

# **FORMULARIOS, VALIDACIONES y PROPIEDADES**

# ÍNDICE DE CONTENIDOS

- **Formularios.**
- **Validaciones.**
- **Propiedades.**

# FORMULARIOS

## CAMPOS CON UN SOLO VALOR

```
<p>  
  <label for="simple">Campo simple</label>  
  <input type="text" name="simple"/>  
</p>
```

JSP

```
PrintWriter out = response.getWriter();  
String simple = request.getParameter("simple");  
out.println("El valor de Simple es: " + simple);
```

SERVLET

# FORMULARIOS

## CAMPOS CON VARIOS VALORES

```
<p><label for="compuesto_checkbox">Campo compuesto usando un checkbox</label></p>
<input type="checkbox" name="checkbox" value="A"/>Opción A<br>
<input type="checkbox" name="checkbox" value="B"/>Opción B<br>
<input type="checkbox" name="checkbox" value="C"/>Opción C<br>
<input type="checkbox" name="checkbox" value="D"/>Opción D<br>
```

JSP

```
String[] valores = request.getParameterValues("checkbox");
out.println("El valor de Compuesto es: " + Arrays.toString(valores));
```

SERVLET

# FORMULARIOS

## CAMPOS CON VARIOS VALORES

`<p><label for="compuesto_checkbox">Campo compuesto usando un select de opción múltiple</label></p>`

```
<select name="multiple" multiple>
  <option>Selección A</option>
  <option>Selección B</option>
  <option>Selección C</option>
  <option>Selección D</option>
</select>
```

JSP

```
String[] seleccion = request.getParameterValues("multiple");
out.println("El valor de Compuesto es: " + Arrays.toString(seleccion));
```

SERVLET

# FORMULARIOS

## CAMPOS DE TIPO FILE

```
<form method="post" [action="..."] [enctype="..."]>
```

```
</form>
```

## VALORES PARA ENCTYPE

- **text/plain** : .....
- **application/x-www-form-urlencoded**: Opción por defecto. Se codifican todos los caracteres antes de ser enviados (ASCII HEX)
- **multipart/form-data**: Si quiero enviar ficheros al servidor (type="file"). No necesario de momento.

# VALIDACIÓN

“ La validación debe realizarse tanto en cliente como en servidor. “PROGRAMACIÓN DEFENSIVA”. ”

# VALIDACIÓN

VALIDACIÓN EN CLIENTE	VALIDACIÓN SERVIDOR
<ul style="list-style-type: none"><li>● Se realiza en el HTML de manera previa a que los datos sean mandados al servidor.</li><li>● Se utilizan estrategias como:<ul style="list-style-type: none"><li>○ Usar el atributo type adecuado</li><li>○ Establecer los campos requeridos</li><li>○ Usar el atributo pattern en las etiquetas HTML <a href="https://www.w3schools.com/tags/att_input_pattern.asp">https://www.w3schools.com/tags/att_input_pattern.asp</a></li></ul></li></ul>	<ul style="list-style-type: none"><li>● Se realiza en el SERVLET (Normalmente) o en JSP.</li><li>● Utilizaremos expresiones regulares <code>java.util.Regex.Pattern</code></li><li>● En esta web podemos probar expresiones regulares <a href="https://regexr.com">https://regexr.com</a></li><li>● Tengo que:<ol style="list-style-type: none"><li>a. Definir el patrón en un variable <code>private final String</code>.</li><li>b. Compilar ese patrón para obtener un objeto <code>Pattern</code></li><li>c. Comprobar si se cumple el patrón usando método del objeto <code>Pattern</code></li></ol></li></ul>



# VALIDACIÓN

- a) *// digit + lowercase char + uppercase char + punctuation + symbol. 8 - 20 characters.*  
`private static final String PASSWORD_REGEX =  
 "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()-[{ }]:;','?/*~$^+=<>]).{8,20}$";`
- b) `private static final Pattern PASSWORD_PATTERN = Pattern.compile(PASSWORD_REGEX);`
- c) `PASSWORD_PATTERN.matcher(password).matches();`

# PROPERTIES

“ Para guardar datos de configuración de una aplicación usaremos la clase **Properties** que nos permitirá archivos con extensión **.properties**, **.xml**, **.config** o **.txt**. ”

# PROPERTIES

```
import java.util.*

Properties prop = new Properties();

InputStream is = null;

is = getClass().getClassLoader().getResourceAsStream("bd.properties");

prop.load(is);

String usuario = prop.getProperty("servidor.usuario");

for (Enumeration e = prop.keys(); e.hasMoreElements(); ) {
    //...
    Object obj = e.nextElement();
    String value = prop.getProperty(obj.toString());
    //...
}
```

# PROPERTIES

## # Propiedades del Sistema

- `java.version`
- `java.vendor`
- `java.home`
- `os.name`
- `os.version`
- `user.name`
- `user.home`
- `java.vm.version`
- `java.vm.name`

# PROPERTIES

## CONSIDERACIONES

- Por defecto la carpeta es `/src/main/resources`
- Puedo añadir otras carpetas desde el build path (Source)
- También puedo añadir otras carpetas desde el `pom.xml`  
<https://maven.apache.org/plugins/maven-resources-plugin/examples/resource-directory.html>

# PROPERTIES

## CONSIDERACIONES

- También puedo añadir otras carpetas desde el `pom.xml`  
<https://maven.apache.org/plugins/maven-resources-plugin/examples/resource-directory.html>  
`<resources>`  
    `<resource>`  
        `<directory>resources</directory>`  
    `</resource>`  
`</resources>`

**END**



**jgardur081@g.educaand.es**