



Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Lic. En Seguridad de Tecnologías de la Información

Práctica 4

Alumno: Roberto Iván Hernández Chavarría

Matricula: 1452359

Materia: Diseño Orientado a Objetos

Profesor: Lic. Miguel Ángel Salazar Santillán

Laboratorio 4

Para esta práctica, realizaremos una actividad en la que utilizaremos el MVC, usando archivos .jsp, una nos servirá como página de login, las otras 2, una llamada success, a la cual accederemos si el login fue correcto, y otra llamada error, a la cual iremos si el login fue equivocado.

En la página de login, agregamos un campo de texto normal para el nombre, una de tipo password para la contraseña y un submit, en la página de error, agregamos un párrafo, en donde indicamos que el nombre o contraseña son inválidos, y una referencia a la página de login, para re direccionar hasta ella. En la página de success, agregamos un párrafo en donde indicamos que el inicio fue correcto y agregamos un saludo.

```
Logincontroller.java x User.java x error.jsp x login.jsp x succes.jsp x
source History
4 | Author : Roberto iván hernandez chavarria 1452359
5 | -->
6 |
7 | <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 | <!DOCTYPE html>
9 | <html>
10 | <head>
11 | <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 | <title>login</title>
13 | </head>
14 | <body>
15 | <h1>Login</h1>
16 |
17 | <form action="logincontroller" method="post">
18 | Usuario <input type="text" id="Username">
19 | Contraseña <input type="password" id="Password">
20 | iniciar sesión <input type="submit" id="enviar">
21 |
22 | </form>
23 | </body>
24 | </html>
```

```
Logincontroller.java x User.java x error.jsp x login.jsp x succes.jsp x
source History
1 | <%--
2 | Document : succes
3 | Created on : 15/09/2017, 08:30:49 PM
4 | Author : Roberto iván hernandez chavarria 1452359
5 | -->
6 |
7 | <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 | <!DOCTYPE html>
9 | <html>
10 | <head>
11 | <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 | <title>Inicio de sesión válido</title>
13 | </head>
14 | <body>
15 |
16 | <h1>Inicio de sesión válido</h1>
17 |
18 | <h2>Hola <%= request.getAttribute("usuario") %></h2>
19 | </body>
20 | </html>
```

```

Source History
1 <!--
2 Document : error
3 Created on : 15/09/2017, 08:31:24 PM
4 Author : Roberto iván hernandez chavarria 1452359
5 -->
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>error</title>
13 </head>
14 <body>
15 <h1>Usuario o contraseña incorrectos </h1>
16 <a href="login.jsp"> Regresar</a>
17
18 </body>
19 </html>
20

```

Después creamos un servlet llamado “login controller”, en un paquete llamado “week5.controllers”, eliminamos el contenido creado por default e importamos unos modelos llamados “user” y “authetication”. Después, creamos una clase de java llamada user, agregamos los parámetros de tipo string de “usuario” y “contraseña”, los cuales obtendrá los datos de login mediante un método “getUsername”.

```

RequestDispatcher dispatcher = null;

//Se recupera la información del request
String txtUsername = request.getParameter("txt-username");
String txtPassword = request.getParameter("txt-password");

//Invoca al método authenticate para validar el usuario
boolean isValidUser = Authentication.authenticate(txtUsername, txtPassword);

if(isValidUser) {

    //Construye instancia del modelo User
    User user = new User(txtUsername,txtPassword);

    //Construye parámetro para enviar a la vista success.jsp
    request.setAttribute("username", user.getUsername());

    //Se define a donde se enviará el objeto request y response.
    dispatcher = request.getRequestDispatcher("success.jsp");
    dispatcher.forward(request, response);

} else {
    //Si el usuario es inválido, reedireccionamos la petición hacia error.jsp
    response.sendRedirect("error.jsp");
}
}

```

```

1 package week5.models;
2
3 /*
4  * To change this license header, choose License Head
5  * To change this template file, choose Tools | Templ
6  * and open the template in the editor.
7  */
8
9 /**
10  *
11  * @author Roberto Iván Hernández Chavarria 1452359
12  */
13
14
15 public class User{
16     private String username;
17     private String password;
18 }

```

También, en la misma carpeta, creamos una clase llamada “authenticate”, el cual tendrá un método estático que devuelve un tipo “boolean” el cual será el resultado de si el usuario puede realizar el login o no, en esta parte, agregamos datos de tipo “dummy” de usuario y contraseña, los cuales interpretarán a una base de datos (que no implementaremos) y los cuales servirán para autenticar al usuario mediante el login. igualmente, aquí es donde implementaremos el redireccionamiento hacia la página de éxito o de error.

```

    *
    * @author Roberto Iván Hernández Chavarría 1452359
    */
    public class Authentication {

        public static boolean authenticate(String username, String password) {

            //Variables *dummy*.
            //Se hardcodean los valores "Miguel" y "MiPassword" para emular que se obtu
            String userDataBase = "Roberto";
            String passwordDataBase = "contraseña";

            //username.equals(userDataBase) realiza una comparación entre las cadenas u
            //Si son iguales devuelve true. Si son diferentes devuelve false.
            if(username.equals(userDataBase) && password.equals(passwordDataBase)) {
                return true;
            }
            else {
                return false;
            }
        }
    }
}
```

Después implementamos la aplicación, ingresando a <http://localhost:8080/week5/login.jsp> a partir de aquí, ingresamos nuestros datos correctamente, para poder ingresar a succes.jsp, y podemos regresarnos a login.jsp, ingresar nuestros datos incorrectos e ingresar a error.jsp.

Preguntas de reflexión:

- 1.- una ventaja de usar MVC, es que ya no necesitamos repetir los modelos o interfaces en cada una de las páginas que desarrollemos, ya que podemos exportar los datos a cada una, ahorrándonos tiempo de desarrollo.
- 2.-para agregar más vistas, tendríamos que crear más páginas de tipo jsp
- 3.-para agregar más funciones, tendríamos que crear más métodos o acciones en el mismo servlet, ya que nos ahorraría tiempo y no tendríamos problemas al mandar llamar a otros servlets, solamente especificaríamos que métodos necesitamos.
- 4.-diagrama de este ejemplo

