



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FACULTAD DE CIENCIAS FISICO MATEMATICAS

## Diseño Orientado a Objetos

Practica Laboratorio 3

Maestro: Miguel Salazar

INTEGRANTE

Gilberto Alejandro Contreras Silva – 1683471

GRUPO: 007      AULA: 409

FECHA: Monterrey, N.L., 21 de febrero de 2017

## Practica 3

Primeramente en esta práctica debimos de haber creado el documento HTML, el cual deberá contener un formulario para que se puedan enviar los datos

```
7 <html>
8   <head>
9     <title>Practica 3</title>
10    <meta charset="UTF-8">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12  </head>
13  <body>
14    <h1>Registro</h1><hr>
15    <form action="Registro" method="POST" name="hola">
16      <br>
17      Usuario: <input type="text" name="usuario" id="usuario"/><br>
18      Contraseña: <input type="password" name="pass" id="pass"/><br>
19      Email: <input type="text" name="mail" id="mail"/><br>
20      Mes de Nacimiento: <select name="mes" id="mes">
21        <option>Selecciona</option>
22        <option value="1">1</option>
23        <option value="2">2</option>
24        <option value="3">3</option>
25        <option value="4">4</option>
26        <option value="5">5</option>
```

Este formulario deberá ser llamado porque posteriormente usaremos un botón para borrar los datos, entonces necesitaremos una referencia a este formulario.

```
33    <option value="12">12</option>
34  </select><br>
35  Sexo:<br>
36  <input type="radio" name="vato" value="M">Masculino<br>
37  <input type="radio" name="vato" value="F">Femenino<br>
38  Soy mayor de edad <input type="checkbox" name="casilla" id="casilla"/><br>
39  Acerca de mi: <br>
40  <textarea rows="5" cols="20" name="acerca">
41  </textarea><br>
42
43  <input type="button" value="BORRAR" onClick="borrar()">
44  <input type="submit" name="boton" id="boton" value="ENVIAR">
45  <script>
46    function borrar() {
47      document.forms["hola"].reset();
48    }
49  </script>
50  </form>
51 </body>
52 </html>
```

Crearemos un Script al final del formulario, que contendrá la función que pusimos en el oneClick del botón de borrar, porque en ella escribiremos lo que hará que los datos introducidos se limpien.

Dejando de lado lo que es el HTML, seguiremos con lo que es el Registro.java, el que contendrá la verdadera funcionalidad de la página.

```
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
31     throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         try (PrintWriter out = response.getWriter()) {
34             String usuario = request.getParameter("usuario");
35             String contra = request.getParameter("pass");
36             String mail = request.getParameter("mail");
37             String mes = request.getParameter("mes");
38             String sexo = request.getParameter("vato");
39             String mayor = request.getParameter("casilla");
40             String acerca = request.getParameter("acerca");
41
42             String nusuario = limpiar(usuario);
43             String ncontra = limpiar(contra);
44             String nmail = limpiar(mail);
45             String nacerca = limpiar(acerca);
46
47
48             out.println("<!DOCTYPE html>");
49             out.println("<html>");
50             out.println("<head>");
```

En la parte superior de esta página contiene las bibliotecas y demás que serán utilizadas, previamente escritas por el mismo NetBeans.

Lo que hicimos fue declarar las variables como cadenas de texto y dándoles el valor de respuesta del request de los campos de texto que escribimos en el HTML, eso se defina sabiendo cómo se declararon los nombres de los inputs en el HTML.

Posteriormente, limpiamos el valor introducido con la función limpiar, por lo que se tiene que crear una nueva variable que contenga el resultado ya procesado por así decirlo. La igualaremos el valor de respuesta de la función limpiar (que más adelante la explico) con la variable anterior que deseemos validar.

```
    }
}
public String limpiar(String cadena){
    try{
        String nuevacadena = java.net.URLEncoder.encode(cadena,"UTF-8");
        return nuevacadena;
    }
    catch(UnsupportedEncodingException e){
        return "Se traba aqui";
    }
}
```

En la parte de los prints pues la verdad no hay nada diferente, solo que esta vez imprimirá los valores de las nuevas variables.

Ahora, la función limpiar. Básicamente lo que hace esta es recibir parámetros de cadena, y va a intentar crear una nueva cadena de texto la cual estará con el valor de URLEncode, con parámetros de la cadena anterior que recibió y espaciado con una coma, el UTF deseado. Esta función regresará el valor de la nueva cadena transformada.

El catch solo está en caso de que no pueda regresar nada pues que me devuelva esa frase para saber dónde está el problema.

Es interesante como el profesor comentó justamente esta tarea en clase durante esa semana de trabajo, de cómo le haríamos para validar los caracteres que se ingresan y creo que esta puede ser una buena solución.