Rubén Sanz y Carles Salvador

Informe final prácticas 2 a 5

Contesta a las siguientes cuestiones referentes a las prácticas.

Práctica 2

1. Copia en el cuadro el código del servlet que recoge los datos del formulario para registrar una imagen, guardarlos en la base de datos y almacenar el fichero con la imagen en disco.

```
String title, description, keywords, author, fileName, creation_date,
storage_date;

                title = request.getParameter("title");

                description = request.getParameter("description");

                keywords = request.getParameter("keyword");

                author = (String) sesion.getAttribute("usuario");

                creation_date =
(String)request.getParameter("creationDate");



                //cogemos decha actual

                DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");

                LocalDateTime dateTime = LocalDateTime.now();

                storage_date = dateTime.format(formatter);



                //recogemos archivo subido, guardamos su nombre en
fileName

                Part filePart = request.getPart("picture"); //
Retrieves <input type="file" name="file">

                fileName =
Paths.get(filePart.getSubmittedFileName()).getFileName().toString();
// MSIE fix.

                InputStream fileContent = filePart.getInputStream();

                //guardamos el fichero subido en nuestro directorio

                File uploads = new
File("C:\\Users\\fenix\\Desktop\\AD\\ADpractica2\\ADpractica2\\web\\Im
agenes");

                File fichero = new File(uploads, fileName);
```

```
                try (InputStream input = fileContent) {
                    Files.copy(input, fichero.toPath());
                }

                query = "select max(id) from image";
                statement = connection.prepareStatement(query);
                ResultSet rs = statement.executeQuery();
                int newId = 1;
                if(rs.next())
                    newId= rs.getInt(1) + 1;


                query = "insert into image (id, title, description,
keywords, author, creation_date, storage_date, filename)"
                        + "values(?, ?, ?, ?, ?, ?, ?, ?)";
                statement = connection.prepareStatement(query);
                statement.setInt(1, newId); //id imagen
                statement.setString(2, title); //titulo imagen
                statement.setString(3, description); //descripcion
imagen
                statement.setString(4, keywords); //palabras clave
                statement.setString(5, author); //autor
                statement.setString(6, creation_date); //fecha
creacion
                statement.setString(7, storage_date); //fecha alta
(actual)
                statement.setString(8, fileName);  //nombre fichero
                statement.executeUpdate();
```

2. Copia en el cuadro el código del formulario html que pide al usuario los datos de una imagen para registrarla. Según como lo hayáis implementado, puede ser código html o una página jsp.

```
<form action="registrarImagen" method="POST" enctype="multipart/form-
data">
                Título:
                <input type="text" name="title" /> <br><br>
                Descripción:
                <input type="text" name="description" /> <br><br>
                Palabras clave (separadas por comas y sin espacios):
                <input type="text" name="keyword" /> <br><br>
                Fecha de creación:
                <input type="date" name="creationDate" /> <br><br>
                Imagen:
                <input type="file" name="picture" /> <br><br>

                <input type="submit" value="Enviar" /> <br><br>
            </form>
```

Práctica 3

1. Copia en el cuadro la operación de registro de una imagen en SOAP.

```
/**
     * Web service operation
     * @param image
     * @return
     */
    @WebMethod(operationName = "RegisterImage")
    public int RegisterImage(@WebParam(name = "image") Image image) {
```

```
        DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("yyyy-MM-dd");

        LocalDateTime dateTime = LocalDateTime.now();

        String storage_date = dateTime.format(formatter);


        Connection connection = null;

        try {

            PreparedStatement statement;

            String query;

            Class.forName("org.sqlite.JDBC");

            connection =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\ruben.sanz.garcia\
\Desktop\\tr\\PracticasAD\\practica3\\ADpractica3-
projecte\\practica3.db");



            query = "select max(id) from image";

            statement = connection.prepareStatement(query);

            ResultSet rs = statement.executeQuery();

            int newId = 1;

            if(rs.next()) newId= rs.getInt(1) + 1;




            query = "insert into image (id, title, description,
keywords, author, creation_date, storage_date, filename)"

                        + "values(?, ?, ?, ?, ?, ?, ?, ?)";

            statement = connection.prepareStatement(query);

            statement.setInt(1, newId); //id imagen

            statement.setString(2, image.getTitle()); //titulo imagen

            statement.setString(3, image.getDescription());
//descripcion imagen

            statement.setString(4, image.getKeywords()); //palabras
clave

            statement.setString(5, image.getAuthor()); //autor
```

```
            statement.setString(6, image.getCreaDate()); //fecha
creacion
            statement.setString(7, storage_date); //fecha alta
(actual)
            statement.setString(8, image.getFilename());  //nombre
fichero
            statement.executeUpdate();



        }catch (SQLException | ClassNotFoundException e) {
                System.err.println(e.getMessage());
                out.println(e.getMessage());
                return 0;
            }finally {
                try {
                    if (connection != null) {
                        connection.close();
                    }
                } catch (SQLException e) {
                    // connection close failed.
                    System.err.println(e.getMessage());
                    return 0;
                }
            }
        return 1;
    }
```

2. Copia en el cuadro la operación de búsqueda por título en SOAP.

```
/**
     * Web service operation
     * @param title
     * @return
     */
    @WebMethod(operationName = "SearchbyTitle")
    public List SearchbyTitle(@WebParam(name = "title") String title)
{
        List<Image> retList = new ArrayList<Image>();

        Connection connection = null;
        try{
            PreparedStatement statement;
            String query;
            Class.forName("org.sqlite.JDBC");
            connection =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\ruben.sanz.garcia\
\Desktop\\tr\\PracticasAD\\practica3\\ADpractica3-
projecte\\practica3.db");

            //cogemos todas las imagenes
            query = "select * from image where title like ?";
            title = '%' + title + '%';
            statement = connection.prepareStatement(query);
            statement.setString(1, title);
            ResultSet rs = statement.executeQuery();
            while(rs.next()) {
                Image tmp = new Image();
                tmp.setId(rs.getInt("id"));
                tmp.setTitle(rs.getString("title"));
                tmp.setAuthor(rs.getString("author"));
```

```
                tmp.setDescription(rs.getString("description"));
                tmp.setKeywords(rs.getString("keywords"));
                tmp.setCreaDate(rs.getString("creation_date"));
                tmp.setAltaDate(rs.getString("storage_date"));
                tmp.setFilename(rs.getString("filename"));

                retList.add(tmp);
            }

        }catch (SQLException e) {
            System.err.println(e.getMessage());
            out.println(e.getMessage());
            return null;
        } catch (ClassNotFoundException ex) {

Logger.getLogger(GestorImagenes.class.getName()).log(Level.SEVERE,
null, ex);
        }finally {
            try {
                if (connection != null) {
                    connection.close();
                }
            } catch (SQLException e) {
                System.err.println(e.getMessage());
                return null;
            }
        }
        return retList;
    }
```

3. Copia en el cuadro el código que llama a una de las operaciones del servicio web de imágenes en SOAP.

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response) {

…

List<Object> retList = new ArrayList<Object>();

retList = listImages();

…

}

private List<Object> listImages() {

        // Note that the injected javax.xml.ws.Service reference as
well as port objects are not thread safe.

        // If the calling of port operations may lead to race
condition some synchronization is required.

        gestor1.GestorImagenes port = service.getGestorImagenesPort();

        return port.listImages();

    }
```

Práctica 4

1. Copia en el cuadro la operación para modificar una imagen ya existente en REST.

```
/**

    * POST method to register a new image

     * @param id

    * @param title

    * @param description

    * @param keywords

    * @param author

    * @param crea_date

     * @param fileName
```

```
    * @return
    */
    @Path("modify")
    @POST
    @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
    @Produces(MediaType.TEXT_HTML)
    public String modifyImage (@FormParam("id") int id,
@FormParam("title") String title, @FormParam("description") String
description,
            @FormParam("keywords") String keywords,
@FormParam("author") String author, @FormParam("creation") String
crea_date, @FormParam("fileName") String fileName){


        html = cabeceras("Modificado");


        Connection connection = null;
            try {
                PreparedStatement statement;
                String query;
                Class.forName("org.sqlite.JDBC");
                connection =
DriverManager.getConnection("jdbc:sqlite:/Users/carles/Desktop/Practic
asAD/practica4/ADpractica4-proyecto/practica4.db");


                query = "update image set title=?, description=? ,
keywords=?, filename=? ,author=? where id=?";
                statement = connection.prepareStatement(query);
                statement.setString(1, title); //titulo imagen
                statement.setString(2, description); //descripcion
imagen
                statement.setString(3, keywords); //palabras clave
                statement.setString(4, fileName);
                statement.setString(5, author);
                statement.setInt(6, id);
```

```
                statement.executeUpdate();


                //todo chido pos te presenta mensaje de exito
                html += "<h2> Modificación ejecutada con exito </h2>"
                        + "<b>Información actual:</b><br>"
                        + "<br>   &nbspTítulo:
"+title+"<br>"
                        + "   &nbspDescripción:
"+description+"<br>"
                        + "   &nbspPalabras clave:
"+keywords+"<br>"
                        + "   &nbspNombre del fichero:
"+fileName+"<br>"
                        + "   &nbspAutor:
"+author+"<br><br>"
                        + "<a
href=\"http://localhost:8080/practica4/menu.jsp\">"
                        + "<small> Volver al menú </small></a>";
            } catch (SQLException | ClassNotFoundException e) {
                System.err.println(e.getMessage());
            }finally {
                try {
                    if (connection != null) {
                        connection.close();
                    }
                } catch (SQLException e) {
                    // connection close failed.
                    System.err.println(e.getMessage());
                }
           }
        return html;
    }
```

2. Copia en el cuadro la operación para buscar una imagen por palabra clave en REST.

```
/**

* GET method to search images by keyword

* @param keywords

* @return

*/

@Path("searchKeywords/{keywords}")

@GET

@Produces(MediaType.TEXT_HTML)

public String searchByKeywords (@PathParam("keywords") String
keywords){

Connection connection = null;

html = cabeceras("Búsqueda palabras clave");

html += "<body><h2>Búsqueda por palabras clave</h2>";

try {

Class.forName("org.sqlite.JDBC");

connection =
DriverManager.getConnection("jdbc:sqlite:/Users/carles/Desktop/Practic
asAD/practica4/ADpractica4-proyecto/practica4.db");

PreparedStatement statement = connection.prepareStatement("select *
from image where keywords like ?");

statement.setString(1, '%' + keywords + '%');

ResultSet rs = statement.executeQuery();

while(rs.next()){

html = html + "<b> ID: </b>" + String.valueOf(rs.getInt("id")) +
"<br>";

html = html + "<b> Título: </b>" + rs.getString("title") + "<br>";

html = html + "<b> Descripción: </b>" + rs.getString("description") +
"<br>";

html = html + "<b> Keywords: </b>" + rs.getString("keywords") +
"<br>";

html = html + "<b> Autor: </b>" + rs.getString("author") + "<br>";

html = html + "<b> Fecha creación: </b>" +
rs.getString("creation_date") + "<br>";
```

```
html += "<br>";

}

html += "<a href='http://localhost:8080/practica4/'> <small> Volver al
menú </small> </a>";



}

catch(SQLException | ClassNotFoundException e ){

System.err.println("SQL EXCEPTION: " + e.getMessage());

return null;

}

finally{

try{

if(connection != null)

connection.close();

System.out.println("closed connection");

}

catch(SQLException e){

// connection close failed.

System.err.println(e.getMessage());

}

}


html += "</body> </html>";

return html;

}
```

3. Copia en el cuadro el código que llama a una de las operaciones del servicio web de imágenes en REST.

```
<h3><a href="/practica4/webresources/gestorImagenes/list">Listar
imágenes</a><br><br></h3>
```

Práctica 5

Compara los siguientes aspectos de la funcionalidad desarrollada en las prácticas 2, 3 y 4.

1. Facilidad de implementación de la parte cliente y la parte servidor.

La parte de cliente tiene muy pocas variaciones, pero consideramos que la más fácil de implementar sería la de REST, porque simplemente con las llamadas GET y POST a recibes el *html* con toda la información mientras que el servicio SOAP te devuelve una estructura de datos la cual debes tratar en el cliente.

La parte de servidor es inversamente proporcional, mientras que en REST tiene que formar todo el *html* para devolverlo en forma de un String, hecho que es más farragoso y en SOAP devuelves simplemente una estructurar de datos la cual ya está definida.

2. Tiempo de respuesta para la funcionalidad de registro de imagen. Para poder realizar la comparación, comenta la parte de upload de la página en la Práctica 2.

Practica 2: 103 ms //falta probar de nuevo

Practica 3: 129 ms → Navegador – WebAppClient – WebService

Practica 4: 90 ms → Navegador – Servicio REST

3. Compara el formato de las peticiones y las respuestas en SOAP y REST. ¿Cómo se realiza el envío de objetos complejos como por ejemplo las listas en ambos servicios?

En SOAP la lista que se transmite en una *list* del objeto *Image* que hemos definido en la práctica mientras que en REST se transmite ya el *html* formado con la lista hecha la cual ha procesado el servidor.

Todas las prácticas

1. Detalla las ampliaciones que hayas realizado en cada práctica. Algunos ejemplos de ampliaciones son: funcionalidades extra de gestión de imágenes (p. e. borrado), jsp para gestión de errores, funciones extra de búsqueda, etc. Puedes copiar el código correspondiente a cada ampliación.

Los cambios y mejoras introducidos durante esta práctica se encuentran más detallados en el informe de la práctica 5, aquí se incluirán los más relevantes.

Todas las prácticas:

- La contraseña se pide dos veces a la hora del registro.

- Se ha implementado control de sesiones en todas las prácticas.

Practica2:

- Aparte de las ampliaciones ya hechas para la primera entrega, ahora la contraseña se pide 2 veces al hacer el registro y el fichero se guarda como “user_nombreFichero”.

Practica3:

- Ahora se reutiliza la estructura de la práctica 2 para navegación del usuario entre las funcionalidades del proyecto.

- Se ha implementado gestión de usuarios

Practica4:

- Se han arreglado los errores relacionados con la modificación de imágenes

- Se ha implementado gestión de usuarios