

DISEÑO, DESARROLLO E INSTALACIÓN DE UN SISTEMA DE INFORMACIÓN WEB CON PERSISTENCIA DE DATOS

Sistemas de Información

Curso 2019-2020

Pedro Tamargo Allué - 758267

Juan José Tambo Tambo - 755742

Raúl Rustarazo Carmona - 715657

Diseño de las clases usadas	2
Metodología de trabajo	3
Dificultades encontradas	3
Pruebas realizadas	4
Plan de mantenimiento	4
Bibliografía	5

1. Diseño de las clases usadas

Se han creado las clases Usuario, Consulta, Comentario y vehículo para poder representar la información almacenada en la BD. Todas ellas contienen los atributos y operaciones necesarias para crear y modificar dichos atributos.

Para la interacción con la base de datos, se han creado las siguientes DAO:

ComentarioDAO, *UsuarioDAO* y *VehiculoDAO*, las cuales contienen operaciones para insertar, buscar, modificar y borrar objetos de la clase Comentario, Usuario y Vehículo respectivamente.

Para poder realizar las conexiones con la base de datos, se utiliza la clase *ConnectionManager*, la actúa como pool de conexiones a la BD desde la cuenta de uno de los administradores.

De cara a los diseños de la capa de modelo de la aplicación, se han convertido los ficheros con extensión *.html* a *.jsp*.

Se ha identificado que todas las páginas de la web tienen un elemento común, la cabecera, para ello, se ha utilizado un fichero "*cabecera.jsp*" y una directiva `<jsp:include page="cabecera.jsp">` para incluir ese fichero en la ubicación de la página establecida. Para poder navegar por las diferentes páginas de la web, se han utilizado "*breadcrumbs*". Para indicar en la página que se encuentra el usuario en cada momento, se resalta con un color diferente al resto. Eso se consigue modificando el atributo "*active*" de los mismos en función de cual esté activo, mediante jsp. En la misma, se comprueba con jsp si un usuario ha iniciado sesión. En ese caso, cambia la presentación de los elementos de la cabecera y muestra su nombre, le da la opción de cerrar sesión y un enlace que dirige a una página dónde se puede modificar cualquier información de su perfil. Además, una vez ha iniciado sesión, desaparece la pestaña de registro.

Todas las páginas en las que se tienen que rellenar formularios tienen control de errores mediante jsp, para poder informar al usuario de cualquier error en los datos introducidos.

También, se han implementado servlets aplicados a los formularios de iniciar sesión, cerrar sesión, registro de un nuevo usuario, modificación de usuario e introducir nuevos comentarios en la BD. Todos ellos comprueban si ha habido algún error en la introducción de los datos (en caso de que existan errores, se vuelve a mostrar la misma página al usuario con los errores cometidos) y realizar las operaciones correspondientes.

Por último, junto con los servlets de las acciones de "modificar datos", "registrarse" y "insertar comentario", el servidor envía un correo electrónico (a partir de la cuenta de

Gmail de ecobicizara@gmail.com) con el contenido de la acción que el usuario ha realizado. De esta manera, un usuario que haya cambiado sus datos tendrá en su correo electrónico los datos que han sido modificados. Para ello se ha utilizado la librería “*javax.mail*”.[4]

2. Metodología de trabajo

Para la facilitar la resolución de la práctica, se ha dividido el trabajo entre los integrantes de la siguiente manera:

- **Juan Jose:** Codificación de Servlets, pruebas y jsp. (10 horas)
- **Pedro Tamargo:** Servlets, clases Java, BD y jsp (10 horas)
- **Raúl Rustarazo:** Modificación HTML y jsp. (6 horas)

Para comprobar los cambios que se realizaban sobre el proyecto, se ha utilizado el servidor tomcat con el entorno Eclipse. De esta manera, se simplifica la depuración y las pruebas.

En cuanto a los servlets, se ha utilizado documentación de la teoría de la asignatura y algunos recursos de internet [1].

3. Dificultades encontradas

Al ser la primera vez que se trabajaba con servlets, se han encontrado diversos problemas al principio, como errores de mapeado de los mismos o empaquetado de las clases.

Lo mismo ha ocurrido con jsp, aunque ha resultado de gran utilidad para añadir funcionalidades a la página, para control de errores, simplificar código y añadir dinamismo la cabecera.

También ha resultado complicado el control de las versiones de los ficheros, ya que se han tenido que realizar diversas modificaciones en cuanto a la estructura de los mismos, y se daba el caso de que dos integrantes del grupo modificaban el mismo fichero. Para solucionarlo, se utiliza la herramienta de control de versiones *Git*[3], para poder mantener la consistencia entre los ficheros modificados.

4.Pruebas realizadas

Para poder detectar fallos en la página, se han probado inserciones, modificaciones y borrados de usuario.

A la hora de registrar un usuario, se controla que ningún campo esté vacío, que la contraseña y su repetición coincidan, que el email introducido sea válido (mediante un patrón) y que el usuario con el login introducido no exista en la BD. Todo ello se refleja en el control de errores de la página de registro, que notifica al usuario de cualquier error.

Lo mismo ocurre al iniciar sesión, si el usuario introducido no existe o la contraseña es incorrecta, se notifica el error al usuario en pantalla.

Una vez el usuario ha ingresado en su cuenta, aparece un icono en la parte superior de la cabecera (un engranaje), mediante el cual permite acceder a la página de perfil de usuario, donde aparecen todos sus datos actuales y da la posibilidad de modificarlos. En esta página también se controlan los errores mencionados anteriormente (campos vacíos, contraseñas que no coinciden...).

Existe un apartado de comentarios, desde el cual los usuarios pueden realizar cualquier consulta a los administradores de la página, sin necesidad de estar registrados en la misma. Como se ha mencionado antes, cuando un usuario envía un comentario, se envía automáticamente un correo a la dirección de correo indicada para confirmar la consulta.

Se ha realizado la pregunta:

Juan José Tambo(jtambo@live.com) pregunta:

Hola, este es el cuerpo del mensaje.

Cuerpo del correo enviado al usuario tras la consulta.

5. Plan de mantenimiento

La página web se aloja en un servidor instalado en una Raspberry Pi, haciendo posible así el acceso a la misma en cualquier momento. La base de datos también está alojada en el mismo dispositivo. Agilizando así las conexiones entre ambas.

Para su mantenimiento a lo largo del tiempo, se tendrán actualizados todos componentes del servidor, se realizarán copias de seguridad periódicas, se controlará periódicamente el estado del mismo y se atenderá a todos comentarios enviados a los administradores de la página, para poder atender a cualquier error o sugerencia que pueda aparecer, ya sea relacionado con la cuenta de usuario como con cualquier aspecto relacionado con la página.

Como se ha mencionado anteriormente, la página se aloja en un dispositivo que siempre se encuentra encendido. Para poder actuar lo más rápido posible frente a cualquier imprevisto, se puede monitorizar el estado y comportamiento de la raspberry a distancia, con el programa “*cockpit*” [2].

Para evitar fallos de seguridad, las contraseñas de acceso a la base de datos se renovarán periódicamente.

6. Bibliografía

[1] (10-11-19) <http://www.edu4java.com/es/servlet/servlet2.html>

[2] (7-10-19) <https://github.com/cockpit-project/cockpit>

[3] (1-10-19) <https://git-scm.com/>

[4] (24-11-19) <https://javaee.github.io/javamail/>