



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM

UNIDAD DE APRENDIZAJE: LABORATORIO DE DISEÑO ORIENTADO A OBJETOS.

PRACTICA #6

PROFESOR. MIGUEL ÁNGEL SALAZAR S.

ALUMNA. VALERIA MARTÍNEZ DE LA ROSA

MATRICULA. 1678575

San Nicolás de los garza, nuevo león a 14 de marzo de 2017

MANEJO DE COOKIES:

Crear una página web que sirva como patrón mvc, y haga uso de sesiones y cookies.

1. Crea un proyecto en java web-web application, llamado laboratorio6.
2. En la carpeta web pages, elimina el archivo html, que se crea por ende.
3. Crear 3 jsp, "login", "error" y "profile".
4. Crear 3 servlet's, en source packages, new servlet, en package se escribe laboratorio6.Controllers, y el nombre de cada servlet, "LogoutController", "LoginController", "ProfileController".
5. Crear 2 archivos java, en source packages, new java class, en package escribe laboratorio6.Models, el nombre de c/u es "Authentication", "User".

VIEW:

Login: tiene una bienvenida y un formulario, en el form tiene un action a LoginController, con el método post. Deberá contener 3 input's, 1 tipo text, otro tipo password, y el otro tipo submit.

```
<h1>Bienvenido</h1>
<form action="LoginController" method="post">
  <label>Usuario:</label>
  <input type="text" name="txt-username"><br>
  <label>Contraseña:</label>
  <input type="password" name="txt-password"><br>
  <input type="submit" value="Iniciar Sesión">
</form>
```

Haremos uso de código de java, extrayendo el atributo de la sesión, que debe ser diferente de nulo, y si esto es así re-direcciona a la página de profile.

```
<%
    if(session.getAttribute("username" ) != null ){
        response.sendRedirect("profile.jsp");
    }
%>
```

Error: contiene un mensaje de error, y un link que re-direcciona a la pagina de login.

```
<body>
  <h1>Usuario o contraseña incorrectos</h1>
  <a href="login.jsp">Regresar</a>
</body>
```

Profile: en la etiqueta del body, se escribe style= "background-color": <%= color %>, dónde color es el nombre de la cookie que se creó en profileControlle. Form tiene un action a ProfileController, contiene una bienvenida y concatena el nombre, que se extrae del atributo de sesión. Después muestra los datos del usuario, que igual, se extraen del atributo de la sesión.

Tiene un select, que es dónde se seleccionara el color; cada option tiene como value el color que se desea que este en elección, tomando en cuenta que deben estar escritos en ingles, y se cierran los option, en medio de estos deberá tener el valor que muestre a seleccionar. Se cierra el select.

Tiene un botón de enviar, de tipo submit. Finalmente tiene un link que hace referencia a LogoutController, para cerrar sesión.

```
<body style="background-color: <%= color %>" >
  <form action = "ProfileController">
    <h1> Bienvenido <%= session.getAttribute("nombre") %> </h1>
    <h1>nombre: <%= session.getAttribute("nombre") %></h1>
    <h1>Usuario: <%= session.getAttribute("username") %></h1>
    <label>Color:</label>
    <select name = "select">
      <option name="" value="purple">morado</option>
      <option name="" value="blue">azul</option>
      <option name="" value="pink">rosa</option>
      <option name="" value="black">negro</option>
    </select>
    <input type = "submit" value = "Guardar"/>
    <h2><a href="LogoutController">Cerrar Sesión</a></h2>
  </form>
```

Se agregara código de java.

```

<%
    if(session.getAttribute("username") == null){
        response.sendRedirect("login.jsp");
    }
    String color= "";

    Cookie cookie ;
    Cookie[] cookiesArray ;

    cookiesArray = request.getCookies();
    for (int i = 0; i < cookiesArray.length; i++){
        if(cookiesArray[i].getName().equals("color")){
            cookie = cookiesArray[i];
            color = cookie.getValue();
        }
    }
%>

```

Compara que el atributo username sea diferente de nulo, si esto es así entonces te re-direcciona a la pagina de logging.

Se crea una cadena llamada color, después se crea una instancia del objeto cookie, y después un arreglo de cookies.

En el arreglo de cookies, se guarda el método request.getCookies(). Después con un ciclo for crea nuevas cookies, compara si el arreglo de cookies es igual a la cookie llamada color(que se crea en profileController), y el arreglo de cookies se guarda en la cookie, y se obtiene el valor de la cookie y se guarda en la cadena color.

MODELS:

User: tiene 3 variables privadas de tipo String, llamadas, username, password, y nombre. Como estas son privadas, otras clases no pueden hacer uso de estas, así que crearemos un método "getVariable" para cada una de estas, deben ser públicos y devolver un String.

```

public class User {
    private String username ;
    private String password ;
    private String nombre;

    public User() {
        this.username = "vale.mdlr";
        this.password = "pass123";
    }

    public String getUsername() {
        return this.username;
    }
    public String getPassword(){
        return password;
    }
    public String getNombre(){
        return nombre = "Valeria";
    }
}

```

Authentication: crea un método que es público, estático y booleano, llamado authenticate con dos parámetros, en su firma crea una instancia de la clase User. Hace una comparación, si el método getUsername es igual a la variable username(que es uno de los parámetros del método authenticate), y simultáneamente el método getPassword es igual a la variable password, entonces devuelve un true; sino un false.

```

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

    User user = new User();
    if (user.getUsername().equals(username) && user.getPassword().equals(password) ){
        return true;
    } else {
        return false;
    }
}

```

CONTROLLERS:

LoginController: este controlador, por defecto tiene un método de processRequest, su firma la eliminaremos y después proseguiremos a re-estructurar la firma.

Se crean dos instancias, una de la clase User(), y otra de HttpSession().

Se crean dos variables de tipo cadena, en las que se guarden los datos que introducen en el login, con el método request.getParameter("atributo");, se crea un método booleano llamado isValidUser en dónde se guarda el método authenticate de la clase Authentication(se crea una instancia del mismo para acceder a este).

Comparar si isValidUser, es correcto, entonces obtiene los atributos de la sesión, del método getUsername y getPassword, de tipo user, y te re-dirige a la página de profile; en caso de no ser correcta la información entonces te re-dirige a la página de error.

```
HttpSession session = request.getSession();

String txtUsername = request.getParameter("txt-username");
String txtPassword = request.getParameter("txt-password");

boolean isValidUser = Authentication.authenticate(txtUsername, txtPassword);

if(isValidUser) {

    User user = new User();

    session.setAttribute("username", user.getUsername());
    session.setAttribute("nombre", user.getNombre());

    response.sendRedirect("profile.jsp");
} else {
    response.sendRedirect("error.jsp");
}
```

LogoutController: este controlador, por defecto tiene un método de processRequest, su firma la eliminaremos y después proseguiremos a re-estructurar la firma.

Se crea una instancia del objeto HttpSession en donde se guardara request.getSession().

Se utiliza la variable del objeto HttpSession e invalida la sesión.

Finalmente te re-direcciona a la página de login.

```
HttpSession session = request.getSession();  
session.invalidate();  
response.sendRedirect("login.jsp");
```

ProfileController: este controlador, por defecto tiene un método de ProcessRequest, su firma la eliminaremos y después proseguiremos a re-estructurar la firma.

Crea una instancia del objeto cookie llamada color, en donde se guarda una cookie que se crea, dentro de los parámetros de la cookie se escribe el nombre de la cookie entre comillas, una coma, después request.getParameter("nombreDelSelect"), después te responde con el añadir la cookie llamada color, y finalmente lo redirige a la página de profile.

```
Cookie color = new Cookie("color", request.getParameter("select"));  
response.addCookie(color);  
response.sendRedirect("profile.jsp");
```

Preguntas:

¿Qué ventajas tienes al hacer manejo de cookies en la aplicación Web?

No se almacenan en el servidor, por lo que no consumen recursos. Además guarda los cambios hechos por el usuario en la vista de la app.

¿Qué problemas potenciales pueden presentarse al manejar las cookies?

Las cookies al guardar información personal, se pueden hacer robo de cookies lo que implica el robo de la información almacenada en esta.

¿Cómo funciona el for(Cookie c : cookies) y para que se utiliza su if interno?

El arreglo de cookies toma el valor de todas las cookies almacenadas, y después crea un arreglo de cookie para cada uno de estos valores, y lo guarda ahí. Y si el

arreglo de cookie es igual a la cookie de profileController, entonces el arreglo lo almacena en la variable cookie, se obtiene el valor de esta y se guarda en una variable string (qué se utiliza para el body).

¿Sería conveniente almacenar el valor del color en una sesión?

No, las sesiones a diferencia de las cookies, al salir de la app, se borra toda información, entonces por ende, esta se eliminaría y no haría lo que deseamos.

¿Sería conveniente almacenar el nombre de usuario en una cookie?

Si y no, por seguridad esto no debería porque implica la posibilidad de robo, etc. Pero por otro lado, hacer esto nos ahorraría "pasos".