



## AUT02\_04. Euromillon

Lee atentamente. Fíjate bien en dónde consigues los puntos y preocúpate de los detalles al final.

El nombre del fichero a subir **AUT02\_04\_Apellido1Apellido2Nombre.zip**

### 1. Creación del proyecto

Se creará un proyecto Java EE 7 con las siguientes características:

Nombre: EuromillónApellido1Nombre

Tipo: Maven > Web Application

Group Id: es.cifpcm

Versión Java EE: Java EE 7 Web

#### 1.1 Página HTML

Vamos a crear una página HTML (index.html) con un formulario que representará un boleto del Euromillon.

EUROMILLON

localhost:8080/EuromillonMartinez/

### ¡Bienvenido a tu EUROMILLON!

Introduce tu combinación ganadora y pulsa el botón enviar

Número 1

Número 2

Número 3

Número 4

Número 5

Ahora las estrellas

Estrella 1

Estrella 2

Enviar Borrar

El código HTML de la página:



```
-
<h1>¡Bienvenido a tu EUROMILLON!</h1>
<body>
<form action="/EuromillonMartinez/resultados" method="POST">
  <fieldset>
    <legend>Introduce tu combinación ganadora
    y pulsa el botón enviar</legend>
    <p>
      <label for="txtNum1"> Número 1</label>
      <input id="txtNum1" type="text" name="numero" />
    </p>
    <p>
      <label for="txtNum2"> Número 2</label>
      <input id="txtNum2" type="text" name="numero" />
    </p>
    <p ...4 lines />
    <p ...4 lines />
    <p>
      <label for="txtNum5"> Número 5</label>
      <input id="txtNum5" type="text" name="numero" />
    </p>
    Ahora las estrellas
    <p>
      <label for="txtEst1"> Estrella 1</label>
      <input id="txtEst1" type="text" name="estrella" />
    </p>
    <p ...4 lines />
  </fieldset>
  <input type="submit" value="Enviar"/>
  <input type="reset" value="Borrar"/>
</form>
</body>
./html>
```

Se observa que todos los campos de tipo text tienen el mismo valor para el atributo **name**; el valor **"numero"**. De esta forma definimos un array de inputs que podrá ser leído en el servidor mediante el método `request.getParameterValues`.

```
String[] szNumeros = req.getParameterValues("numero");
```

## 2. El Servlet

Se creará un servlet **ResultadosServlet**, mapeado en la URL **/EuromillonApellido/resultados**.



## 2.1 Inicialización de la combinación ganadora en método init

El servlet define un **TreeSet** estático como variable miembro. En este TreeSet se almacenará la combinación ganadora, que se crea en la inicialización del servlet (método **init**). Busca la información necesaria para hacerlo.

**NOTA:** Se utiliza TreeSet porque al ser un set no permite duplicados. Además TreeSet es una colección ordenada. Presta atención y respeta la estructura que se pide.

**NOTA:** Se utiliza **java.util.Random** para generar aleatorios. El método **nextInt** devuelve un entero de 0 a n.

```
@Override
public void init() throws ServletException{
    super.init();
    Random rnd= new Random();
    while (COMBINACION_GANADORA.size() < 5) {
        int num = rnd.nextInt(49)+ 1;
        if(!COMBINACION_GANADORA.contains(num)){
            COMBINACION_GANADORA.add(num);
        }
    }
}
```

## 2.2 Procesamiento de los campos de entrada

El método **getParameterValues** devuelve un array de String (**String[]**) debemos convertir estos String en Integer, utilizando el método **Integer.parseInt** y se recomienda almacenar estos Integer en una nueva colección **TreeSet<Integer>**.

## 2.3 Obtención del resultado

El servlet deberá ir comprobando, uno a uno, los números introducidos. Si un número está en el TreeSet de la combinación ganadora (método **contains**) el n.º de aciertos se incrementará en 1.

En caso de que hubiese un error en la entrada:

- i. Números repetidos en el boleto.
- ii. Valores no enteros en el boleto.

El servlet devolverá una página con el texto Error (error.jsp).



En caso contrario el servlet devolverá una página donde se indique la combinación ganadora, el número de aciertos y los números introducidos. Si n.º aciertos=5 se indica al usuario que gana. BRAVO!!



Más adelante espero ampliar esta práctica con bases de datos. Por favor hazla lo mejor posible!! para que se pueda reutilizar.

La práctica se valora de 0 a 9. Para obtener el 10, hay que implementar correctamente la parte de las estrellas. Las estrellas van de 1 a 9.

### 3. Puntuación:

Nombres y Comentarios	2
Mapeado correcto	1
Métodos init y post	1
Uso TreeSet con sus funcionalidades	1
Funciona	3
Salida de error y éxito	1
Implementación Estrellas	1