

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.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");

    // Create path components to save the file
    final String path = ("C:\\Users\\rando\\OneDrive\\Documents\\GitHub\\FIB-AD\\Laboratori\\P2\\WAP1\\web\\Image");
    final Part filePart = request.getPart("imatge");
    String fileName = getFileName(filePart);
    int id;
    String titol = request.getParameter("titol");
    String descripcio = request.getParameter("descripcio");
    String tags = request.getParameter("tags");
    String autor = request.getParameter("autor");
    String datac = request.getParameter("datac");
    String user=null;
    Cookie[] cookies = request.getCookies();
    if(cookies !=null){
        for(Cookie cookie : cookies){
            if(cookie.getName().equals("username")) user = cookie.getValue();
        }
    }

    Date date= new Date();
    long time = date.getTime();
    String timestamp = Long.toString(time);
    OutputStream out = null;
    InputStream filecontent = null;
    final PrintWriter writer = response.getWriter();
    Connection conn = null;

    int lastDot = fileName.lastIndexOf('.');
    fileName = timestamp + fileName.substring(lastDot);

    try {

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

        // Connexió Sergi
        try {
            conn = DriverManager.getConnection("jdbc:sqlite:C:\\Users\\rando\\OneDrive\\Documents\\GitHub\\FIB-AD\\Laboratori\\P2\\WAP1\\FotOK.db");
            Statement statement = conn.createStatement();
            statement.setQueryTimeout(30);
        }
        // Connexió Marc
        catch(SQLException e) {
            try{
                conn = DriverManager.getConnection("jdbc:sqlite:FotOK.db");
                Statement statement = conn.createStatement();
                statement.setQueryTimeout(30);
            }
            catch(SQLException ex){
                System.out.println("No es troba cap base de dades d'usuaris");
            }
        }

        PreparedStatement getId = conn.prepareStatement("SELECT MAX(id) FROM imatges");
        ResultSet rs = getId.executeQuery();
        if(rs.next()) id = rs.getInt(1)+1;
        else id=0;

        try{
            PreparedStatement pujafoto = conn.prepareStatement("INSERT INTO imatges(filename,id,titol,descripcio,tags,autor,datac,timestamp,username) VALUES(?,?,?,?,?,?,?,?,?)");
            pujafoto.setString(1,fileName);
            pujafoto.setInt(2,id);
            pujafoto.setString(3,titol);
            pujafoto.setString(4,descripcio);
            pujafoto.setString(5,tags);
            pujafoto.setString(6,autor);
            pujafoto.setString(7,datac);
            pujafoto.setString(8,timestamp);
            pujafoto.setString(9,user);
            pujafoto.executeUpdate();
        }
        catch (SQLException e){
            response.sendRedirect("error.jsp");
        }
        out = new FileOutputStream(new File(path + File.separator + fileName));
        filecontent = filePart.getInputStream();

        int read = 0;
        final byte[] bytes = new byte[1024];
```

```

while ((read = filecontent.read(bytes)) != -1) {
    out.write(bytes, 0, read);
}
writer.println("New file " + fileName + " created at " + path);
LOGGER.log(Level.INFO, "File{0}being uploaded to {1}",
    new Object[]{fileName, path});
response.sendRedirect("UploadOK.jsp");
}

catch (FileNotFoundException fne) {
    writer.println("You either did not specify a file to upload or are "
        + "trying to upload a file to a protected or nonexistent "
        + "location.");
    writer.println("<br/> ERROR: " + fne.getMessage());

    LOGGER.log(Level.SEVERE, "Problems during file upload. Error: {0}",
        new Object[]{fne.getMessage()});
    response.sendRedirect("error.jsp");
}
catch (SQLException ex) {
    Logger.getLogger(registrarImagen.class.getName()).log(Level.SEVERE, null, ex);
}
catch (ClassNotFoundException ex) {
    Logger.getLogger(registrarImagen.class.getName()).log(Level.SEVERE, null, ex);
}
finally {
    if (out != null) {
        out.close();
    }
    if (filecontent != null) {
        filecontent.close();
    }
    if (writer != null) {
        writer.close();
    }
    try {
        if (conn != null)
            conn.close();
    }
    catch (SQLException e) {
        // connection close failed.
        System.err.println(e.getMessage());
    }
}
}

private String getFileName(final Part part) {
    final String partHeader = part.getHeader("content-disposition");
    LOGGER.log(Level.INFO, "Part Header = {0}", partHeader);
    for (String content : part.getHeader("content-disposition").split(";")) {
        if (content.trim().startsWith("filename")) {
            return content.substring(
                content.indexOf('=') + 1).trim().replace("\"", "");
        }
    }
    return null;
}
}

```

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 class="form-signin" action="registrarImagen" method="POST" enctype="multipart/form-data">

    <input class="form-control" type="file" id="imatge" name="imatge" required autofocus>
    <input class="form-control" type="text" name="titol" placeholder="Títol" required>
    <input class="form-control" type="text" name="descripcio" placeholder="Descripció" required>
    <input class="form-control" type="text" name="tags" placeholder="Tags separats amb ',' Exemple: (naturalesa;animals;maincra) " required>
    <input class="form-control" type="text" name="autor" placeholder="Autor" required>
    <input class="form-control" type="date" name="datac" required>

    <button class="btn btn-lg btn-primary btn-block" type="submit">Puja</button>
</form>

```

Práctica 3:

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

```

@WebMethod(operationName = "registerImage")
public int registerImage(@WebParam(name = "image") ImageWS image) {

    Date date = new Date();
    long time = date.getTime();
    String timestamp = Long.toString(time);
    Connection conn = null;
    int id;

    try {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
    }
}

```

```

// create a database connection
conn = DriverManager.getConnection("jdbc:derby://localhost:1527/FotOK;user=mcasellas;password=1234");
Statement statement = conn.createStatement();
statement.setQueryTimeout(30);

PreparedStatement getid = conn.prepareStatement("SELECT MAX(id) FROM imatges");
ResultSet rs = getid.executeQuery();
if (rs.next()) {
    id = rs.getInt(1) + 1;
} else {
    id = 0;
}

String filename = timestamp + ".jpg";

try {
    PreparedStatement pujafoto = conn.prepareStatement("INSERT INTO imatges(filename,id,titol,descripcio,tags,autor,datac,timestamp,username)
VALUES(?,?,?,?,?,?,?,?)");
    pujafoto.setString(1, filename);
    pujafoto.setInt(2, id);
    pujafoto.setString(3, image.getTitol());
    pujafoto.setString(4, image.getDescripcio());
    pujafoto.setString(5, image.getTags());
    pujafoto.setString(6, image.getAutor());
    pujafoto.setString(7, image.getDatac());
    pujafoto.setString(8, timestamp);
    pujafoto.setString(9, image.getUsername());
    pujafoto.executeUpdate();

} catch (SQLException e) {
    return 0;
}

} catch (SQLException ex) {
    System.out.println("No es troba cap base de dades d'usuari");
} catch (ClassNotFoundException ex) {
    Logger.getLogger(FotOKWS.class.getName()).log(Level.SEVERE, null, ex);
} finally {
    try {
        if (conn != null) {
            conn.close();
        }
    } catch (SQLException e) {
        // connection close failed.
        System.err.println(e.getMessage());
    }
}

return 1;
}

```

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

```

@WebMethod(operationName = "searchByTitle")
public List searchByTitle(@WebParam(name = "title") String title) {
    //TODO write your implementation code here:
    Connection con = null;
    ArrayList<ImageWS> resultat = new ArrayList<ImageWS>();
    try {

        Class.forName("org.apache.derby.jdbc.ClientDriver");

        // create a database connection
        con = DriverManager.getConnection("jdbc:derby://localhost:1527/FotOK;user=mcasellas;password=1234");
        Statement statement = con.createStatement();
        statement.setQueryTimeout(30);
        try {
            PreparedStatement getphotos = con.prepareStatement("SELECT * FROM imatges WHERE titol LIKE ?");
            getphotos.setString(1, title);
            ResultSet rs = getphotos.executeQuery();
            while (rs.next()) {
                ImageWS temp = new ImageWS();
                temp.filename = rs.getString("filename");
                temp.id = rs.getInt("id");
                temp.titol = rs.getString("titol");
                temp.descripcio = rs.getString("descripcio");
                temp.tags = rs.getString("tags");
                temp.autor = rs.getString("autor");
                temp.datac = rs.getString("datac");
                temp.timestamp = rs.getString("timestamp");
                temp.username = rs.getString("username");

                resultat.add(temp);
            }
        } catch (SQLException e) {
            System.err.println(e.getMessage());
            return null;
        }

    } catch (SQLException e) {
        System.err.println(e.getMessage());
        return null;
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(FotOKWS.class.getName()).log(Level.SEVERE, null, ex);
    } finally {

```

```

    try {
        if (con != null) {
            con.close();
        }
    } catch (SQLException e) {
        // connection close failed.
        System.err.println(e.getMessage());
        return null;
    }
}

return resultat;
}

```

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

```

<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>FotOK</title>
    <link href="/css/bootstrap.min.css" rel="stylesheet">
    <link href="/css/bootstrap-theme.min.css" rel="stylesheet">
    <link href="theme.css" rel="stylesheet">
</head>

<body>

<nav class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">FotOK</a>
        </div>
        <div id="navbar" class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li><a href="/menu.jsp">Inici</a></li>
                <li class="active"><a href="/registrarImagen.jsp">Registrar Imatge</a></li>
                <li><a href="/list.jsp">Llista les imatges</a></li>
                <li><a href="/buscarImagen.jsp">Busca una imatge</a></li>
                <li><form class="form-signin" action="logout" method="POST">

</form> </li>
            </ul>
        </div><!-- /.nav-collapse -->
    </div>
</nav>

<div class="container theme-showcase" role="main">

    <!-- Main jumbotron for a primary marketing message or call to action -->

    <div class="page-header">
        <h1>Registrar imatge:</h1>
    </div>

    <div class="row">

        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title">Nova imatge</h3>
            </div>
            <div class="panel-body">
                <form class="form-signin" action="registrarImatge" method="POST">

                <input class="form-control" type="text" name="titol" placeholder="Títol" required>
                <input class="form-control" type="text" name="descripcio" placeholder="Descripció" required>
                <input class="form-control" type="text" name="tags" placeholder="Tags separats amb ',' Example: (naturalesa;animals;maincra) " required>
                <input class="form-control" type="text" name="autor" placeholder="Autor" required>
                <input class="form-control" type="date" name="datac" required>

```

```

        <button class="btn btn-lg btn-primary btn-block" type="submit">Puja</button>
    </form>

    </div>
</div>

</div>
</div> <!-- /container -->

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../../assets/js/vendor/jquery.min.js"></script>')</script>
<script src="/js/bootstrap.min.js"></script>
<script src="/js/docs.min.js"></script>
<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
<script src="/js/ie10-viewport-bug-workaround.js"></script>
</body>
</html>

```

Pràctica 4:

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

Para modificar una imagen usamos dos operaciones, una para modificar los campos, y otra que se encarga de hacer la modificación en la base de datos.

```

/**
 * GET method to modify by id
 * @param id
 * @return
 */
@Path("modify")
@POST
@Consumes(MediaType.APPLICATION_FORM_URLENCODED) @Produces(MediaType.TEXT_HTML)
public String modifyImage (@FormParam("id") int id) throws ClassNotFoundException {
    //TODO write your implementation code here:
    Connection con = null;
    String resultat = "Error";
    try {

        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            con = DriverManager.getConnection("jdbc:derby://localhost:1527/FotOK;user=mcasellas;password=1234");

            Statement statement = con.createStatement();

            statement.setQueryTimeout(30);

        } catch (SQLException e) {
            System.out.println("No es troba cap base de dades d'usuari");
            System.err.println(e.getMessage());
        }
        try {

            PreparedStatement getImatge = con.prepareStatement("SELECT * FROM IMATGES WHERE ID = ?");

            getImatge.setInt(1, id);
            ResultSet rs = getImatge.executeQuery();

            while (rs.next()) {
                resultat = "<form class='form-signin' action='http://localhost:8080/RestAD/webresources/generic/modify_values/' + rs.getString("id") + "' method='POST'>"
                + " <input class='form-control' value='" + rs.getString("titol") + "' type='text' name='title' placeholder='Títol' required>"
                + " <input class='form-control' value='" + rs.getString("descripcio") + "' type='text' name='description' placeholder='Descripció' required>"
                + " <input class='form-control' value='" + rs.getString("tags") + "' type='text' name='keywords' placeholder='Tags separats amb ; Exemple:"
                + "(naturalesa;animals;maincra)' required>"
                + " <input class='form-control' value='" + rs.getString("autor") + "' type='text' name='author' placeholder='Autor' required>"
                + " <input class='form-control' value='" + rs.getString("datac") + "' type='date' name='creation' required>"
                + " <button class='btn btn-lg btn-primary btn-block' type='submit'>Puja</button>"
                + " </form>";
            }

        } catch (SQLException e) {
            System.err.println(e.getMessage());
        }
    } catch (ClassNotFoundException ex) {
        System.err.println(ex.getMessage());
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            // connection close failed.
            System.err.println(e.getMessage());
        }
    }
}

```

```

        return banner + "Modificar imatge:" + mig + resultat + footer;
    }

/**
 * POST method to register a new image
 * @param id
 * @param title
 * @param description
 * @param keywords
 * @param author
 * @param crea_date
 * @return
 */
@Path("modify_values/{id}")
@POST
@Consumes(MediaType.APPLICATION_FORM_URLENCODED) @Produces(MediaType.TEXT_HTML)
public String modifyImage (@PathParam("id") int id, @FormParam("title") String title,
    @FormParam("description") String description, @FormParam("keywords") String keywords, @FormParam("author") String author, @FormParam("creation") String creation){

    Connection con = null;
    String resultat = "<h3>No s'ha pogut modificar la imatge<h3>";
    try {

        try {

            Date date = new Date();
            long time = date.getTime();
            String timestamp = Long.toString(time);
            String filename = timestamp + ".jpg";

            Class.forName("org.apache.derby.jdbc.ClientDriver");
            con = DriverManager.getConnection("jdbc:derby://localhost:1527/FotOK;user=mcasellas;password=1234");

            PreparedStatement updateFoto = con.prepareStatement("UPDATE IMATGES SET TITOL = ?, DATAC = ?, TAGS = ?, DESCRIPCIO = ?, AUTOR = ? WHERE ID = ?");
            updateFoto.setString(1, title);
            updateFoto.setString(2, creation);
            updateFoto.setString(3, keywords);
            updateFoto.setString(4, description);
            updateFoto.setString(5, author);
            updateFoto.setInt(6, id);

            int num = updateFoto.executeUpdate();

            resultat = "<h3>S'ha modificat la imatge correctament<h3>"
+ "<div class='list-group'><div class='list-group'>"
+ "<a href='./webresources/generic/list' class='list-group-item active'>Llista les imatges</a>"
+ ""
+ "</div>";

        } catch (SQLException e) {
            System.err.println(e.getMessage());
        }
    } catch (ClassNotFoundException ex) {
        System.err.println(ex.getMessage());
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            // connection close failed.
            System.err.println(e.getMessage());
        }
    }

    return banner + "Modificar imatges:" + mig + resultat + footer;
}

```

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) throws ClassNotFoundException{
    Connection con = null;
    String resultat = "<h3>No hi han imatges<h3>";
    try {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        con = DriverManager.getConnection("jdbc:derby://localhost:1527/FotOK;user=mcasellas;password=1234");

        PreparedStatement getphotos = con.prepareStatement("SELECT * FROM imatges WHERE TAGS LIKE ?");
        String cerca = '%' + keywords + '%';
    }
}

```

```

        getphotos.setString(1, cerca);
        ResultSet rs = getphotos.executeQuery();
        resultat = llistarImatges(rs);
    }
    catch (SQLException e) {
        System.out.println("Error amb la base de dades");
        System.err.println(e.getMessage());
    }
    return banner + "Buscar imatges per tags:" + mig + resultat + footer;
}

```

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

Código que llama a las operaciones de búsqueda de imagenes.

```

<%@page import="java.util.ArrayList"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>FotOK</title>
    <link href="/css/bootstrap.min.css" rel="stylesheet">
    <link href="/css/bootstrap-theme.min.css" rel="stylesheet">
    <link href="theme.css" rel="stylesheet">
    <link rel="shortcut icon" href="/favicon.ico">
  </head>

  <body>

    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">FotOK</a>
        </div>
        <div id="navbar" class="navbar-collapse collapse">
          <ul class="nav navbar-nav">
            <li><a href="/menu.jsp">Inici</a></li>
            <li><a href="/registrarImagen.jsp">Registrar Imatge</a></li>
            <li><a href="/webresources/generic/list">Llista les imatges</a></li>
            <li class="active"><a href="/buscarImagen.jsp">Busca una imatge</a></li>
            <li>
              <form class="form-signin" action="logout" method="POST"></form>
            </li>
          </ul>
        </div><!--/.nav-collapse -->
      </div>
    </nav>

    <div class="container theme-showcase" role="main">

      <!-- Main jumbotron for a primary marketing message or call to action -->

      <div class="page-header">
        <h1>Buscar imatges:</h1>
      </div>

      <div class="row">

        <div class="panel panel-default">
          <div class="panel-heading">
            <h3 class="panel-title">Nova imatge</h3>
          </div>
          <div class="panel-body">
            <form onchange="afegirPath()" id="formulari" class="form-signin" action="/webresources/generic/searchByID/" method="GET">
              <select name="accio" id="accio">
                <option value="ID">Id</option>
                <option value="Title">Titol</option>
                <option value="Author">Autor</option>
                <option value="CreationDate">Data de creació</option>
                <option value="Keywords">Tags</option>
              </select>
              <input class="form-control" type="text" id="text" name="text" required autofocus>
            </form>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

        <button class="btn btn-lg btn-primary btn-block" type="submit">Buscar</button>
    </form>

    </div>
</div>

</div>

</div> <!-- /container -->
<script>
    function afegirPath() {
        var accio = document.getElementById("accio").value;

        document.getElementById("text").type = accio == "ID" ? "number" : "text";

        var data = document.getElementById("text").value;
        var result = "../webresources/generic/search" + accio + '/' + data;
        document.getElementById("formulari").action = result;
    }
</script>

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../../assets/js/vendor/jquery.min.js"></script>')</script>
<script src="../../js/bootstrap.min.js"></script>
<script src="../../js/docs.min.js"></script>
<!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
<script src="../../js/ie10-viewport-bug-workaround.js"></script>
</body>
</html>

```

Pràctica 5:

1. Compara los siguientes aspectos de la funcionalidad desarrollada en las prácticas 2, 3 y 4. Facilidad de implementación de la parte cliente y la parte servidor.
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.
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?

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.