
<input type="text" value="77777777G"/>	<input type="text" value="Alumno 7"/>		
Edad:	Ciclo:	Curso:	
<input type="text" value="19"/>	<input type="text" value="DAW"/>	<input type="text" value="1"/>	
<input type="checkbox"/> Alumno en Erasmus	Interesado en:	Lenguaje favorito:	
	<input type="checkbox"/> Backend <input type="checkbox"/> Frontend	<input type="checkbox"/> Java	
Género:	Horario:	País:	matriculado En:
<input checked="" type="radio"/> Hombre	<input type="text" value="Tarde"/>	<input type="text" value="Francia"/>	<div>Programación</div> <div>Desarrollo Web en Entorno Servidor</div>
<input type="radio"/> Mujer			
Hobbies:			
<input type="text" value="Diseño gráfico y retoque fotográfico"/>			
<input type="button" value="Modificar"/>			



## UD 2: Modelo Vista Controlador

### 8.- Librería Spring Form Tag

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



#### EJERCICIO:

Sigue todos los pasos de los PDF. 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 UD2\_practica5\_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.