

## Licenciado en Seguridad en Tecnologías de Información

### Diseño Orientado a Objetos

#### Séptima Práctica en el laboratorio

### **Practica 7**

Lic. Miguel Ángel Salazar Santillán

Grupo: 007 Matrícula: 1732645

Diana Elizabeth Díaz Rodríguez

21/ 03/ 2017

## PRACTICA

En esta práctica, es una de las más difíciles que puedo elaborar, tuve muchas dudas y me ayudaron mucho, aun no estoy segura de haber aclarado todas pero en su momento si lo entendí.

El objetivo de esta práctica es crear una aplicación que pueda conectarse a una base de datos. Esto es donde se almacena información en este caso sería el nombre y la palabra o comentario.

Lo primero que hicimos fue crear un nuevo proyecto con el nombre de Lab7.

Después creamos la base de Datos en JavaDB.

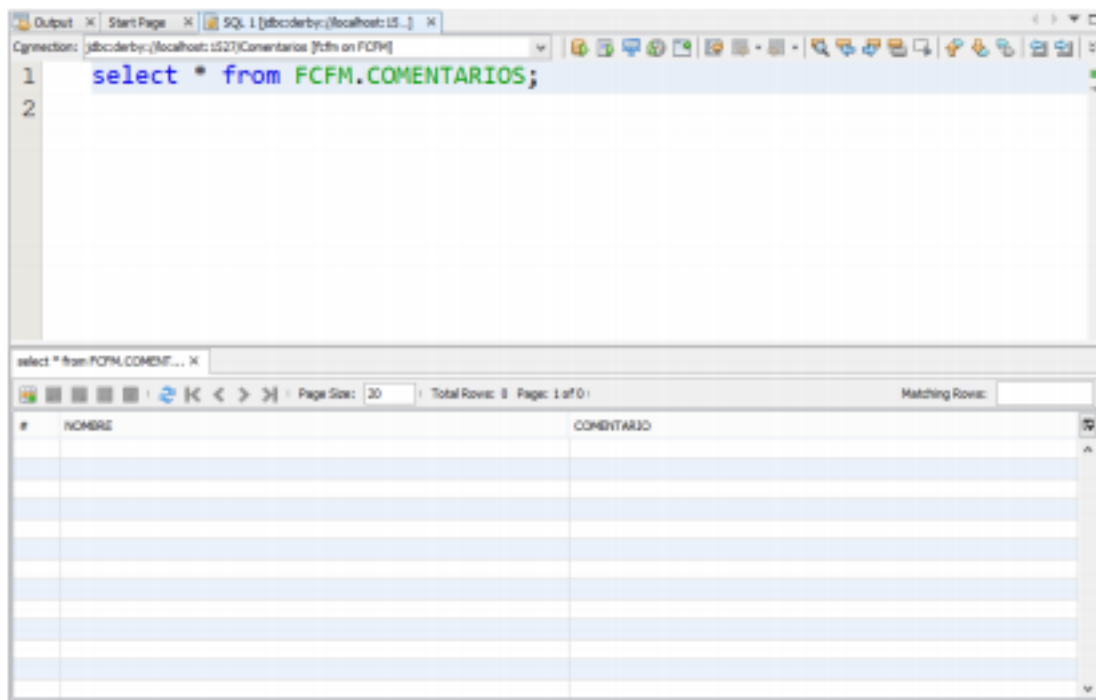
Los datos que tendría serán...

Database name: Comentarios

User name: fcfm

Password: Isti01

Luego lo conectamos que nos mostrara otro esquema llamado FCFM, y luego vendrán varias opciones una de ellas será la Tabla, le insertamos y creamos dos columnas una para nombre y otra para los comentarios, contestando sus especificaciones.



Los cuales se irán guardando todo lo que se vaya insertando cuando la aplicación esté terminada.

El siguiente paso es crear el modelo y una clase para que interactúe con la base de datos. Creamos una clase con para el acceso de los datos con el nombre de DAO y otra clase como objeto de transferencia de los mismos datos.

Con la clase de transferencia de datos recibirá el nombre de ComentariosPOJO que sirve para representar un registro de una tabla de base de datos, que tendrá dos variables de tipo String: Nombre y comentario.

Este es uno de los más sencillos, solo teníamos que hacer los get y los sets de cada uno de ellos.

```
public class ComentariosPOJO {

    private String nombre;
    private String comentario;

    /**...3 lines */
    public String getNombre() {
        return nombre;
    }

    /**...3 lines */
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    /**...3 lines */
    public String getComentario() {
        return comentario;
    }

    /**...3 lines */
    public void setComentario(String comentario) {
        this.comentario = comentario;
    }
}
```

Que olvide mencionar, se encuentra en un paquete llamado modelos.

Ahora sigue la clase de ComentariosDAO, que también se encuentra en el paquete de modelos.

Agregamos una variable de instancia privada con el nombre de conexión e importamos el `java.sql.Connection`.

Y abrimos una conexión, declarando una url para la conexión con la base de datos, declarando usuario y contraseña y que obtenga un DriverManager con el método `getConnection()`... y cerramos la conexión.

```

+ /**...4 lines */
public class ComentariosDAO {
    Log log = Log.getInstance("G:\\lab7\\src\\Text.txt");
    private Connection conexion;

    private void abrirConexion() throws SQLException {
        String dbURI = "jdbc:derby://localhost:1527/Comentarios";
        String username = "fcfm";
        String password = "lsti01";
        conexion = DriverManager.getConnection(dbURI, username, password);
    }

    private void cerrarConexion() throws SQLException {
        conexion.close();
    }
}

```

Después hacemos otro método para insertar() que reciba los ComentariosPOJO, con la intención de que al insertar los datos tiene que pasar por el controlador y abrirá la conexión a la base de datos.

```

public void insertar(ComentariosPOJO dto){
    try{
        abrirConexion();
        String sqlInsert = "insert into COMENTARIOS values ('"+dto.getNombre()+"','"+dto.getComentario()+"')";
        Statement stmt = conexion.createStatement();
        stmt.executeUpdate(sqlInsert);
        cerrarConexion();
    }catch(Exception ex){
        log.write("Ocurrió un error por favor vuelve a intentarlo");
    }
}

```

Y después agregamos un método para buscar que regrese un objeto de tipo list. También devolvamos un ResultSet con el objetivo de almacenar los registros que coincidan con la búsqueda.

```

public ArrayList buscar(ComentariosPOJO dto){
    ResultSet regis;
    ArrayList<ComentariosPOJO> comentarios = new ArrayList();
    try{
        abrirConexion();
        String sqlBuscar = "select from comentario where nombre = '" + dto.getNombre() + "' y comentario '%" + dto.getComentario() + "%'";
        Statement statement = conexion.createStatement();
        ResultSet result = statement.executeQuery(sqlBuscar);

        while(result.next()){
            String nombre = result.getString("Nombre");
            String comentario = result.getString("Comentario");
            ComentariosPOJO coment = new ComentariosPOJO();
            coment.setNombre(nombre);
            coment.setComentario(comentario);
            comentarios.add(coment);
        }
        cerrarConexion();
    }catch(Exception ex){
        log.write("Ocurrió un error por favor vuelve a intentarlo");
    }
    return comentarios;
}

```

Después creamos las paginas para la vista un index.html que ya viene por default, tiene que tener un POST y tiene que enlazar con el ervlet que se llamara ComentariosController.

Enviara el nombre, y el texto para los comentarios con un botón de submit.

```
<body>
  <form action="ComentarioController" method="POST">
    Nombre: <input type="text" value="" name="nombre" placeholder="inserte su nombre">
    Inserte su comentario
    <textarea rows="3" cols="5" name="comentario"></textarea>
    <input type="submit">
    <input type="hidden" name="action" value="buscar">
  </form>
```

Y también se agregara un campo oculto de las acciones y comentarios, es la tabla de base de datos.

```
<% if (session !=null){
    List comentarios = (List) session.getAttribute("comentarios");
    if (comentarios != null){
%>

    <table border="1">
      <tr>
        <th>Nombre</th>
        <th>Comentario</th>
      </tr>
      <%
        for (Object o : comentarios){
          ComentariosPOJO comentario = (ComentariosPOJO) o;
%>
        <tr>
          <td><%=comentario.getNombre() %></td>
          <td><%=comentario.getComentario() %></td>
        </tr>
      <% } %>
    </table>

    <%
    }
  } %>
```

Luego crear un controlador, con el nombre que ya mencione anteriormente con un método de processRequest(), y la acción es comentar, que tengan como parámetros el nombre y el comentario. Agragar un POJO para los datos recibidos desde el formulario, e invocar el método para insertar pasándole el pojo como argumento.

Después de todo el proceso rediccionar la pagina a Buscar.jsp

```

ComentariosPOJO pojo = new ComentariosPOJO();
ComentariosDAO dao = new ComentariosDAO();

HttpSession session = request.getSession();

String action = request.getParameter("action");
String nombre = request.getParameter("nombre");
String comentario = request.getParameter("comentario");

pojo.setNombre(nombre);
pojo.setComentario(comentario);

if(action.equals("comentar")){
    dao.insertar(pojo);
    response.sendRedirect("buscar.jsp");
} else {
}
if(action.equals("search")){
    ArrayList<ComentariosPOJO> comentarios = dao.buscar(pojo);
    session.setAttribute("comentarios",comentarios);
    response.sendRedirect("buscar.jsp");
}

```

Después sigue la acción buscar, con los parámetros del nombre y comentario, construimos un POJO con lo mismo, invocar el método buscar del modelo y agregando la variable tipo List. Si hay un error enviarlo a la página error.jsp.

```

if(action.equals("comentar")){
    dao.insertar(pojo);
    response.sendRedirect("buscar.jsp");
} else {
}
if(action.equals("search")){
    ArrayList<ComentariosPOJO> comentarios = dao.buscar(pojo);
    session.setAttribute("comentarios",comentarios);
    response.sendRedirect("buscar.jsp");
}
else{
}
if(!"comentar".equals(action) & !"buscar".equals(action)){
    response.sendRedirect("error.jsp");
    response.sendRedirect("error.js");
}

```

1	SELECT * FROM FCFM.COMENTARIOS FETCH FIRST 100 ROWS ONLY;	
2		

SELECT * FROM FCFM.COMENT...		
Max. rows: 100   Fetched Rows: 1		
#	NOMBRE	COMENTARIO
1	Diana	Hola

localhost:8080/Lab7/

Aplicaciones Hotmail, Noticias, El C Elizabethdr/DOO1732 Paradise Summerland MisBooks

## Enviar comentario

Nombre:  Inserte su comentario

localhost:8080/lab7/buscar.jsp

Aplicaciones Hotmail, Noticias, El C Elizabethdr/DOO1732 Paradise Summerl

Nombre:  Inserte su comentario

Nombre:	Comentario:
Diana	Hola
Diana	Hola