



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UANL



FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Universidad Autónoma de Nuevo León Facultad de Ciencias Físico Matemáticas

Laboratorio 7

LDOO

Maestro: Miguel Salazar

Alumno:

William Guillermo Villegas Rodríguez

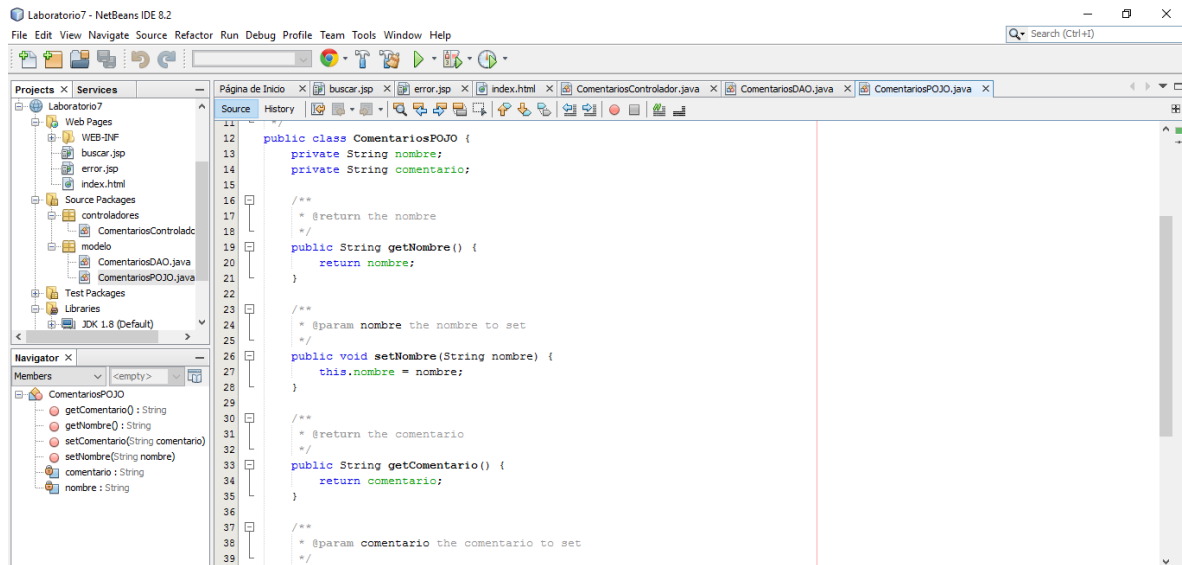
Matricula: 1660542

Actividad 1. Crear una base de datos en JavaDB.

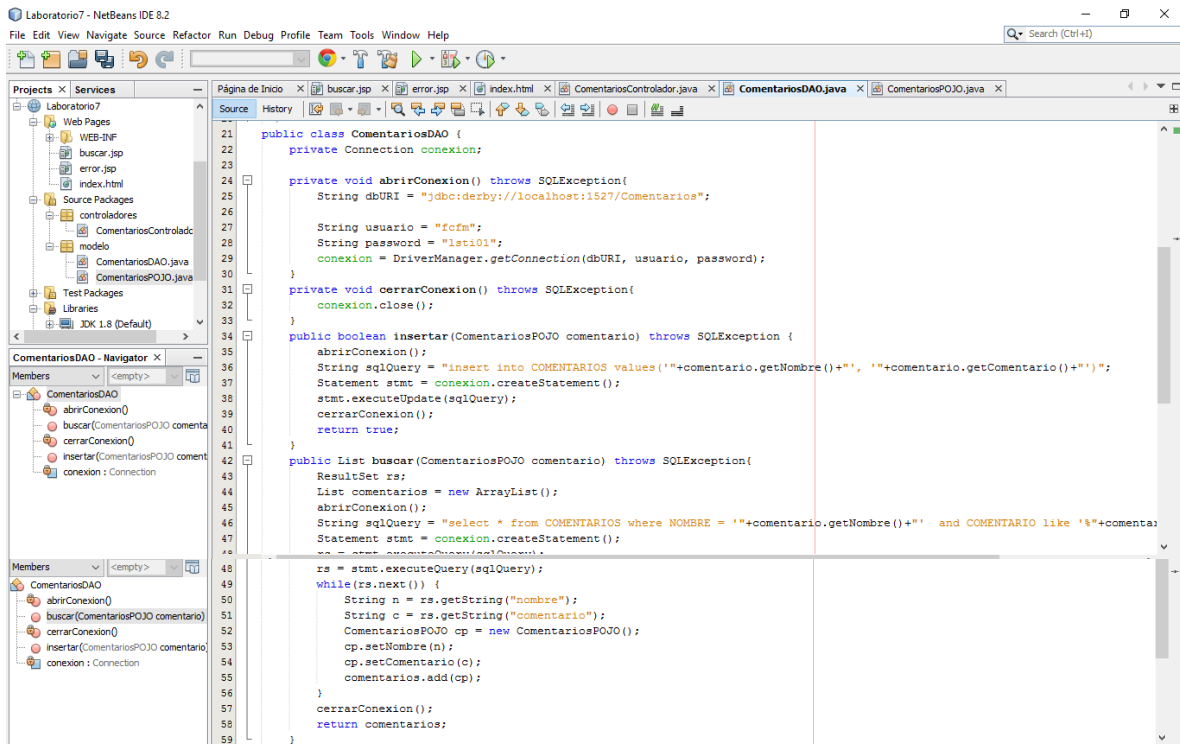
1. En la sección Services del navegador (lado izquierdo) expandir Databases. Debe mostrarse JavaDB que es el RDBMS incluido con NetBeans.
2. Dar clic derecho sobre JavaDB y en Create Database...
3. En el cuadro de diálogo de Create Java DB Database, crear una base de datos con los siguientes datos:
 - a. Database Name: Comentarios
 - b. User Name: fcfm
 - c. Password: Isti01
4. En la línea de conexión, dar clic derecho en Connect... y expandir. Debe aparecer el nuevo esquema llamado FCFM.
5. Dar clic derecho en Tables y en Create Table. La nueva tabla se llamará COMENTARIOS y contendrá dos campos (columnas):
 - a. NOMBRE, de tipo VARCHAR, tamaño 50.
 - b. COMENTARIO, de tipo VARCHAR, tamaño 200.

Actividad 2. Crear el modelo y una clase para la transferencia de datos

1. En el proyecto lab7, crear un nuevo paquete llamado modelo.
2. Dentro del paquete modelo, crear una nueva Java class llamada ComentariosPOJO
3. Agrega dos variables de instancia privadas a la clase ComentariosPOJO, uno por cada campo de la tabla COMENTARIOS de la base de datos:
 - a. Una variable de tipo String llamada nombre que corresponde al campo NOMBRE de tipo VARCHAR en la base de datos.
 - b. Una variable de tipo String llamada comentario que corresponde al campo NOMBRE de tipo VARCHAR en la base de datos
4. Agrega los métodos getters y setters para cada campo.



5. Dentro del paquete modelo, crear una nueva Java class llamada ComentariosDAO
 6. En la clase ComentariosDAO agrega una variable de instancia privada llamada conexion que sea de tipo Connection. El import que se debe agregar es java.sql.Connection.
 7. Agrega un método privado abrirConexion() que no regrese nada y que:
 - a. Declare una cadena con la URI de conexión a la base de datos.
 - b. Declare cadenas con el nombre de usuario y contraseña.
 - c. Obtenga una conexión del DriverManager con el método getConnection().
 - d. Declare que puede lanzar la excepción SQLException.
 8. Agrega un método privado cerrarConexion() que no regrese nada y que invoque al método close() de la conexión.
- En ComentariosDAO agrega un método público insertar() que reciba como argumento un objeto de tipo ComentariosPOJO.
- Agrega un método público buscar() que devuelva un objeto de tipo List (java.util) y que reciba como argumento un objeto POJO. Este objeto POJO tomará los datos que son proporcionados dentro de un formulario de búsqueda que harás más adelante. Los import son del package java.sql



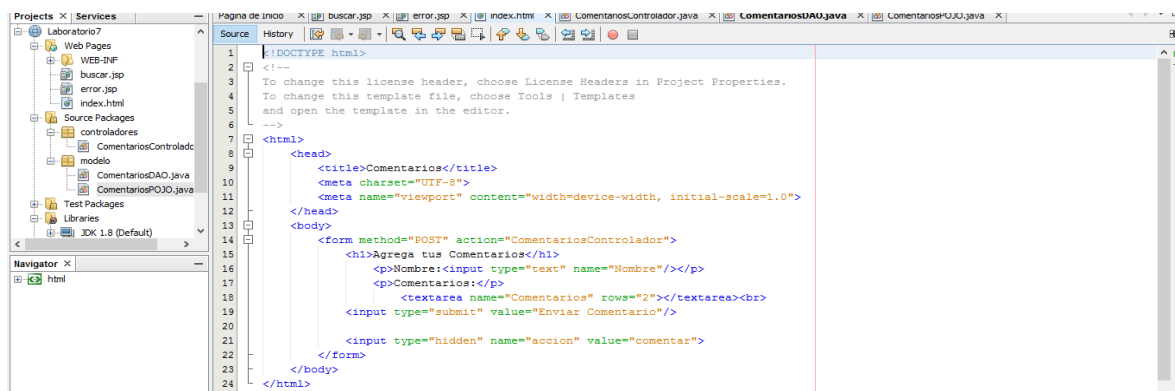
Actividad 3. Crear las páginas HTML/JSP para la vista

1. Abre la página index.html y modifícala para que contenga un formulario para enviar los comentarios:

a. Un formulario HTML que envíe los datos por POST y que contacte a un servlet controlador que se llamará ComentariosControlador.

Dentro del formulario debe incluirse:

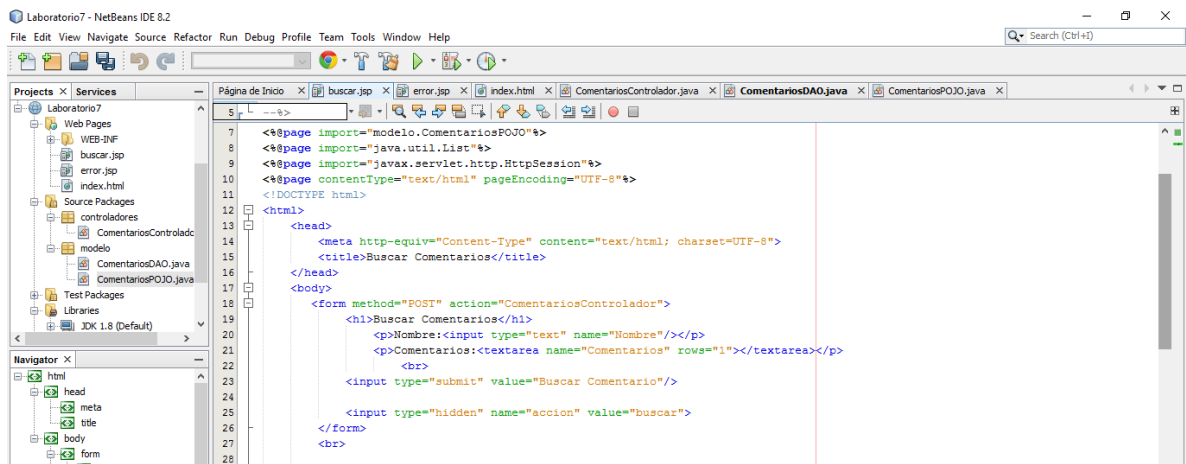
- Un campo de texto para enviar el nombre.
- Un área de texto para enviar los comentarios.
- Un botón de submit.
- Un campo oculto (hidden) de nombre "accion" con el valor "comentar".



En Web Pages agrega una página JSP llamada buscar.jsp

Debe contener:

- a. Un formulario HTML que envíe los datos por POST y que contacte a un servlet controlador que se llamará ComentariosControlador
Dentro del formulario debe incluirse:
 - i. Un campo de texto para enviar el nombre.
 - ii. Un área de texto para enviar los comentarios.
 - iii. Un botón de submit.
 - iv. Un campo oculto (hidden) de nombre "accion" con el valor "buscar".



Después de agregar el formulario, agrega una tabla HTML que se vea como esta

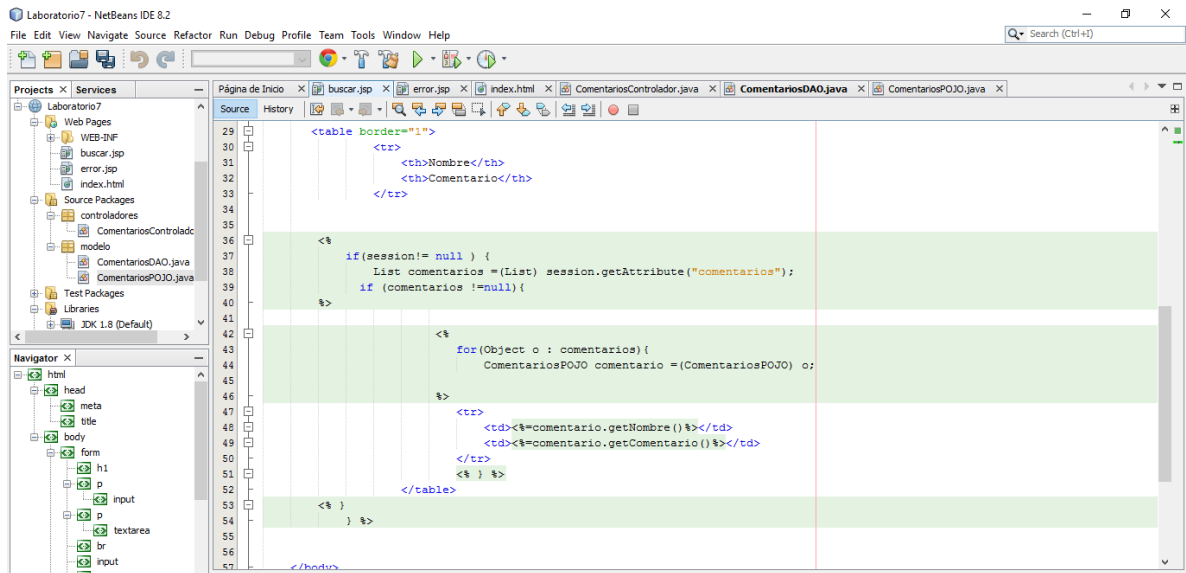
A continuación agregarás antes de iniciar la tabla el código Java necesario para leer los comentarios que provengan de una búsqueda, en caso de que la acción haya sido la de buscar. El código deberá colocarse intercalado con el HTML, cuidando de abrir y cerrar adecuadamente las llaves de JSP: `<% %>`, y las llaves de los bloques de código Java: `{ }`.

a. Si el objeto session no es null:

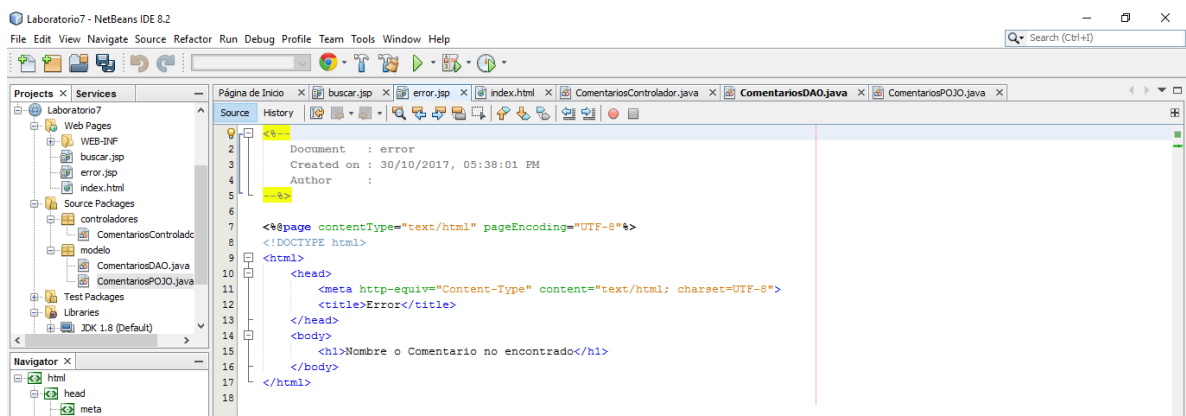
i. Obtener el atributo que llamaremos "comentarios" y asignarlo a un objeto de tipo List. Considera que el objeto se guarda en la sesión como Object, por lo que será necesario hacer un casting a List.

ii. Si el objeto con la lista de comentarios (List) no es null:

1. Hacer un ciclo for que recorra la lista. a. Asignar el objeto obtenido a una variable de tipo ComentariosPOJO



Agrega una página JSP llamada error.jsp que sirva para mostrar errores



Actividad 4. Crear el Controlador

1. Crea un servlet llamado ComentariosControlador.
2. En el método processRequest() lee el parámetro llamado "accion".
3. Si la acción es "comentar":
 - a. Obtén los parámetros del nombre y comentario.
 - b. Construye un objeto del modelo.
- c. Construye un POJO y agrégale los datos recibidos desde el formulario.
- d. Invoca al método insertar() del modelo, pasándole el POJO como argumento.

e. Re-dirige a la página “buscar.jsp”

4. Si la acción es “buscar”: a. Obtén los parámetros del nombre y comentario.

b. Construye un objeto del modelo. c. Construye un POJO y agrégale los datos recibidos desde el formulario.

d. Invoca al método buscar() del modelo, pasándole el POJO como argumento y asignando el valor de retorno a una variable de tipo List.

e. Obtén la sesión del request.

f. Coloca la variable de tipo List (que contiene los registros leídos de la base de datos) en como un atributo de sesión llamado “comentarios”.

g. Re-dirige a la página “buscar.jsp”

5. Si la acción no es ninguna de las dos anteriores, redirige a la página “error.jsp”

