



UNIVERSIDAD AUTONOMA DE NUEVO LEON



FACULTAD DE CIENCIAS FISICO MATEMATICAS

Diseño Orientado a Objetos

Practica 9

Alumno: Alexis Blanco González

Matricula: 1725357

Carrera: LSTI

Grupo: 007

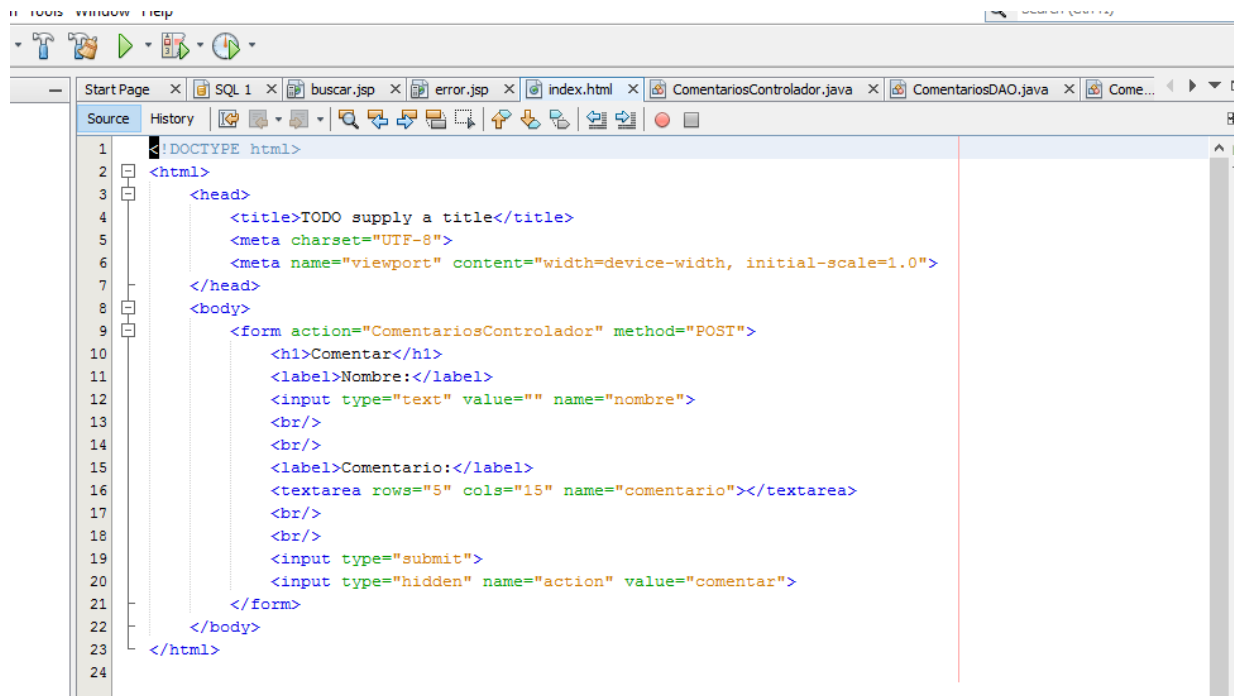
Aula: Lab de DOO

04 de Abril del 2017

Explicación de los códigos de la práctica

Index

Solo se muestra en como si fuera un titulo la palabra Comentar, después se muestra la palabra nombre, después se ve un cuadro de texto donde se introducirá el nombre.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>TODO supply a title</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   </head>
8   <body>
9     <form action="ComentariosControlador" method="POST">
10      <h1>Comentar</h1>
11      <label>Nombre:</label>
12      <input type="text" value="" name="nombre">
13      <br/>
14      <br/>
15      <label>Comentario:</label>
16      <textarea rows="5" cols="15" name="comentario"></textarea>
17      <br/>
18      <br/>
19      <input type="submit">
20      <input type="hidden" name="action" value="comentar">
21    </form>
22  </body>
23 </html>
24
```

Buscar

Es un archivo JSP en donde se busca el resultado que se ingresó en el index, se busca el nombre y el comentario en un cuadro de texto, se envía la información a un servlet, el if es para acomodar la información en una tabla, lo lee en un for donde se busca todos los registros iguales.

```

6 <html>
7 <head>
8 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
9 <title>JSP Page</title>
10 </head>
11 <body>
12 <form action="ComentariosControlador" method="POST" >
13 <h1>Buscar</h1>
14 <label>Nombre: </label>
15 <input type="text" value="" name="nombre">
16 <br/>
17 <br/>
18 <label>Comentario: </label>
19 <textarea rows="5" cols="15" name="comentario"></textarea>
20 <br/>
21 <br/>
22 <input type="submit">
23 <input type="hidden" name="action" value="buscar">
24 </form>
25 <% if(session != null){
26 ArrayList comentarios = (ArrayList)session.getAttribute("comentarios");
27 if(comentarios!=null){
28 %>
29 <table border="1">
30 <tr>
31 <th>Nombre: </th>
32 <th>Comentario: </th>
33 </tr>
34 <%
35 for(Object o : comentarios){
36 ComentariosPOJO comentario = (ComentariosPOJO) o;
37 %>

```

Error

Muestra las palabras Esto no es lo tuyo carnal

```

ile Team Tools Window Help
Search (Ctrl+I)
StartPage x SQL 1 x buscar.jsp x error.jsp x index.html x ComentariosControlador.java x ComentariosDAO.java x Co
Source History
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <title>JSP Page</title>
7 </head>
8 <body>
9 <h1>Esto no es lo tuyo carnal :u (Con un demonio lo que faltaba)</h1>
10 </body>
11 </html>
12

```

ComentarioDAO

En esta clase de java se crea un método para abrir la conexión hacia la base de datos, se crean 3 Strings donde se guarda la dirección de la base de datos, el usuario, contraseña, después el método de cerrar conexión, después el método insertar

laboratorio9 - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Start Page x SQL 1 x buscar.jsp x error.jsp x index.html x ComentariosControlador.java x ComentariosDAO.java x ComentariosPOJO.java x Log.java x web.xml x

Source History

```
42 }
43 }
44
45 public ArrayList buscar(ComentariosPOJO pojo){
46     ArrayList<ComentariosPOJO> comentarios = new ArrayList();
47     try{
48         abrirConexion();
49         String sqlBuscar = "select * from COMENTARIOS where NOMBRE = '" + pojo.getNombre() + "' and COMENTARIO like '%" + pojo.getComentario() + "%'";
50         Statement statement = conexion.createStatement();
51         ResultSet result = statement.executeQuery(sqlBuscar);
52         while(result.next()){
53             String nombre = result.getString("Nombre");
54             String comentario = result.getString("Comentario");
55             ComentariosPOJO coment = new ComentariosPOJO();
56             coment.setNombre(nombre);
57             coment.setComentario(comentario);
58             comentarios.add(coment);
59         }
60         cerrarConexion();
61     }catch(Exception ex){
62     }
63     return comentarios;
64 }
65 }
```

ComentarioPOJO

Esta clase de java se guarda el parámetro de nombre y contraseña para mandarla hacia la base de datos, se crean los métodos set y get de cada parámetro, son de tipo String

laboratorio9 - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search

Start Page x SQL 1 x buscar.jsp x error.jsp x index.html x ComentariosControlador.java x ComentariosDAO.java x ComentariosPOJO.java x Log.java x web.xml x

Source History

```
7
8 /**
9  * Alexis Blanco Gonzalez 1725357
10  */
11
12 public class ComentariosPOJO {
13     private String nombre;
14     private String comentario;
15
16
17     public String getNombre() {
18         return nombre;
19     }
20
21
22     public void setNombre(String nombre) {
23         this.nombre = nombre;
24     }
25
26
27     public String getComentario() {
28         return comentario;
29     }
30
31
32     public void setComentario(String comentario) {
33         this.comentario = comentario;
34     }
35 }
36 }
```

Preguntas

1. ¿Qué ventajas identificas con el uso de un sistema de Logging de eventos?

Puedo llevar un control de que usuario es el que ingresa en ese momento y que es lo que hace.

2. ¿Qué ventajas tienes al utilizar una clase singleton?

Evitan la construcción de múltiples objetos que hacen muchas peticiones y se están creando cada que el usuario haga una petición de ese objeto, solo se utilizara un solo objeto en todo lo que dure la aplicación.

3. ¿Qué "pros" y "contras" identificas al utilizar singleton vs clases estáticas?

Ventajas de usar la clase singleton, es que nos permite utilizar la herencia, el polimorfismo, implementar interfaces y proporcionar diferentes implementaciones, acceso controlado a la única instancia