PRACTICA 3

DISEÑO E IMPLEMETACIÓN DE UN SISTEMA DE INFORMACIÓN WEB CON PERSISTENCIA DE DATOS

Diego Santolaya Martínez
Abel Chils Trabanco
Alexandru Oarga Hategan

ÍNDICE

1 Introducción

En esta práctica se terminan el resto de aspectos de nuestra web de venta de libros que quedaban pendientes en las anteriores. Esto incluye la creación y poblado de la base de datos, la implementación de los modelos de datos en java y de los controladores que manejan estos datos, la fachada de la web *(facade)*, los servlets necesarios y los ficheros *.jsp* que posteriormente generarán el código html de la web.

2 Metodología

En este apartado se detallan el entorno de programación utilizado para el desarrollo de la web, así como los recursos que esta usa. También se explica la repartición del trabajo y el tiempo dedicado por cada miembro.

2.1 Entorno y recursos

La web se ha desarrollado en el entorno InteliJ (idea), lo que permite una gestión más cómoda del servidor web, ya que sólo con seleccionar al inicio del proyecto la versión de tomcat instalada que se desea usar, el programa se encarga de la configuración de éste y se puede comprobar el resultado final sólo con darle a *Play.*

Para la realización de los ficheros se toman como referencia los que se encuentran en la máquina virtual proporcionada en clase. Esto ficheros son la mayoría en java, aunque también hay *.jsp*, en los que convertimos el código html de la práctica anterior en dinámico. También se ha recurrido al entorno Eclipse para probar aisladamente el código *java*.

2.2 Distribución del trabajo

Se ha tratado de distribuir el trabajo de manera equitativa para equilibrar así la carga de trabajo entre los miembros del grupo. La web se ha dividido en página principal (Abel Chils), página de compra de libro (Diego Santolaya) y gestión de usuarios (Alexandru Oarga), encargándose cada integrante del grupo de todos los archivos relacionados con el uso de dicha web.

2.3 Horas de trabajo

El total de horas que aproximadamente se han dedicado a la realización de la práctica es el siguiente:

- Diego Santonalaya:
  -Horas de practicas: 4h
  -Configuración del entorno y descarga de los recursos necesarios: 2h
  -Implementación y pruebas de los ficheros de:
    Modelo de datos: 3h
    Controlador de datos: 3h
    Servlets: 2h
    .jsp correspondiente a la vista de la compra del libro: 3h
    Facade de la tienda: 1h
  -Pruebas adicionales: 1h
  -Redacción de la memoria: 3h
  -Total: 20h

- Abel Chils:
  -Horas de practicas: 4h
  -Configuración del entorno y descarga de los recursos necesarios: 2h
  -Implementación y pruebas de los ficheros de:
    Modelo de datos (incluye sistema de recomendaciones): 5h
    Controlador de datos: 3h
    Servlets: 4h
    .jsp correspondiente a inicio, login, búsqueda y error: 4h
    Facade de la tienda: 1h
  -Pruebas adicionales: 4h
  -Total: 23 horas
- Alexandru Oarga:
  -Horas de practicas: 4h
  -Configuración del entorno y descarga de los recursos necesarios: 2h
  -Implementación y pruebas de los ficheros de:
    Modelo de datos: 2h
    Controlador de datos: 1h
    Servlets: 5h
    .jsp páginas de perfil y configuración: 5h
    Facade de la tienda: 2h
  -Pruebas adicionales: 1h
  - Total: 19 horas

3 Diseño e implementación

Basándonos en el diseño de la web de la práctica anterior se construye el modelo de datos en java, se crea la conexión con la base de datos creada también anteriormente y se desarrolla la aplicación web con InteliJ creando los ficheros correspondientes a la *facade,* a los *servlets* y a las páginas web (ficheros *.jsp*).

3.1 Modelo de datos

Con el fin de poder usar la base de datos MySql creada en la práctica anterior se implementan todos los ficheros VO y DAO necesarios para obtener una interfaz suficiente con la que interactuar desde la web. Para cada una de las entidades del esquema entidad/relación (libro, usuario ,autor ,etc.) y las relaciones con atributos (valoración, comentario, etc.) se crea un fichero VO que contiene como atributos privados todos los que tiene la entidad y las funciones necesarias para interactuar con la clase (constructores, funciones *getParameter* y *setParameter,*etc.).
Los ficheros DAO son 4: autor, libro, género y usuario. Éstos se encargan de crear las sentencias SQL necesarias para interactuar con la base de datos, por ejemplo realizar los INSERTS cuando en la web se haga un nuevo comentario, las consultas que sean necesarias para rellenar los carruseles de libros (según género, más vendidos, etc.) o para el sistema de recomendaciones (libros que más visita el usuario, que más valora, etc.) y muchos otros aspectos relacionados que se listan a continuación:

- Inserciones de libros, autores, géneros, visitas, compras, etc.
- Eliminaciones de datos
- Actualizaciones de libros, autores, usuarios, valoraciones ,comentarios etc.
- Encontrar los datos de un libro

- Encontrar las valoraciones y comentarios de un libro
- Encontrar datos de un autor
- Listar libros de un autor
- Encontrar los géneros de un libro
- Encontrar los datos de un usuario
- Encontrar los comentarios, compras, visitas y valoraciones de un usuario

Estas funciones serán la interfaz con la que nuestra web interactuará para comunicarse con la base de datos. El gestor de la base de datos se encarga de establecer la conexión con esta. Se encarga de la gestión de excepciones para evitar fallos al interactuar con la base.

3.2 FACADE

El fichero *TiendaFacade.java* hace de puente entre la web y la base de datos, implementando todas las funciones necesarias para realizar las transacciones correspondientes con la base. Éstas son: listar libros según su género, según los libros que visita, valora y comenta un usuario, según su categoría en la web, obtener los datos de un libro en concreto, listar las valoraciones totales de un libro, listar todos los comentarios de un libro etc.

De está manera la web es independiente de la implementación de la base, y no será necesaria su modificación si hay algún cambio en la base de datos o en el modelo y controladores.

3.3 SERVLETS Y CONTROLADOR

Como otro punto a favor de la web, se ha de destacar el uso de *servlets* para aumentar la seguridad de nuestra web, evitando que un usuario externo pueda informarse de las direcciones de los ficheros de la página. Estos *servlets* hacen que cuando se accede a una url concreta, por ejemplo al pulsar un libro, al poner la secuencia *"/libro/"* en la *url* el *servlet* te redirecciona a la página *.jsp* correspondiente a la página de compra del libro en cuestión. Se encargan también del inicio y cierre de sesión de los usuarios comprobando que el usuario y contraseña son correctos.

Además, en el controlador se incluye un fichero con las constantes comunes para tener mayor consistencia entre módulos. También en caso de error un *servlet* nos redireccionará a una página de error, evitando así posibles inconsistencias si ha habido alguna excepción o algún caso inesperado.

3.4 PÁGINAS

Para hacer las páginas dinámicas se han sustituido los ficheros *.html* por los *.jsp*. Estos últimos generan el código *html* en base a unas variables que se definen en apartados de código java. Desde estos se accede a las funciones del *facade* y se obtienen los datos necesarios para rellenar los títulos, autores, descripciones, comentarios, caloraciones, etc.

La página principal se compone de varios carruseles que muestran distintos libros agrupados por categorías (más vendidos, populares y novedades), que irán variando según vayamos accediendo a las distintas secciones de la web. Las categorías a las que puedes acceder en cada momento son dinámicas, es decir, varían según en la página en la que estés. Esto se hace para evitar repeticiones de categorías y proporcionar unas opciones más ajustadas a lo que está buscando.

Al acceder a registrarte la web mostrará un error si la contraseña no coincide con el usuario o si no existe ningún usuario con el nombre seleccionado, reenviándote a la página de error. Si en cambio pulsas cualquiera de los libros sugeridos en los carruseles, serás redireccionado a la página de compra de libro. Está página recibe por *get* (como casi todos los pasos de información en nuestra web) el isbn del libro a mostrar, invoca a la función de encontrar un libro concreto, rellenando las variables de autor, título, etc. con los datos de dicho libro. También consulta las valoraciones del

libro, calcula la media y muestra las estrellas que ha obtenido. Por último consulta la lista de comentarios del libro y los lista junto al usuario que lo envío y la fecha.

Para saber si un usuario ha iniciado sesión en nuestra web se usa el objeto sesión, de esta manera se le permitirá comentar y valorar los libros que haya comprado previamente, o acceder a modificar sus datos en cualquier momento.

La idea de la compra es que el usuario ya haya introducido previamente los datos de su tarjeta para posteriormente cuando toque comprar se le pueda cargar a su cuenta el importe del libro.

Somos conscientes de la vulnerabilidad que supone el paso de información por *get* frente a un posterior cambio las variables en la url, pero en esta práctica hemos decidido focalizar en otros aspectos.

4 Sistema de recomendaciones

El sistema de recomendaciones de nuestra web se basa en un sistema de puntos de afinidad con otros usuarios. Es decir, si un usuario ha comprado los mismos libros que otro, estos dos tendrán más puntos de afinidad entre ellos que otros que no tengan libros en común. A cada libro, se le asigna la suma de los puntos de afinidad que tiene cada usuario que haya comprado ese libro. Se listan los primeros libros que más puntos de afinidad hayan sumado (es decir, los libros que más hayan comprado los lectores más afines a ti son los que te mostrará el carrusel de recomendaciones).

5 Pruebas

Con la ayuda del entorno de desarrollo InteliJ, las pruebas simplemente se han traducido en lanzar el servidor y comprobar manualmente el correcto funcionamiento de la web, haciendo los *“prints”* necesarios para saber que partes del código se ejecutaban y cuáles no, hasta conseguir el correcto funcionamiento del servidor web.

También se ha probado aisladamente el código java en un entorno de desarrollo específico para comprobar su corrección.

6 Mantenimiento en entorno de producción

La idea es que nuestra web no necesitase de mantenimiento para funcionar, sino que mediante la gestión que realiza con la base de datos le es suficiente para que los clientes puedan disfrutar del servicio cuando quieran. Sólo sería necesaria la intervención de una tercera persona cuando sea necesaria la adición, modificación o eliminación de algún libro o autor, ya que como es lógico la base de datos no se actualiza sola automáticamente.