

# **INFORME PRÁCTICA 2 - APP WEB**

Carles Salvador Bertran, Rubén Sanz García

Aplicacions Distribuïdes  
07/10/2019

# Índice

<b>Introducción</b>	<b>2</b>
<b>Signup, login y logout</b>	<b>2</b>
<b>Menú</b>	<b>3</b>
<b>Registrar imagen</b>	<b>4</b>
<b>Listar imágenes</b>	<b>5</b>
<b>Modificar imagen</b>	<b>6</b>
<b>Buscar imagen</b>	<b>7</b>

# 1. Introducció

El objetivo de esta práctica es realizar una aplicación web con *NetBeans*. Esta aplicación web se basa en la gestión de imágenes. Habrá usuarios que podrán registrarse y hacer *login*, registrar nuevas imágenes, modificar los registros de las imágenes ya introducidas por ellos, listar todas las imágenes y buscar imágenes a partir de sus características.

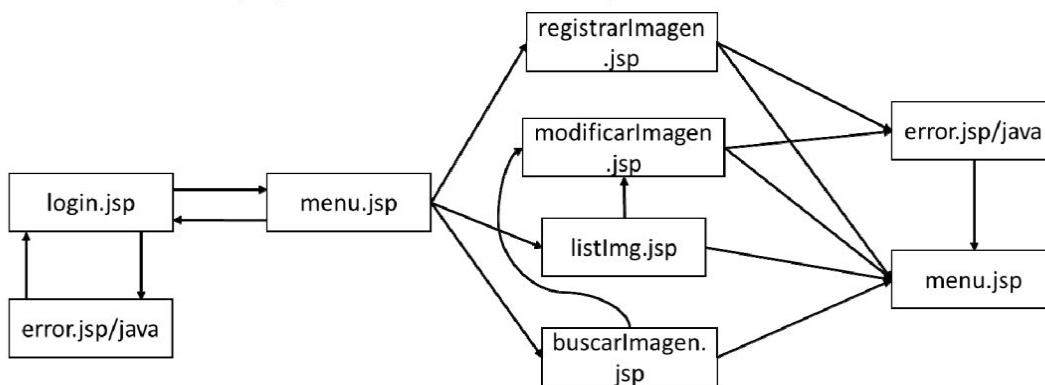


Fig 1. Flujo entre páginas

## 2. Signup, login y logout

La aplicación web requiere de usuarios para subir/modificar las imágenes. Por lo tanto necesitamos de un sistema de gestión de *login* y *logout*. La página *login.jsp* será la primera página que verá el usuario si entra por primera vez a la aplicación. Ésta página simplemente se trata de un formulario con dos campos: usuario y contraseña.

```
<form action="login" method="POST">
  Usuari:
  <input type="text" name="user" /> <br>
  Contrasenya:
  <input type="password" name="password" /> <br>
  <input type="submit" value="Enviar" /> <br><br>
</form>
```

Cuando el usuario envía estos parámetros, se tratan en el pertinente servlet, *login.java*. En el servlet hacemos las consultas con la base de datos para comprobar que el usuario introducido existe y la contraseña asociada es correcta.

En el caso que el usuario que intenta hacer el *login* exista y los datos introducidos sean los correctos, se le redirige hacia *menu.jsp*. En caso contrario, redirigimos a la página *error.jsp*. La página *error.jsp* la usaremos durante toda la aplicación web para redirigirnos a ella en caso que el usuario introduzca datos erróneos y consiste en un simple texto de “error” y un botón para ir de vuelta al menú.

Algo a tener en cuenta aquí es que al trabajar con sesiones, una vez el login sea correcto, debemos establecer el nombre del usuario como atributo a la sesión (*sesion.setAttribute("usuario", username)*).

Para que la aplicación tuviera mayor consistencia decidimos implementar un logout, disponible desde el menú, dónde simplemente invalidamos la sesión actual.

Por último, necesitamos una página para la creación de nuevos usuarios, *signup.jsp*. Muy parecido al login, aquí simplemente pedimos al usuario que introduzca su nombre de usuario y contraseña, pero esta vez enviamos los datos al servlet *signup.java*. Allí introducimos los datos a nuestra base de datos. En el caso que el usuario ya exista, saltará una *SQLException* ya que el nombre de usuario es *primary key*, y redirigiremos a la página error.

### 3. Menú

El menú es la página principal de la aplicación. Si el usuario ha introducido correctamente los datos al hacer login, ésta será la primera página que verá.

Se trata de una página sencilla donde simplemente hay enlaces a las páginas jsp de gestión de las imágenes.

```
<h1>BIENVENIDO</h1> ${sessionScope.usuario}  
<br>  
<a href="http://localhost:8080/ADpractica2/registrarImagen.jsp">  
  <h2>Registrar Imagen</h2>  
</a>  
<a href="http://localhost:8080/ADpractica2/list.jsp">  
  <h2>Mostrar imágenes</h2>  
</a>  
<a href="http://localhost:8080/ADpractica2/buscarImagen.jsp">  
  <h2>Buscar imagen</h2>  
</a>  
  
<a href="http://localhost:8080/ADpractica2/logout.jsp">  
  <small> Logout </small>  
</a>
```

También tenemos el enlace a *logout* en caso que el usuario quiera cerrar la sesión.

## 4. Registrar imagen

Para registrar una imagen le pedimos al usuario, mediante un formulario, los siguientes datos:

- Título de la imagen
- Descripción
- Palabras clave (separadas por comas y sin espacios)
- Fecha de creación
- Archivo de la imagen

Destacar que los campos título, descripción y palabras clave son de tipo *text*. En cambio, el campo fecha de creación es de tipo *date* y el campo para introducir el archivo de la imagen es de tipo *file*.

Aparte de estos campos, para registrar la imagen usamos más variables: *id*, fecha de alta, *i* y nombre del fichero. Estos 3 campos son asignados automáticamente en el servlet.

Para asignar la fecha actual hacemos uso de las clases *LocalDateTime* y *DateTimeFormatter*

```
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");  
LocalDateTime dateTime = LocalDateTime.now();  
storage_date = dateTime.format(formatter);
```

Para subir el fichero introducido en el formulario, primero de todo necesitamos que el formulario de *registrarImagen.jsp* tenga el atributo *enctype="multipart/form-data"*. Esto nos permitirá leer correctamente el archivo. Seguidamente, en el servlet hemos usado las clases

*Part*, *File* y *InputStream* para leer, crear y copiar el archivo en un directorio accesible por la página web.

```
//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\\Imagenes");
File fichero = new File(uploads, fileName);
try (InputStream input = fileContent) {
    Files.copy(input, fichero.toPath());
}
}
```

Como podemos ver, guardamos todas las imágenes en el directorio *imagenes* dentro de la carpeta *web* de nuestro proyecto.

Por último, para asignar el *id* a nuestra imagen primero hacemos una consulta para saber cuál es el mayor *id* actual en nuestra base de datos y le sumamos 1 para hacer un autoincremento y que cada imagen tenga un id mayor a la anterior.

```
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;
```

## 5. Listar imágenes

En esta página mostraremos los títulos de todas las imágenes registradas en el sistema. Cada título mostrado será un enlace para poder visualizar esa misma imagen.

Para conseguir esto hemos tenido que hacer una consulta a la base de datos de todos los títulos y su *filename* para así introducir el nombre del fichero al final del enlace en el atributo *href* del enlace.

```
out.write("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href=\"http://localhost:8080/ADpractica2/Imagenes/" + p.getValue() + "\">" + p.getKey() + "</a>");
```

Como podemos ver en la línea de código anterior, *p* es un *Pair* que contiene como llave el título de la imagen y como valor el *filename*.

Además de listar todas las imágenes, necesitábamos añadir la opción al usuario de poder modificar aquellas que él mismo ha subido. Haciendo una consulta a la base de datos sobre qué imágenes tienen de autor al mismo usuario que está logueado en la sesión actual obtenemos una lista de todas las imágenes subidas por él.

Tenemos una lista de todos los *filename* del usuario. Por lo tanto, si la imagen pertenece al usuario mostraremos otro enlace que redirige a *modificarImagen.jsp* pasándole por parámetro el *filename* en cuestión.

## 6. Modificar imagen

Página que sirve para modificar los datos de una imagen. Se puede acceder a ella desde la página de listar imágenes y desde buscar imágenes.

Para saber qué imagen queremos modificar le debemos pasar por parámetros URL el *filename* de la imagen. En el *modificarImagen.jsp* cogeremos el parámetro con *request.getParameter("imagen")*.

Una vez sepamos la imagen a modificar, mostraremos un formulario con los datos de ésta para que el usuario, si quiere, pueda modificarlos. También hemos añadido un *checkbox* para confirmar si el usuario quiere cambiar la imagen asociada.

La nueva información será enviada al servlet para modificar la base de datos con los nuevos datos.

### Modifica:

Título:

Descripción:

Palabras clave (separadas por comas y sin espacios):

☐ Marca esta casilla si quieres cambiar el fichero asociado a una imagen mediante el campo siguiente

Imagen sustituta:

Ningún archivo seleccionado

[Volver al menú](#)

## 7. Buscar imagen

Página que permite buscar imágenes a partir de los campos asociados a la imagen. Para las imágenes que sean del usuario que hace la búsqueda aparece la posibilidad de modificar la imagen (enlace a la página *modificarImagen*). Nosotros hemos utilizado los campos título, descripción y palabras clave.

En la página *buscarImagen.jsp* mostramos un formulario para que el usuario introduzca los términos a buscar separados por comas y sin espacios. Enviamos estos datos al servlet para hacer la consulta a la base de datos y dependiendo del resultado mostrarlos.

La lógica que usamos en nuestra búsqueda es que cualquiera de las palabras introducidas coincida con alguna de los campos explicados anteriormente de las imágenes. En el caso de que coincidan los campos con las palabras introducidas mostraremos un enlace a esa imagen. Además, si el usuario es el autor de alguna de estas imágenes mostraremos un enlace a *modificarImagen.jsp* de la misma forma que hacíamos en el *list.jsp*.

```
// Query por titulo, descripcion y keywords
out.write("Resultados relacionados con " + s + "<br>");
query = "select title, filename from image where title like '%" + s + "%' or description like '%" + s + "%' or "
      + "keywords like '%" + s + "%'";
statement = connection.prepareStatement(query);
rs = statement.executeQuery();
if(rs.next() == false){
    out.write("&nbsp;&nbsp;&nbsp;No se han obtenido resultados :(<br>");
}
else{
    do{
        out.write("&nbsp;&nbsp;&nbsp;<a href=\"Imagenes/\" + rs.getString(2) + \"\">
              + rs.getString(1) + "</a>");

        //si es una imagen del autor, opcion a modificar
        if(authorPics.contains(rs.getString(2)))
            out.write ("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href=\"modificarImagen.jsp?imagen="
                  + rs.getString(2) + "\"> Modificar Imagen"
                  + "</a>");
        out.write("<br>");
    }while(rs.next());
}
```