AUT02_04. Primitiva

1. Proyecto Euromillón.

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

1.1 Creación del proyecto

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

Nombre: Euromillón.

Tipo: Maven > Web Application

Group Id: es.cifpcm

Versión Java EE: Java EE 7 Web

1.1.1 Creación del servlet

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

1.2 Página HTML

Vamos a crear una página HTML (index.html) con un formulario que representará un boleto de la Primitiva.

El código HTML de la página:

```
<form action="/Primitiva/resultados" method="POST">

    <fieldset>
        <legend>Introduce tu combinación y pulsa el botón de 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 ...4 lines />
        <p>
            <label for="txtNum6">Número 6</label>
            <input id="txtNum6" type="text" name="numero"/>
        </p>
    </fieldset>

    <input type="submit" value="Enviar"/>
    <input type="reset" value="Borrar"/>

</form>
```

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");
```

1.3 El Servlet

1.3.1 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. Como ya vimos en nuestra actividad inicial en el ejercicio AUT02_01_3

```
@Override
public void init() throws ServletException {
    super.init(); //To change body of generated methods, choose Tools | Tem
    Random rnd = new Random();

    while (COMBINACION_GANADORA.size() < 6) {

        int num = rnd.nextInt(45) + 1;
        if (!COMBINACION_GANADORA.contains(num)) {
            COMBINACION_GANADORA.add(num);
```

1.3.2 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>.

1.3.3 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.

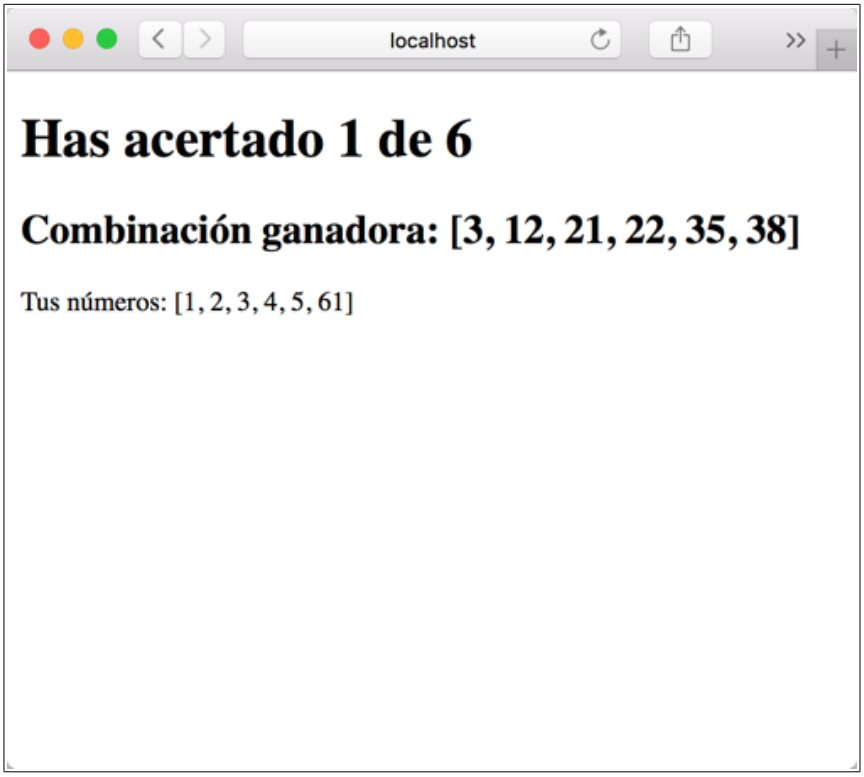
En caso de que hubiese un error en la entrada:

i. Números repetidos en el boleto.

ii. Valores no enteros en el boleto.

El servlet devolverá una página con el texto Error.

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=6 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.