

# **INFORME PRÁCTICA 3 - Desarrollo de servicios web con SOAP**

Carles Salvador Bertran, Rubén Sanz García

Aplicacions Distribuïdes  
21/10/2019

# Índice

<b>1. Introducción</b>	p. 2
<b>2. Desarrollo de la práctica</b>	
<b>2.1 Crear clase Image y GestorImagenes.java</b>	p. 2
<b>2.2 Creación <i>JavaAppClient</i></b>	p. 3
<b>2.4 Creació <i>WebAppClient</i></b>	p. 4

## 1. Introducció

En esta práctica partiremos del proyecto creado en la práctica 2 para desarrollar un servicio web basado en SOAP. Antes de comenzar, se ha seguido un tutorial proporcionado por Netbeans el cual nos presenta cómo crear un simple servicio web que funciona de calculadora. Una vez completado y teniendo un esqueleto sobre el cual adaptar nuestro proyecto, procedemos a implementarlo.

## 2. Desarrollo de la práctica

### 2.1 Crear clase Image y GestorImágenes.java

La nueva clase *Image* contiene una serie de parámetros con información sobre las imágenes y será la clase principal con la que tratará nuestro servicio web. Se ha decidido implementar estos atributos como elementos private a los que se accederán con funciones get y set. En la siguiente imagen podemos ver los campos que contiene la clase *Image*.

```
public class Image {  
    private int id;  
    private String title;  
    private String author;  
    private String description;  
    private String keywords;  
    private String creaDate;  
    private String altaDate;  
    private String filename;  
}
```

En cuanto a las funciones que se implementarán en el archivo GestorImágenes.java, son aquellas indicadas en el enunciado.

Copiar código de las funciones .jsp y .java de la práctica anterior y adaptarlos a esta. Las operaciones con la BD pasarán a ser ejecutadas desde la clase GestorImágenes.java a través del cliente WebApp y el cliente JavaApp.

Los proyectos de estos clientes se encuentran dentro de la carpeta ADpractica3. Los archivos *Image.java* y *GestorImágenes.java* han sido añadidos a un nuevo paquete en source llamado *gestor1*, que será el que utilizarán los clientes para ejecutar sus llamadas.

## 2.2 Creación *JavaAppClient*

Este cliente lo que nos permitirá será testear las funciones que hemos creado a través de un simple menú de texto en la consola.

Siguiendo los pasos del tutorial, para que este cliente acceda a la funciones de nuestro servidor tenemos que añadirle a este nuevo proyecto como *WebServiceClient*. Una vez hecho eso aparecerá una carpeta de Web Service References en la que se lista las funciones que puede ejecutar nuestro servidor y para añadirlas a nuestro código podemos arrastrarlas a nuestro *JavaAppClient.java* para luego llamarlas desde el *main*.

```
compile:
run:
Escull una opció:
1 - Registrar una imatge:
2 - Modificar una imatge
3 - Llistar totes les imatges
4 - Buscar una imatge
5 - Sortir del programa
1

Introdueix titol:
hola
Introdueix autor:
carles
Introdueix descripció:
fototest
Introdueix keywords:
test,prueba
Introdueix data de creació:
24/09/2019
Introdueix filename:
hola.jpg
Imatge registrada amb exit!

case 1: // Registrar imatge
    System.out.println(" ");
    Image tmp = new Image();
    input.nextLine();
    System.out.println("Introdueix titol:");
    tmp.setTitle(input.nextLine());
    System.out.println("Introdueix autor:");
    tmp.setAuthor(input.nextLine());
    System.out.println("Introdueix descripció:");
    tmp.setDescription(input.nextLine());
    System.out.println("Introdueix keywords:");
    tmp.setKeywords(input.nextLine());
    System.out.println("Introdueix data de creació:");
    tmp.setCreaDate(input.nextLine());
    System.out.println("Introdueix filename:");
    tmp.setFilename(input.nextLine());
    int ret = registerImage(tmp);
    if (ret == 0) System.out.println("Operació fallida");
    else System.out.println("Imatge registrada amb exit!");
    break;
```

En esta imagen podemos ver a la izquierda el output que se produce al ejecutar el cliente java y registrar una imagen, en la derecha podemos ver el código correspondiente a registrar imagen y como al final hace una llamada a la función *registerImage* que es la se conecta con el servidor.

## 2.3 Creació *WebAppClient*

En este caso, queremos acceder a las funciones de nuestro servidor a través del navegador. En el navegador, podremos observar una página que nos muestra varios formularios con los que podemos ejecutar las diferentes funciones pedidas en el enunciado, estos formularios sirven para hacer llamadas al servidor y que, seguidamente, este acceda a la base de datos para modificarla.

En la imagen a continuación podemos ver la apariencia del menú mencionado anteriormente donde aparecen los diversos formularios para las funciones implementadas.

The image shows a web application interface titled "Web app Client". It contains four distinct forms arranged in a 2x2 grid:

- Llistar totes les imatges:** A simple form with a single blue button labeled "Llistar Imatges".
- Buscar una imatge:** A form with a dropdown menu labeled "Selecciona camp a buscar" (currently showing "ID"), a text input field labeled "Introdueix cerca:", and a blue button labeled "Cerca".
- Modifica una imatge:** A form with a text input field labeled "Introdueix ID:", a dropdown menu labeled "Selecciona camp a modificar" (currently showing "Autor"), a text input field labeled "Introdueix nou valor:", and a blue button labeled "Modifica".
- Registra una imatge:** A form with five text input fields labeled "Introdueix titol:", "Introdueix autor:", "Introdueix descripcio:", "Introdueix keywords:", and "Introdueix data de creació:", followed by a text input field labeled "Introdueix filename:" and a blue button labeled "Registra".

Estos formularios se comunican cada uno con su servlet correspondiente el cual hace la llamada a la función que conecta con el servidor una vez.

La forma de añadir esas funciones es la misma que con la aplicación Java, se ha de añadir para que pueda hacer referencia a nuestro servidor y una vez tiene constancia de las funciones que puede ejecutar estas pueden ser añadidas al código arrastrando desde la vista de proyectos.

En esta aplicación no se ha añadido control de sesiones y usuarios, pero se podrían haber implementado añadiendo las páginas de login y registro hechas para la práctica dos y modificando la estructura de la página web.