

Informe de pràctiques

Entrega 2

Alumne: Albert Acedo

Alumne: Alexandre Pérez

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Contingut

Introducció	1
Decisions de disseny.....	2
Repositoris de codi consultats	4
Bibliografia consultada.....	4

Introducció

En aquesta pràctica ens hem centrat en la realització del gestor de la base de dades, que ens servirà per a futurs projectes, ja que es un procés molt centralitzat i escalable. Hem implementat completament l'enunciat de la pràctica 2, a més d'afegir certes modificacions esmentades a continuació.

A part dels fitxer citats en l'enunciat, hem afegit els següents amb les seves funcionalitats:

- index.jsp: arrel de l'arbre que ens redirigeix a l'inici de sessió o al menú (per fer proves).
- error.jsp: recull els possibles errors i els mostra a l'usuari per pantalla (problemes en modificacions en la base de dades o problemes d'identificació).
- success.jsp: recull quan les tasques es realitzen correctament i les mostra a l'usuari per pantalla (modificacions en la base de dades correcte).
- register.jsp: formulari per registrar-se com a usuari de l'aplicació.
- Utils.java: classe auxiliar que s'utilitza com a gestor de les UUID's de les imatges i la neteja de l'ordre de les paraules clau per a realitzar cerques d'imatges.
- list.java: obté les dades que s'han de mostrar a list.jsp i les emmagatzema com una variable request List<Map> on la llista conté les files de la taula *images*.
- Database.java: gestor centralitzat de la base de dades, que conté totes les funcions que els fitxers JSP cridaran.
- register.java: implementa el registre de register.jsp.
- style.css: complementa la visualització dels fitxers JSP.

Decisions de disseny

En primer lloc, hem decidit optar per la utilització de les llibreries *JavaServer Pages Standard Tag Library* (JSTL) en els fitxers JSP de la aplicació. Aquestes llibreries funcionen com a extensió de les anteriors i ens ofereixen un estàndard més universal i una especificació més detallada. De cara a futurs projectes ens poden servir com a eina per millorar la escalabilitat al afegir més funcionalitats.

També hem afegit previsualitzacions i missatges per a l'usuari quan es modifiquen les imatges en la base de dades a través del fitxer `success.jsp`. Així, l'usuari pot visualitzar si les seves modificacions s'han realitzat correctament o no.

El fitxer `error.jsp`, a part de mostrar els errors referents a les modificacions en la base de dades, també ho fa quan l'usuari intenta accedir sense identificar-se o s'identifica incorrectament.

A més, la nostre aplicació compta amb una gestió centralitzada de la base de dades (en `Database.java`), que ens permet una gestió molt més fàcil per part dels Servlets i simplifica les crides que aquests fan al gestor. Es minimitza el codi ja que els diferents tipus de consultes només es programen una vegada i no per a cada Servlet. La metodologia és la següent:

- Quan volem realitzar una consulta o modificació a la BD el Servlet corresponent realitzarà una crida a la funció pública (nom_funció) descrita pel gestor de base de dades de JAVA.
- Per a cada funció pública hi ha una funció protegida associada (sql_nom_funció). Aquesta funció es crida a través de la funció *execute* que s'encarrega de preparar la connexió i el tractament d'errors de la BD.

Com a extensió en la compatibilitat, hem utilitzat el fitxer `Utils.java` per a implementar la utilització de les paraules clau i la reassignació del nom del fitxer al disc.

Per altre banda, el Servlet `images.java` gestiona la lectura de les imatges del disc mitjançant un atribut en la URL que indica el fitxer que volem consultar. A més, podem mostrar imatges, no només JPEG, sinó de qualsevol extensió de fitxers multimèdia (JPEG, JPG, PNG, GIF...) de forma dinàmica.

En el cas que volguéssim, podríem limitar aquesta ampliació (i limitar-nos a fer un gestor d'imatges JPEG com així ens especifica l'enunciat) o inclús fer una comprovació dels fitxers que l'usuari els puja al servidor i només realitzar la pujada si el fitxer compleix una sèrie de requisits d'extensió (només X extensions de fitxers determinades, només imatges no mòbils...).

En aquest aspecte, també hem decidit aleatoritzar els noms que reben els fitxer quan són emmagatzemats en memòria. Utilitzant els UUID's ens assegurem (en una probabilitat molt elevada) que els fitxers que es guardaran tindran noms únics. Així, evitem duplicar noms en memòria i possibles conflictes de domini.

També hagués sigut vàlida una concatenació d'una sèrie de valors variables (Ex: nom de l'usuari + data + extensió de fitxer), però d'aquesta forma no ens podem assegurar que en un futur, en un sistema més gran, aquests valors siguin únics. Si a més es poguessin descarregar les fotos, estaríem mostrant informació que podria ser utilitzada per a atacar al sistema (noms d'usuaris, data de pujada...).

Per últim, a l'hora de realitzar les cerques d'imatges, hem optat per un buscador tipu Google, en el que podem anar fent cerques simultànies i sempre amb la opció de tornar al menú. Si deixem els camps en blanc, per defecte, busquem entre totes les fotos publicades. Sinó també podem buscar per paraules clau o títol, mostrant sempre el nombre de resultats que obtenim.

També podem modificar qualsevol atribut en la pàgina de modificació d'imatges.

Repositoris de codi consultats

-Pas de funcions com a paràmetres en funcions de JAVA:

<https://codereview.stackexchange.com/questions/186972/passing-a-generic-function-as-parameter>

-Gestió dels inicis de sessió d'usuari mitjançant JSP:

<https://stackoverflow.com/questions/14128950/jsp-how-to-check-if-a-user-is-logged-in>

-Obtenció de l'extensió d'un fitxer en JAVA:

<https://stackoverflow.com/questions/3571223/how-do-i-get-the-file-extension-of-a-file-in-java>

-Gestió de fitxers d'imatge en Servlets/JSP i com guardar-los en memòria:

<https://stackoverflow.com/questions/2422468/how-to-upload-files-to-server-using-jsp-servlet/2424824#2424824>

<https://stackoverflow.com/questions/31129327/how-do-i-display-a-html-file-with-an-image-from-a-servlet>

<https://www.roseindia.net/answers/viewqa/JSP-Servlet/4313-How-display-a-Image-on-servlet-from-file-upload.html>

Bibliografia consultada

-Consulta sobre la creació de sessions en JAVA i la gestió dels atributs associats:

<https://docs.oracle.com/cd/E19857-01/819-6518/gcxvp/index.html>

<https://www.careerride.com/Servlet-getSession-methods.aspx>