



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

**UANL**



**FCFM**

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

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

## **Laboratorio 4**

### **LDOO**

**Maestro: Miguel Salazar**

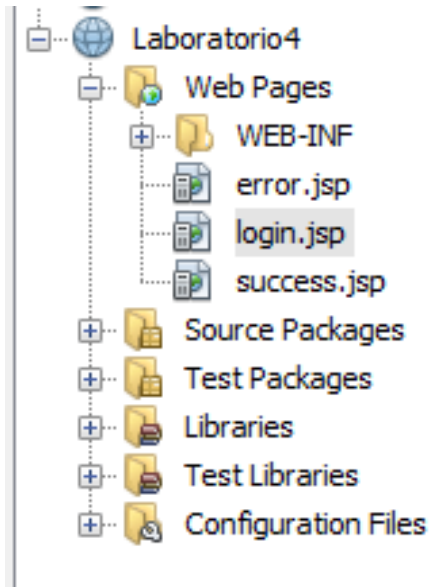
**Alumno:**

**William Guillermo Villegas Rodríguez**

**Matricula: 1660542**

## Crear un inicio de sesión simple con MVC

- 1.- Crear un proyecto de tipo Java Weben NetBeans.
- 2.- Eliminar el archivo index.html que crea por defecto NetBeans y que se encuentra dentro de la carpeta Web Pages.
- 3.- Tres de Agrega Archivos de tipo JSP: login.jsp, success.jspy error.jsp.



4-en la página login.jsp le pondremos:

Un título h1 con un mensaje de bienvenida.

Un formulario.

El formulario debe ejecutar una LoginController acción.

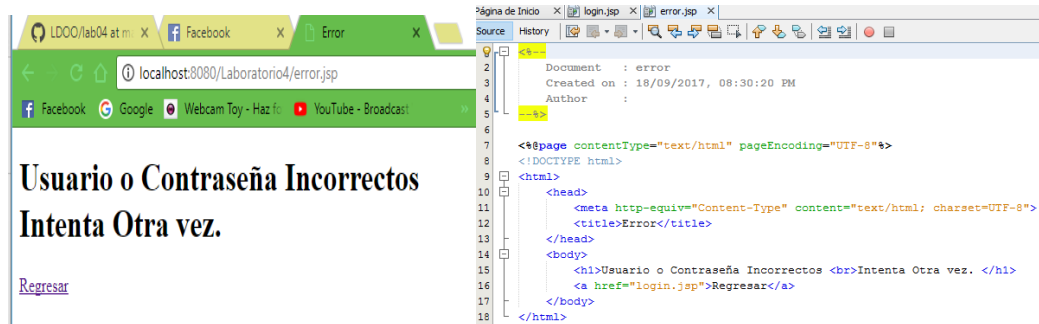
El formulario debe enviar los datos mediante el método de transferencia POST.

El nombre justo contiene tres input: uno de tipo text, el segundo de tipo password y el tercero de tipo submit con el valor Iniciar sesión.

Y el código y la paina se deben ver así



5.- La página error.jsp debe contener un título h1 con el contenido Usuario o contraseña incorrectos. Además, un enlace con el atributo href para reedireccionar hacia login.jsp.



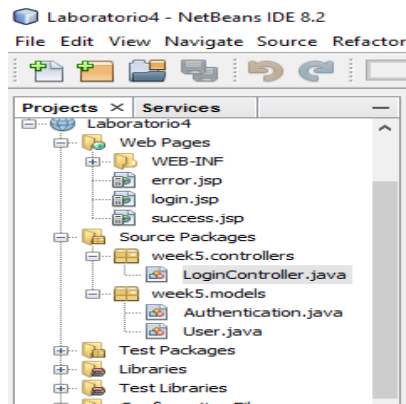
Después vamos a crear un servlet de nombre LoginController. Durante la creación del servlet debe especificarse que sea creado dentro del paquete week5.controllers.

Para iniciar el trabajo con el código del servlet eliminaremos el contenido del método processRequest.

Dentro del servlet, en el método processRequest, se debe obtener los parámetros enviados por el cliente haciendo uso del método request.getParameter

Ahora vamos a crear el modelo antes de seguir modificando las páginas anteriores

Crear una clase de Java con nombre User que funcionará como modelo. La clase debe estar dentro de un paquete nuevo de nombre week5.models



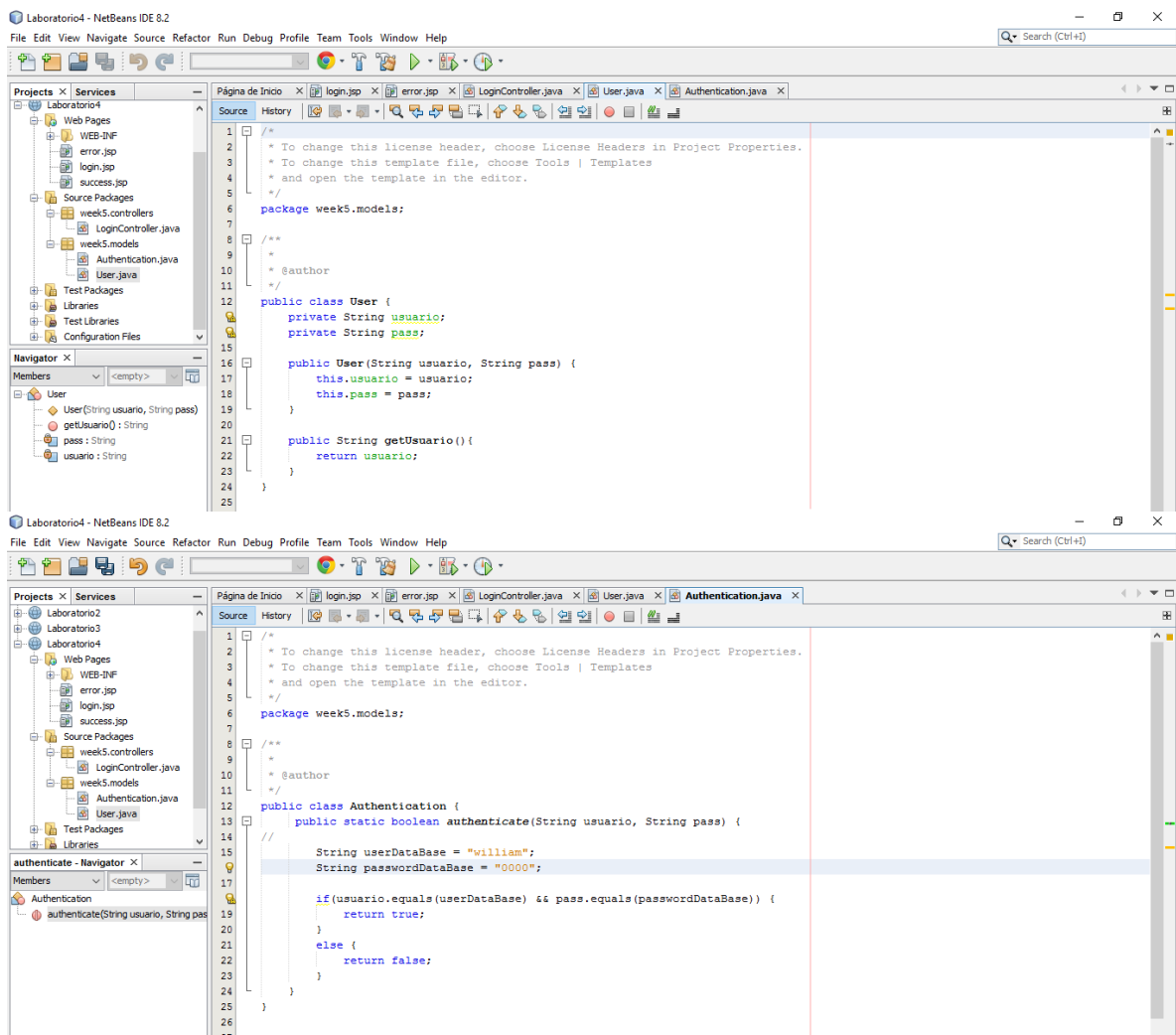
La User única fuente

- Dos propiedades privadas de tipo String: username y password.
- Un constructor que recibe como parámetro de entrada dos String. El primero representa el usuario y el segundo la contraseña. En el cuerpo del constructor, establecer el valor de los parámetros de entrada hacia las propiedades privadas username y password.

- Un método de nombre `getUsername` que devolverá `String`, es decir, el valor de la variable privada `username`.

3.- Crear una segunda clase `Authentication`, dentro del mismo paquete `week5.models`, como se puede contener:

- Un método estático de nombre `authenticate` que devuelva un tipo `boolean`, el cual nos indicará si el usuario está o no autorizado.
- En el método `authenticate` definirá dos variables de tipo `String`: `userDataBase` y `passwordDataBase` que actuará como información *dummy*, es decir, información falsa y *hardcoded* para emular que obtiene esa información de la base de datos. Esta información es *ficticia*, no servirá para comparar la información de "la base de datos" contra lo que capturó el usuario.
- La validación de la realización a través del método `equals`.



Una vez con las vistas, el controlador y los modelos, podemos continuar el código de nuestro controlador (servlet).

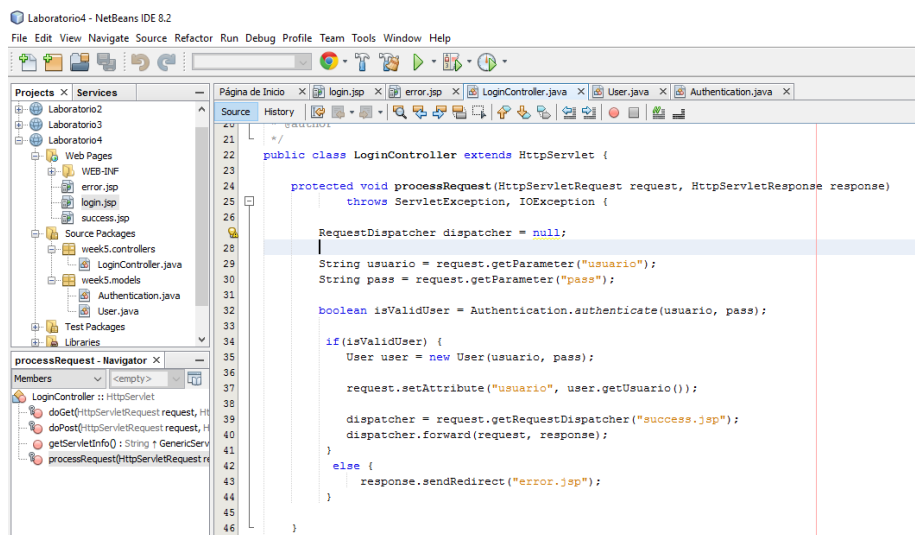
En nuestro controlador, haremos uso de nuestros modelos. Para utilizarlos, debemos importar el paquete `week5.models.*`. Además, importaremos la clase `RequestDispatcher` que está dentro del paquete `javax.servlet`:

Una vez importadas las clases necesarias dentro de su controlador `LoginController`, construiremos las siguientes lógicas en el método `processRequest`:

Recuperar los valores enviado por el cliente, descrito en la Actividad 2.

Invocar el método `authenticate` de la clase `Authentication`, descrito en la Actividad 3. Enviaremos la información de los valores obtenidos del request.

Si el método `authenticate` devuelve `true` se debe construir una instancia del modelo `User` y enviar un valor a la página `login.jsp`. Si el método devuelve `false` se debe redirigir a la página `error.jsp`.



Una vez creada la lógica del `LoginController`, ajustaremos la vista `success.jsp` de esta manera

