

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

Sistemas de Información

Curso 2019-2020

Raúl Rustarazo Carmona - 715657

Pedro Tamargo Allué - 758267

Juan José Tambo Tambo - 755742

Diseño de las clases usadas	2
Metodología de trabajo	2
Dificultades encontradas	3
Pruebas realizadas	3
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:

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

Para poder realizar las conexiones con la base de datos, se utiliza la clase ConnectionManager, la cual da permiso de acceso 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. En la misma, se comprueba con jsp si un usuario ha ingresado en su cuenta. En ese caso, se muestra su nombre y se le da acceso a una página donde puede modificar cualquier información de su perfil.

Todas las páginas con las que interactúa el usuario, como la de registro, la de login, la de modificación de perfil, contienen control de errores mediante jsp, para poder informar al usuario de cualquier error.

Por último, se han implementado una serie de servlets aplicados a los formularios de iniciar sesión, registro, modificación de usuario y cerrar sesión. Todos ellos comprueban si ha habido algún error en la introducción de los datos (en ese caso se vuelve a mostrar la misma página que ahora muestra al usuario los errores) y realizar las operaciones correspondientes.

1. 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 y jsp. (10 horas)
- **Raúl Raustarazo:** Modificación HTML y jsp. (7 horas)

Para poder comprobar los cambios que se iban realizando sobre el proyecto, se ha instalado un servidor apache en eclipse. De esta manera, se simplifica la depuración en gran medida.

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

2. 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 y para simplificar código.

También ha resultado complicada el control de 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 *Git*[3], para poder elegir la versión deseada y actualizada de un archivo.

3. 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 coinciden, 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 logearse, 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 puede acceder a la página de perfil de usuario, donde aparecen todos sus datos actuales, los cuales puede modificar.

En esta página también se controlan los errores mencionados anteriormente (campos vacíos, contraseñas que no coinciden...)

4. Plan de mantenimiento

La página web se aloja en un servidor instalado en una raspberry de un integrante del grupo, haciendo posible así el acceso a la misma en cualquier momento. La base de datos también está alojada en el mismo dispositivo.

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

Sería conveniente realizar copias de seguridad periódicas, para evitar pérdidas de información.

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 deberían renovar periódicamente,

5. Bibliografía

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

[2] (7-10-19) <https://ubunlog.com/cockpit-administra-red-navegador-web/>

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