



**Universidad Autónoma de  
Nuevo León  
Facultad de Ciencias Físico  
Matemáticas**



**Unidad de aprendizaje:** Laboratorio de Diseño Orientado a Objetos

**Profesor:** Miguel Salazar

**Alumno:** José Julio Mancha Hernández

**Matricula:**1727375

**Correo Electrónico:** josejulio2119@gmail.com

miércoles, 22 de marzo de 2017

## Reporte #7

Para empezar con mi opinión acerca de la práctica, hablare de que esta es la primera que desarrollo una base de datos y a su vez la primera vez que conviva o funciona a la par de una página Web, para ello fue necesario utilizar el IDE Net Beans Version 8.2 EE además de derbyclient.jar el cual fue descargado de internet el URL de descarga estará en las referencias de este documento una vez echo se siguió con la creación cotidiana del proyecto, una vez teniendo eso lo cual fue relativamente fácil luego de eso se creo la base de datos la cual yo pensaba sería más compleja de desarrollo pero no fue así la manera con la que se trabajó con esta fue muy fácil, esta funciona a través de tablas por nombres “en el caso de NetBeans no sé cómo se trabaje con demás base de datos o lenguajes como SQL” una vez creada nuestra base de datos creamos a lo que yo entendí las clases las cuales servirán para asociar métodos los cuales interactúen en las base de datos por el ejemplo **abrirConexion()** o **cerrarConexion()** los cuales su mismo nombre dice la acción que estos realizaran, las clases que se encargaron de esto fueron DAO y POJO, estos a principio no entendía su funcionamiento pero gracias a un poco de análisis lo entendí, también obviamente se tendrá que construir las vistas o sea el index.html y buscar.jsp además del controller el cual recibe las peticiones de estas hacia al servidor.

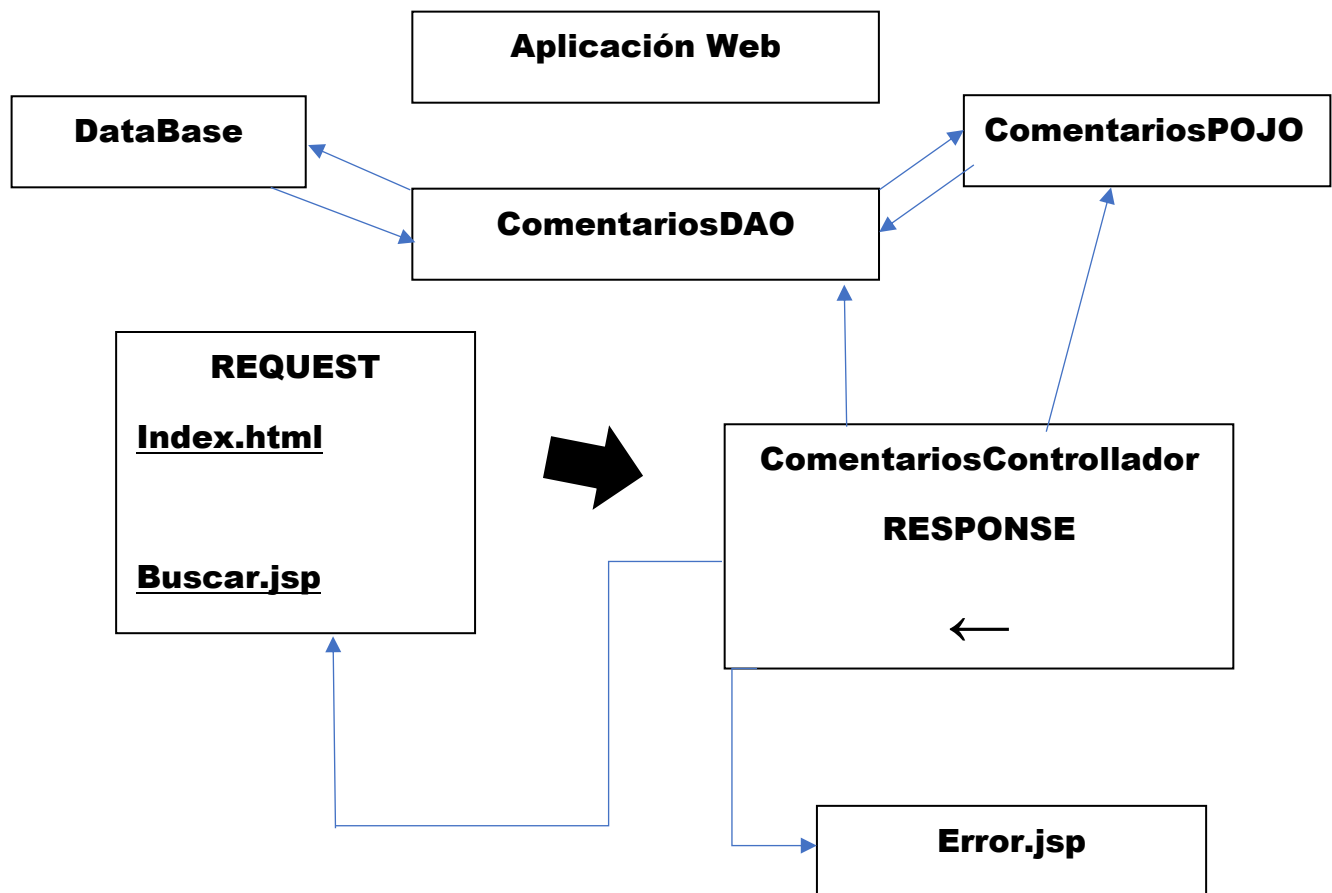
Como esta es la primera vez que realizó un de este tipo me encontré con problemas y dificultades la primera ya la mencione arriba es que no lograba entender el funcionamiento de DAO y para ser más específico los métodos **insertar()** y **buscar()** no entendían la manera en que estos funcionaban una ves resuelto eso el siguiente problema con el cual me enfrente fue la conectividad a través de error 404 el cual se debía a la mala programación dentro del controller y a su vez la sintaxis del mismo.

A través de estos errores también aprendí muchas cosas la principal a emplear base de datos con aplicaciones web y algo como punto extra fue que gracias a la herramienta de Refactor esta da la posibilidad de agregar los métodos **set()** y **get()** sin necesidad de escribirlos, también de que un controller puede recibir peticiones de diferentes jsps o htmls y usar el metodo según del que venga, también el cómo insertar cadenas de caracteres a la misma base de datos y el cómo hacer una búsqueda dentro de la misma a través obviamente desde una aplicación web. Algo importante para mi acerca de esto es que el desarrollo de todo fue muy fácil obviamente es algo pequeño, pero pensaba que esto iba tener mayor complejidad al momento de programarlo. Ya como información extra también NetBeans ofrece amplias opciones para facilitar el desarrollo del código en mucho menor tiempo y forma.

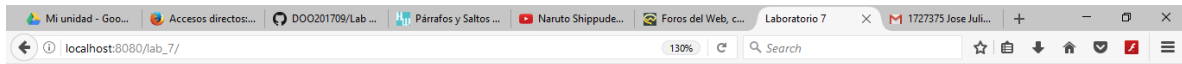
## Preguntas de reflexión

- ¿Cuál piensas que es el propósito de haber hecho una clase DAO en el modelo en lugar de acceder a la base de datos directamente desde el controlador?  
**Obviamente mucha más seguridad para las variables y acceso al mismo a su vez evita el tener que estar creando cada vez un metodo insertar o select facilita la manera de trabajar de este y a la vez brinda seguridad.**
- ¿Para qué sirve un objeto POJO o JavaBean?  
**Sirve para enfatizar el uso de clases simples y que no depende de framework en especial.**
- En caso de que los comentarios fueran muchos (digamos, cientos o miles) sería impráctico mostrarlos todos en una misma página. Generalmente los sitios de búsqueda (como Google) usan una técnica llamada “paginación”, para ir mostrando solo cierta cantidad de registros cada vez. Describe cómo harías esa paginación en esta aplicación (cuál es la lógica que seguirías en el programa).  
**A través de una serie de un tag <section> que nos sirva para poner una listas cada una de estas llevara a una página diferente de html en las cuales estará la información almacenada**
- Cuando se muestra la tabla con los resultados de la búsqueda, desaparecen los valores de los campos de búsqueda. ¿Qué harías para que se sigan mostrando?  
**Usar cookies para guarde los strings**
- Haz una búsqueda pero ahora, en lugar de escribir un nombre, escribe lo siguiente en el campo de búsqueda de nombre (la comilla inicial es importante, y también los dos guiones al final):  
`' or 1=1 --`  
¿Cuál fue el resultado de la búsqueda?  
**Muestra la información de toda la base de datos.**
- A lo que hiciste en la pregunta anterior se le conoce como *SQL Injection (SQLi)*, y es una de las vulnerabilidades más explotadas en las aplicaciones Web. De acuerdo a la cadena de búsqueda y a los resultados obtenidos, explica qué fue lo que ocurrió.  
**Este género un error a través de una igualación errónea provocando true erróneo o falses verdaderos.**

- ¿Cómo piensas que puede evitarse un SQL injection como el de la pregunta 4? (A estas alturas del curso no se vale responder “no sé” a una pregunta así).  
**A través de validaciones en todas las clases del programa o por lo menos en las principales.**
- Elabora un diagrama donde muestres todos los elementos que construiste en esta práctica y cómo están relacionados entre ellos.



# Capturas de pantalla



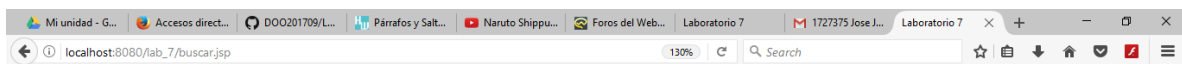
## Bienvenido

Si desea enviarnos algun comentario, ingrese los campos

Nombre:

Comentarios:

Activar Windows  
Ve a Configuración para activar Windows.



## Buscar

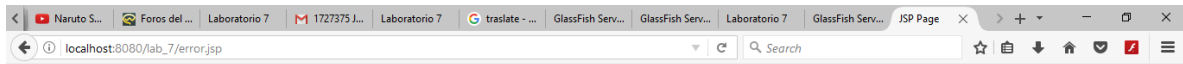
Nombre:

Comentarios:

Activar Windows  
Ve a Configuración para activar Windows.







**La pagina ha tenido un error. Porfavor intenta introducir cadenas de palabras  
O caracteres validos**

Activar Windows  
Ve a Configuración para activar Windows.

