

Laboratorio 4

Maestro

Miguel Ángel Salazar

Matricula

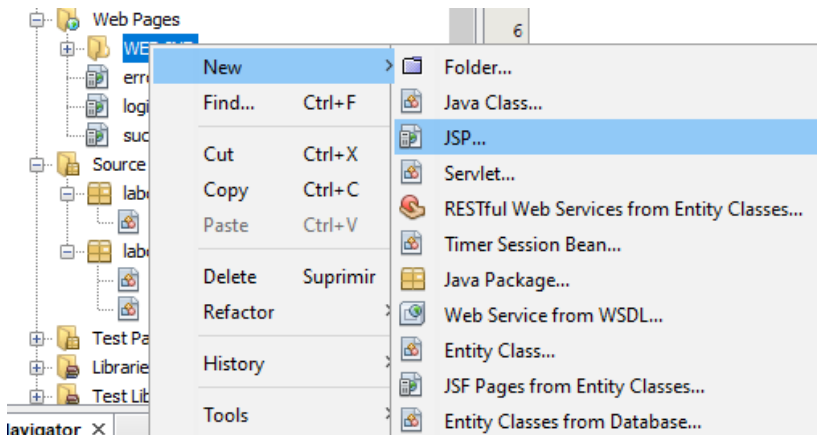
1702824

Alumno

Fernando Martínez Bautista

Reporte de laboratorio 4

Se inicia iniciando 3 archivos nombrados login, error y succes



Iniciaremos con el archivo de login como no es un archivo html se debe de especificar al inicio con:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
```

Se crea un archivo normal de HTML en la que se van a solicitar un usuario y una contraseña

localhost:8080/laboratorio4/login.jsp

Bienvenido

Usuario:

Password:

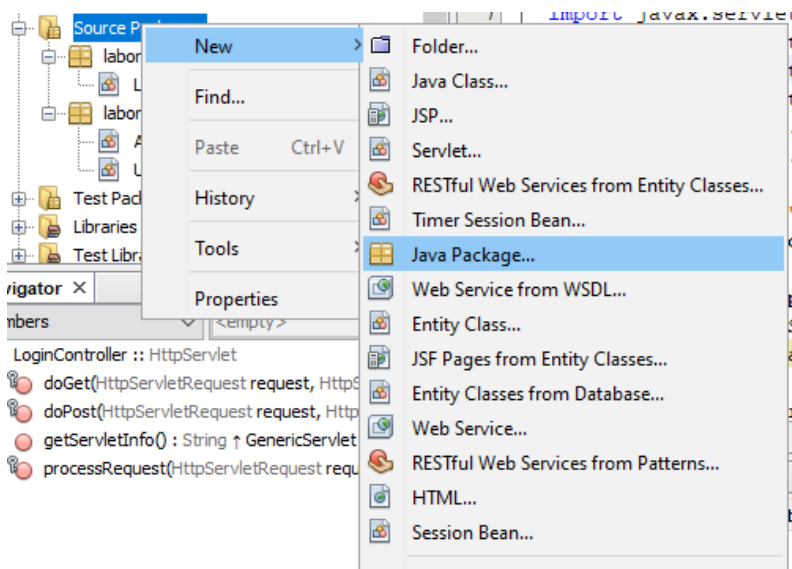
Esto se realiza de la siguiente forma:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Inicio de sesión</title>
</head>
<body>
<center>
  <h1>Bienvenido</h1>
  <form action="LoginController" method="post" name="login">
    <label>Usuario: </label><input type="text" name="usuario"><br><br>
    <label>Password:</label><input type="password" name="password"><br><br>
    <br><input type="submit" value="Iniciar sesion">
  </form>
</center>
</body>
</html>
```

En el form se utilizará el método action para poder solicitar al archivo de LoginController

```
<form action="LoginController" method="post" name="login">
```

Crearemos en Source Packages controllers en el cual crearemos un servlet llamado LoginController



Ya creado importaremos

```
import laboratorio4.models.Authentication;  
import laboratorio4.models.User;
```

Borraremos el código creado hasta response.setContentType

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html;charset=UTF-8");  
    try (PrintWriter out = response.getWriter()) {  
        /* TODO output your page here. You may use following sample code. */  
        out.println("<!DOCTYPE html>");  
        out.println("<html>");  
        out.println("<head>");  
        out.println("<title>Servlet NewServlet</title>");  
        out.println("</head>");  
        out.println("<body>");  
        out.println("<h1>Servlet NewServlet at " + request.getContextPath() + "</h1>");  
        out.println("</body>");  
        out.println("</html>");  
    }  
}
```

Y agregaremos lo siguiente
`RequestDispatcher dispatcher= null;`

Y utilizaremos request.getParameter para llamar el usuario y la contraseña de login.jsp

```
String usuario = request.getParameter("usuario");  
String password = request.getParameter("password");
```

Ahora validaremos que el usuario y la contraseña se encuentren correctos

```
boolean isValidUser = Authentication.authenticate(usuario, password);
if (isValidUser){
    User user = new User (usuario, password);
    request.setAttribute("username", user.getUsername());
    dispatcher = request.getRequestDispatcher("success.jsp");
    dispatcher.forward(request, response);
}
else {
    response.sendRedirect("error.jsp");
}
```

Crearemos otro paquete llamado models y creamos un archivo java llamado authentication en el cual se establecerá como usuario y contraseña de la siguiente forma:

```
public class Authentication {
    public static boolean authenticate(String username, String password){
        String userDataBase = "Miguel";
        String passwordDataBase = "MiPassword";

        if(username.equals(userDataBase) && password.equals(passwordDataBase)){
            return true;
        }
        else{
            return false;
        }
    }
}
```

Y por último crearemos otro archivo java llamado user

```
package laboratorio4.models;
```

```
/**...4 lines */
```

```
public class User {

    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }
}
```

```
/** Get the value of username ...5 lines */
```

```
public String getUsername() {
    return username;
}
```

```

/** Set the value of username ...5 lines */
public void setUsername(String username) {
    this.username = username;
}

/** Get the value of password ...5 lines */
public String getPassword() {
    return password;
}

/** Set the value of password ...5 lines */
public void setPassword(String password) {
    this.password = password;
}

```

Ya con estos pasos se validaron el usuario y contraseña, y crearemos otro archivo jsp en caso de usuario o contraseña se encuentran mal

```

<%--
Document    : error
Created on  : 19/09/2017, 02:20:40 AM
Author      : hifum
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Error</title>
    </head>
    <body>
        <center>
            <h1>Usuario o Password incorrectos</h1>
            <a href="login.jsp">Regresar</a>
        </center>
    </body>
</html>

```

En caso de ser así no da un link en el cual podemos regresar a la página principal

Y en succes accedara al inicio de la sesión

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Inicio de sesion valida</title>
</head>
<body>
<h1>Inicio de sesion valida</h1>
<h2>hola <%= request.getAttribute("username") %> </h2>
</body>
</html>
```

¿Qué ventajas tienes al usar un patrón MVC en lugar de que un servlet realice todo?

Se dividen la lógica del diseño , haciendo que el proyecto sea mas escalable

En el caso donde hay varios desarrolladores, el seguir métodos comunes de programación hace que el código sea mas entendible entre todos

Si quieres agregar más vistas, ¿Qué cambios tendrías que hacer?

Tendríamos que crear las pagina jsp, y dependiendo de que tipo de funciones vaya a realizar tendremos que agregar mas controles y modelos

Si requiriéramos más funcionalidades y no solamente un login de acceso, ¿crees que un solo servlet o una sola clase de modelo serían suficientes?

No, inicialmente por que la primicia del modelo MVC indica que debe de ser mas accesible a modificaciones

Crea un diagrama para este ejemplo basado en el diagrama habitual de MVC

