

# Informe Pràctica 3

Sergi Sendrós Plana  
Marc Casellas Muns  
Aplicacions Distribuïdes  
25/10/2018  
Tardor 2018

<b>1. WEB Service</b>	<b>3</b>
1.1 RegisterImage	3
1.3 ListImages	3
1.5 SeachByTitle	4
1.6 SearchByAuthor	4
1.7 SearchByCreaDate	4
1.8 SearchByKeywords	4
<b>2. Java application</b>	<b>4</b>
2.1 Introducció	4
2.2 Estructura	4
<b>3. Web application</b>	<b>5</b>
3.1. Llistat de .jsp	5
3.1.1 Menu.jsp	5
3.1.2 RegistrarImagen.jsp	5
3.1.3 UploadOK.jsp	6
3.1.4 List.jsp	6
3.1.5 BuscarImagen.jsp	6
3.1.6 ModificarImagen.jsp	7
3.2. Llistat de servlets	7
3.2.1 RegistrarImagen	7
3.2.4 BuscarImagen	7
3.2.5 ModificarImagen	7
3.3. Carpetes i Bases de dades	7
3.3.1 Images	7
3.3.2 FotOK	7

# 1. WEB Service

## 1.1 RegisterImage

Funció que rep una ImageWS com a paràmetre d'entrada, a aquesta imatge, li falten alguns paràmetres com el timestamp, que es crea dins la funció a partir de la data actual en que s'està pujant la imatge. Per afegir el camp *id*, es farà una cerca a la base de dades i s'incrementarà en 1 el màxim trobat. Un cop tots els camps estan complets, s'afegeix la imatge a la base de dades. Si el procés es duu a terme sense cap problema, es retornarà un 1. En cas d'error la funció retorna un 0.

## 1.2 ModifyImage

Funció que rep una ImageWS com a paràmetre d'entrada. Aquesta imatge parteix d'una imatge anterior amb cap, un, o més camps modificats. Es fa un update de la nova imatge amb el *id* que té assignat, de manera que es modifica la imatge original. Si el procés es duu a terme sense cap problema, es retornarà un 1. En cas d'error la funció retorna un 0.

## 1.3 ListImages

Funció sense paràmetres d'entrada. Retorna una llista de ImageWS mitjançant una crida Select de tots els camps de la base de dades en format ImageWS.

## 1.4 SearchById

Funció que rep un integer com a paràmetre d'entrada corresponent al camp *id*. A continuació la funció executa una cerca SQL que encapsula el resultat en una ImageWS corresponent a la imatge amb el id del paràmetre d'entrada. Si el procés

es duu a terme sense cap problema, es retornarà la imageWS. En cas d'error la funció retorna un null.

### **1.5 SeachByTitle**

Funció que rep un String *títol* com a paràmetre d'entrada. A continuació, s'executa una sentència SQL que recull totes les imatges que tinguin *títol* en aquest mateix camp. El resultat és retorna en una Llista de ImageWS, en cas d'error es retorna un null.

### **1.6 SearchByAuthor**

Funció que rep un String *autor* com a paràmetre d'entrada. A continuació, s'executa una sentència SQL que recull totes les imatges que tinguin *autor* en aquest mateix camp. El resultat és retorna en una Llista de ImageWS, en cas d'error es retorna un null.

### **1.7 SearchByCreaDate**

Funció que rep un String *datac* com a paràmetre d'entrada. A continuació, s'executa una sentència SQL que recull totes les imatges que tinguin *datac* en aquest mateix camp. El resultat és retorna en una Llista de ImageWS, en cas d'error es retorna un null.

### **1.8 SearchByKeywords**

Funció que rep un String *tags* com a paràmetre d'entrada. A continuació, s'executa una sentència SQL que recull totes les imatges que tinguin *tags* en aquest mateix camp. El resultat és retorna en una Llista de ImageWS, en cas d'error es retorna un null.

## **2. Java application**

### **2.1 Introducció**

Aplicació java que està formada per un main() que permet l'execució d'una sèrie de passos per proporcionar un servei. Aquest servei implementa els mètodes del WEB Service, i permet executar les funcions d'aquest. L'aplicació permet registrar imatges, modificar imatges, llistar imatges i fer cerques per diferents camps.

### **2.2 Estructura**

Al iniciar l'aplicació es demana un nom d'usuari (serveix per publicar imatges). A continuació es poden triar 4 opcions:

- Registrar Imatge
- Modificar Imatge
- Llistar Imatges
- Buscar Imatges

Si triem Registrar Imatge, ens demana els camps necessaris per crear la imatge i es crida a la funció RegisterImage.

Si triem Modificar Imatge, ens demana el id de la imatge a modificar, i posteriorment ens permet modificar els camps.

## 3. Web application

Si triem Llistar Imatges, ens llista totes les imatges de la base de dades, a més, ens permetrà modificar la imatge que vulguem.

Si triem Buscar Imatges, ens demana per quin paràmetre volem buscar la imatge: per títol, autor, data de creació, tags, etc. I ens retorna les imatges que coincideixin amb la cerca.

### 3.1. Llistat de .jsp

S'han aprofitat algunes de les pàgines de la pràctica anterior. S'han mantingut aquelles necessàries per implementar el servei SOAP.

#### 3.1.1 Menu.jsp

Pàgina on hi trobem el menú principal de l'aplicació. Des del menú, es poden accedir a diferents serveis:

- RegistrarImagen.jsp
- List.jsp
- BuscarImagen.jsp

#### 3.1.2 RegistrarImagen.jsp

Pàgina on hi trobem un formulari per pujar una imatge a la webapp. Aquest formulari consta de:

- Títol de la imatge
- Descripció de la imatge
- Tags de la imatge
- Autor de la imatge
- Data de creació de la imatge

Un cop omplert el formulari aquest s'envia via POST al servlet RegistrarImagen que s'encarrega de processar la petició.

### 3.1.3 UploadOK.jsp

Pàgina a la que s'accedeix un cop pujada una imatge exitosament. Compta amb dos botons, un que ens redirigirà al Menu.jsp i una altre que ens permetrà pujar una nova imatge redirigint-nos a RegistrarImagen.jsp

### 3.1.4 List.jsp

Pàgina on hi apareixen llistades totes les imatges de la WebApp. Això s'aconsegueix mitjançant una crida a la funció ListImages que es connecta amb el servei SOAP que conté totes les funcions per aconseguir els atributs de la imatge. Per cada imatge es mostren:

- La pròpia imatge
- Títol
- Descripció
- Tags
- Autor
- Data creació

Apareix un botó extra que ens permet modificar la imatge redirigint-nos a ModificarImagen.jsp.

### 3.1.5 BuscarImagen.jsp

Pàgina de cerca d'imatges mitjançant una entrada de text. La pàgina té tres estats diferents: El primer estat és la pàgina de cerca inicial, on hi apareix una barra de cerca on hi podem afegir la paraula clau a buscar. Un cop fet el submit aquest s'envia al servlet BuscarImagen i la pàgina s'actualitza segons si s'ha obtingut resultats o no.

Si s'ha obtingut resultats, es mostraran totes les imatges que tinguin la paraula clau en un camp de la seva informació ja sigui autor, usuari, tag, títol, etc. A més, si el username associat a la imatge és el mateix que l'usuari connectat, apareix un botó extra que ens permet modificar la imatge redirigint-nos a ModificarImagen.jsp.

Si no ha obtingut cap resultat, es mostrarà un missatge en pantalla indicant-ho.

Com abans, a la pàgina només s'hi pot accedir amb la sessió iniciada, ja que abans d'entrar-hi es comprovarà que existeixi una sessió correcta. Si no s'ha iniciat una sessió, es redirigirà al login.jsp.

### **3.1.6 ModificarImagen.jsp**

Pàgina a la que s'accedeix des de BuscarImagen.jsp o List.jsp. En aquesta pàgina ens apareix un formulari com el de RegistrarImagen.jsp amb la diferència que els camps apareixen amb la informació vigent de la imatge, el que permet a l'usuari de la WebApp canviar-los al seu gust, sabent la informació que hi havia anteriorment. El formulari compta amb un submit que enviara tota la informació al servlet ModificarImagen que s'encarregarà de connectar-se amb el WEB Service per atendre la petició.

## **3.2. Llistat de servlets**

### **3.2.1 RegistrarImagen**

Servlet que rep un formulari de registrarImagen.jsp. Aquest formulari s'utilitza com a argument de RegisterImage. Un cop executada la funció, es redirecciona a UploadOK.jsp.

### **3.2.4 BuscarImagen**

Servlet que rep un text i un camp de la base de dades. Segons el camp que sigui (autor, titol, datac...) executarà una funció o altre i retornarà una llista amb les coincidències.

### **3.2.5 ModificarImagen**

Servlet que rep una ID d'una imatge i que crea un formulari amb els valors per defecte corresponents a aquella ID. Permet que l'usuari modifiqui els camps o els deixi intactes, finalment es crida a la funció ModifyImage que actualitzarà la base de dades.

## **3.3. Carpetes i Bases de dades**

### **3.3.1 Images**

Carpeta on s'emmagatzemen totes les imatges que puja l'APP reanomenades amb el timestamp de pujada.

### **3.3.2 FotOK**

Base de dades d'usuaris (usuari i contrasenya) i fotos ({id, titol, autor, descripcio, tags, datac, timestamp, filename, username}).

La base de dades és derby, ja que ens permet una millor gestió per treballar en grup.