	<p style="text-align: center;">UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLÓGICOS</p>
<p style="text-align: center;">CICLO: 2/ 2014</p>	<p>GUIA DE LABORATORIO #2</p> <p>Nombre de la Practica: Subir archivos al Servidor</p> <p>Lugar de Ejecución: CENTRO DE COMPUTO</p> <p>Tiempo Estimado: 3 HORAS CLASE</p> <p>MATERIA: PROGRAMACIÓN ORIENTADA A OBJETOS II</p>

I. OBJETIVOS.

- Que el estudiante pueda crear una aplicación en la cual pueda subir archivos al servidor.
- Que el estudiante implemente lo visto en el desarrollo de sus tareas y proyectos.

III. PROCEDIMIENTO.

Subida de un archivo al servidor:

Para esta partes se subiran archivos al servidor utilizando la clase **MultipartRequest**.

Subir Archivos al servidor:

1. Para esta guía tendrá que crear un proyecto llamado Guia2POO2 y crear dentro del proyecto una carpeta llamada **img**(esta contiene toda las imágenes).
2. Agregar los siguientes .jar(será proporcionados por el docente):
 - jstl.jar
 - estándar.jar
 - mysql-conector.jar
 - cos.jar
3. Crear una página jsp que se llame **IngresoDatos.jsp** y agregar el siguiente código:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Ingreso de Datos </title>
</head>
<body>
  <form action="SubirArchivos" enctype="multipart/form-data" method="POST">
```

```

Ingrese sus Nombre: <input type="text" name="nombres" value="" /><br>
Ingrese sus Apellidos: <input type="text" name="apellidos" value="" /><br>
Buscar Fotografia: <input type="file" name="archivo"/><br>
<input type="submit" value="Subir Datos"/><input type="reset" value="Limpiar">
</form>
</body>
</html>

```

4. Crear el servlet **“SubirArchivos”** y proceder a eliminar el contenido del método **processRequest**(solamente el contenido en ningún momento eliminar el método o cualquier otro).

```

response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
try {
    //obtener conexiones
    Context init = new InitialContext();
    Context context = (Context) init.lookup("java:comp/env");
    DataSource dataSource = (DataSource)context.lookup("jdbc/mysql");
    Connection conexion = dataSource.getConnection();

    //obtener url física de la carpeta img
    String direc=getServletContext().getRealPath("img");
    System.out.println("URL: "+ direc);

    //Permite subir el archico 3 parametros
    //1. reques, 2. direccion fisica, 3. Tamaño
    MultipartRequest multi = new MultipartRequest(request,direc,1*1024*1024);

    //Obteniendo los campos del formulario
    String nombres=multi.getParameter("nombres");
    String apellidos=multi.getParameter("apellidos");
    String ruta=multi.getFile("archivo").toString();

    // obtiene el fichero
    File ficheroTemp = multi.getFile("archivo");
    out.println("nombres:"+nombres+"<br>");
    out.println("apellidos:"+apellidos+"<br>");
    out.println("url: "+ruta+"<br>");
}

```

//obtenemos el nombre del archivo

```
String nombreArchivo = ficheroTemp.getName();
```

```
Statement s = conexion.createStatement();
```

```
s.executeUpdate("Insert into empleados  
values("+null+",\""+nombres+"\", \""+apellidos+"\", \""+nombreArchivo+"\")");  
}  
catch(FileNotFoundException e){  
    out.println("La ruta de ubicacion no existe");  
}  
catch(IOException e){  
    out.print("El archivo seleccionado supera el maximo de 1MB");  
}  
catch (Exception e)  
{  
    e.printStackTrace();  
}  
finally {  
    out.close();  
}
```

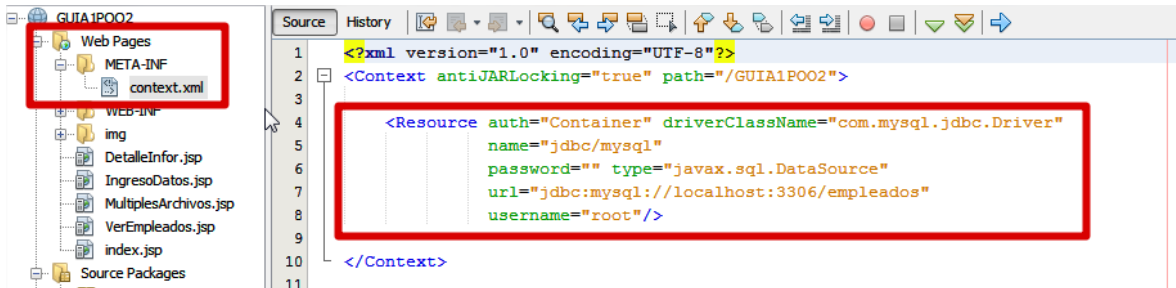
6. Para el servlet anterior debe importar las siguientes librerías.

```
import com.oreilly.servlet.MultipartRequest;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.PrintWriter;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.naming.Context;  
import javax.naming.InitialContext;  
import javax.servlet.ServletException;  
import javax.sql.DataSource;  
import java.sql.Connection;  
import java.sql.Statement;
```

5. Para poder implementar el ejemplo anterior se necesita crear una base de datos llamada **“empleados”** con una tabla llamada de la misma manera, deberá tener la siguiente estructura.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>id</u>	int(11)			No	None	auto_increment
<input type="checkbox"/>	nombres	varchar(40)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	apellidos	varchar(40)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	url_foto	varchar(300)	latin1_swedish_ci		No	None	

6. Crear el Pool de conexiones con apache tomcat.



Nota: Cambie los respectivos valores para la conexión al servidor de base de datos que usted va a utilizar.

7. Ahora proceder a correr el archivo **IngresoDatos.jsp**, deberá de verse de la siguiente manera:

Ingrese sus Nombre:

Ingrese sus Apellidos:

Buscar Fotografía:

8. Llenar los datos y dar click en “**Subir Datos**”, esto manda a llamar al servlet “**SubirArchivos**”, el cual realizara el proceso de inserción de datos en la base de datos y subirá de la imagen al servidor.

Ver contenido de la base de datos e Imagen asociada a la persona:

1. Crear una página jsp llamada “**VerEmpleados.jsp**” y agregar el siguiente código.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
```

```

<sql:query var="q1" dataSource="jdbc/mysql">
    SELECT * from Empleados
</sql:query>
<table>
    <tr><th>Id</th><th>Nombres</th><th>Apellidos</th><th>Información</th></tr>
<c:forEach var="name" items="{q1.rows}">
    <tr><td><c:out value="{name.id}"/></td><td><c:out value="{name.nombres}"/></td>
    <td><c:out value="{name.apellidos}"/></td><td><a href="DetalleInfor.jsp?id=<c:out
value="{name.id}"/>">Ver Información</a></td></tr>
</c:forEach>
</table>
</body>
</html>

```

2. Al correr esta página esta deberá verse de la siguiente manera:

Id	Nombres	Apellidos	Información
13	Rafael	Torres	Ver Información
14	susana	castellanos	Ver Información
15	irvin	monterroza	Ver Información
16	oscar	hen	Ver Información

3. Ahora nos falta crear la pagina que mostrar la imagen asociada a la persona, para ello crear una página jsp llamada “**DetalleInfor.jsp**” y agregarle el siguiente código.

```

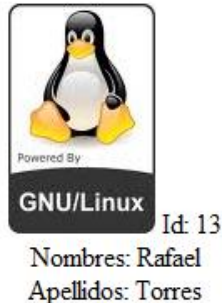
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <sql:query var="q1" dataSource="jdbc/mysql">
            SELECT * from Empleados where id=<c:out value="{param.id}"/>
        </sql:query>
        <center>

        <c:forEach var="name" items="{q1.rows}">
            " /><br/>
            Id: <c:out value="{name.id}"/><br>
            Nombres: <c:out value="{name.nombres}"/><br>
            Apellidos: <c:out value="{name.apellidos}"/><br>
        </c:forEach>
    </center>
    </body>
</html>

```

```
</center>
</body>
</html>
```

4. Ahora cuando de click en el link **Ver Información**, aparcerá la imagen y la información de la persona.



5. Ahora proceder a correr su proyecto desde otra máquina, dejar el tomcat levantado e irse a otra máquina e ingresar en la url la dirección de la maquina que corre su proyecto por ejemplo.

<http://10.0.25.58:8080/Guia4POO2/IngresoDatos.jsp>

Proceder a subir información y una imagen luego ver en su máquina servidor la información en la base de datos y la imagen subida en la ruta definida.

Subida de Múltiples archivos al Servidor

Para esta parte de la guía se pretende subir Múltiples archivos al servidor utilizando la clase **MultipartParser**, para ello seguir los siguientes puntos.

1. Crear un archivo jsp llamado **MultipleArchivos.jsp** y agregar el siguiente código.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
</head>
<body>
  <h1>Subida de Multiples Archivos</h1>
  <form name="multiples" action="SubirMultiplesArchivos" method="POST"
enctype="multipart/form-data">
    <table>
      <tr>
        <td>Ingrese sus Nombre: </td>
        <td><input type="text" name="nombres" value="" /> </td>
      </tr>
      <tr>
        <td>Ingrese sus Apellidos: </td>
        <td><input type="text" name="apellidos" value="" /> </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

```

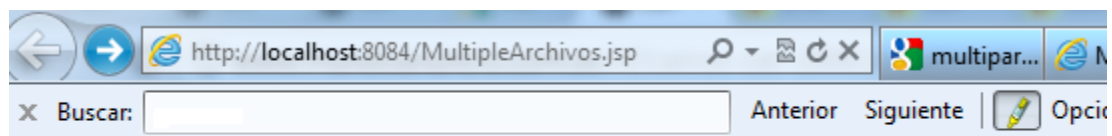
</tr>
<tr>
  <td>Ingrese su Foto: </td>
  <td><input type="file" name="file1"> </td>
</tr>
<tr>
  <td>Ingrese su Curriculum: </td>
  <td><input type="file" name="file2"></td>
</tr>
<tr>
  <td>Ingrese una copia del DUI: </td>
  <td><input type="file" name="file3"> </td>
</tr>
</table>

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

</body>
</html>

```

Este se visualizara de la siguiente manera:



Subida de Multiples Archivos

Ingrese sus Nombre:	<input type="text" value="Rafael"/>	
Ingrese sus Apellidos:	<input type="text" value="Torres"/>	
Ingrese su Foto:	<input type="text" value="C:\Users\Rafael\Desktop"/>	<input type="button" value="Examinar..."/>
Ingrese su Curriculum:	<input type="text" value="C:\Users\Rafael\Desktop"/>	<input type="button" value="Examinar..."/>
Ingrese una copia del DUI:	<input type="text" value="C:\Users\Rafael\Desktop"/>	<input type="button" value="Examinar..."/>
<input type="button" value="Enviar"/>		

- Ahora deberá de crear un servlet llamado **SubirMultiplesArchivos** y agregarle el siguiente codigo en el método **processRequest** .

```

try{

    // Tamaño Limite 1 MB

```

```

MultipartParser parser = new MultipartParser(request, 1 * 1024 * 1024);

//Url donde se guardaran las imagenes
String direc=getServletContext().getRealPath("img");

Part _part = null;
//Permite recorrer cada uno de los campos sean estos texto o file
while ((_part = parser.readNextPart( )) != null) {
    //Si es un tipo de campo File
    if (_part.isFile( )) {

        //Obtener información del archivo
        FilePart fPart = (FilePart) _part;
        String name = fPart.getFileName( );
        //Si el campo no venia vacío realizara esta parte
        if (name != null) {
            //Escribe el archivo en el servidor segun la ruta brindada
            //Devuelve el tamaño del archivo
            long fileSize = fPart.writeTo(new java.io.File(direc));
            //Nombre del archivo
            out.println("The user's file path for the file: " +
                fPart.getFilePath( ) + "<br>");

            //Tipo de contenido del archivo
            out.println("The content type of the file: " +
                fPart.getContentType( ) + "<br>");

            //Tamaño del archivo
            out.println("The file size: " +fileSize+ " bytes<br><br>");

            //commence with another file, if there is one

        } else {
            //Si uno de estos campos tipo File viene vacío ingresa esta parte
            out.println("The user did not upload a file for this part.");
        }

    }

    //Si el tipo de campo es tipo text ingresa a esta parte del código
    else if (_part.isParam( )) {
        // do something else if it is a non-file-type parameter,
        //such as a username

        ParamPart paramPart = (ParamPart) _part;
        String value = paramPart.getStringValue();
        out.println("El valor de los campos son : " + value+"<br><br>");
    }

}

} // end while

```



```

out.println("</body>");
out.println("</html>");

    } catch (java.io.IOException ioe){
        out.print("El archivo seleccionado supera el maximo de 1MB"+ioe);
    }
    finally {
        out.close();
    }
}

```

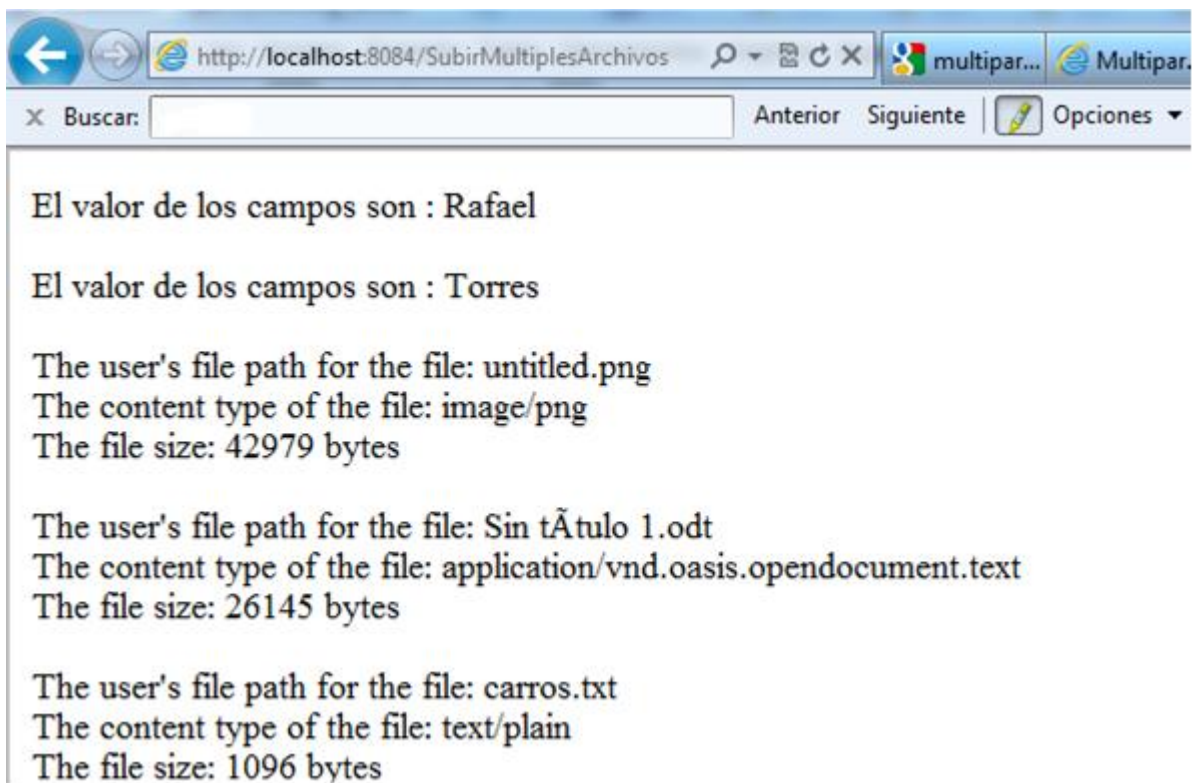
3. Al servlet anterior deberá importar las siguientes clases:

```

import com.oreilly.servlet.multipart.FilePart;
import com.oreilly.servlet.multipart.MultipartParser;
import com.oreilly.servlet.multipart.ParamPart;
import com.oreilly.servlet.multipart.Part;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


```

El resultado será similar al siguiente:



III. EJERCICIOS COMPLEMENTARIOS.

- Al ejemplo realizado usted deberá de completarlo de tal manera que se tenga un menú principal que me permita ir a los link: Ingresar Empleado, Ver Información Empleado. Tomar en cuenta que debe de agregar una hoja de estilo para que este se vea mucho más presentable.
- Para la página VerEmpleados.jsp usted deberá mostrar el listado con el displaytag y agregara una nueva columna que diga generar Reporte. Al dar click a los link de cada empleado deberá crear un reporte con su información y con la foto que este posea tal y como se muestra en la siguiente imagen.

Reporte		
		
ID	Nombres	Apellidos
5	Alejandro	Figueroa

- Para la parte de subir Multiples archivos usted deberá de guardar los datos que sean de tipo **text** en un arreglo para luego guardarlos en la base de datos y a la vez los nombres de los archivos deberán de guardarse en un arreglo para ingresarlo en un campo en la base de datos separados por una “,” por ejemplo.

Carlos	Orellana	Foto1.jpg,CV1.doc,DUI.jpg
Manuel	Hernández	FotoManuel.jpg,CVm.doc,D1.jpg

IV. REFERENCIA BIBLIOGRAFICA.

- <http://www.servlets.com/cos/>

Ultima Fecha de visita (19/06/2011)