INFORME PRÁCTICA 3 - Desarrollo de servicios web con SOAP

Carles Salvador Bertran, Rubén Sanz García

Aplicacions Distribuïdes 21/10/2019

Índice

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

1. Introducción

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 GestorImagenes.java

La nueva classe *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 GestorImagenes.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 GestorImagenes.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 *GestorImagenes.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*.

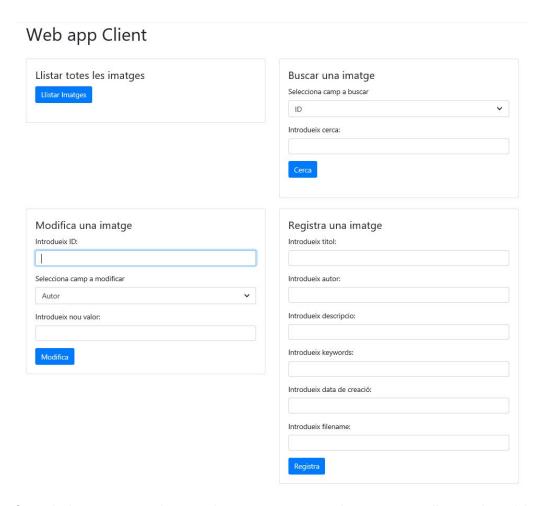
```
compile:
                               case 1: // Registrar imatge
Escull una opció:
                                 System.out.println(" ");
1 - Registrar una imatge:
                                  Image tmp = new Image();
2 - Modificar una imatge
                                  input.nextLine();
3 - Llistar totes les imatges
                                   System.out.println("Introducix titol:");
4 - Buscar una imatge
                                   tmp.setTitle(input.nextLine());
5 - Sortir del programa
                                   System.out.println("Introdueix autor:");
                                   tmp.setAuthor(input.nextLine());
                                   System.out.println("Introdueix descripció:");
Introdueix titol:
                                    tmp.setDescription(input.nextLine());
                                   System.out.println("Introdueix keywords:");
Introdueix autor:
                                   tmp.setKeywords(input.nextLine());
                                   System.out.println("Introdueix data de creació:");
                                   tmp.setCreaDate(input.nextLine());
Introdueix descripció:
                                   System.out.println("Introdueix filename:");
fototest
                                   tmp.setFilename(input.nextLine());
Introdueix keywords:
                                   int ret = registerImage(tmp);
test.prueba
                                  if (ret == 0) System.out.println("Operació fallida");
Introdueix data de creació:
                                   else System.out.println("Imatge registrada amb exit!");
24/09/2019
                                  break;
Introdueix filename:
hola.ipg
Imatge registrada amb exit!
```

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.



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.