

# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

Web App Development.

Ejercicio 6: Subir imagen (file upload)

*Profesor: M. en C. José Asunción Enríquez Zárate*

*Alumno: Aarón Antonio García González*

*aarongarcia.ipn.escom@gmail.com*

*3CM9*

December 22, 2020

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introducción</b>   | <b>2</b>  |
| <b>2</b> | <b>Conceptos</b>  | <b>3</b>  |
| 2.1      | Jasper Reports . . . . .  | 3         |
| 2.2      | jsPDF . . . . .   | 3         |
| <b>3</b> | <b>Desarrollo</b>   | <b>4</b>  |
| 3.1      | Modificar <i>formUsuario.jsp</i> . . . . .                        | 4         |
| 3.2      | Modificar <i>UserServlet.java</i> — <i>crearUsuario</i> . . . . . | 8         |
| 3.3      | Modificar <i>listaUsuario.jsp</i> . . . . .                       | 9         |
| <b>4</b> | <b>Resultados</b>   | <b>12</b> |
| 4.1      | Ejecución del Programa . . . . .                                  | 12        |
| 4.1.1    | Capturas de funcionamiento . . . . .                              | 12        |
| <b>5</b> | <b>Conclusión</b>   | <b>15</b> |
| <b>6</b> | <b>Referencias Bibliográficas</b>                                 | <b>16</b> |

Lista de figuras

|   |   |    |
|---|---|----|
| 1 | Jasper Report Entorno . . . . .             | 3  |
| 2 | Java Script PDF . . . . .                   | 3  |
| 3 | Formulario de registro de usuario . . . . . | 12 |
| 4 | Selección de imagen . . . . .               | 13 |
| 5 | Lista de usuarios . . . . .                 | 14 |
| 6 | Información de usuarios . . . . .           | 14 |

**Lista de tablas**

# 1 Introducción

Los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

El reporte, de esta forma, confiere una mayor utilidad a los datos. No es lo mismo trabajar con una planilla de cálculos con 10.000 campos que con un dibujo en forma de torta que presenta dichos campos de manera gráfica. Los reportes tienen diversos niveles de complejidad, desde una lista o enumeración hasta gráficos mucho más desarrollados.

De la misma forma, gracias a los reportes cualquier persona puede proceder a realizar un resumen de datos o a clasificar estos en grupos determinados. Por todo ello, se entiende que estos documentos sean tan importantes en cualquier empresa ya que gracias a ellos cuenta con sus propias bases de datos (de trabajadores, de clientes, de proveedores. . .), realiza recopilaciones de las ventas acometidas en un periodo determinado.

Así, teniendo en cuenta los datos que abordan y la extensión que tienen, estos reportes se pueden luego clasificar en diversos niveles en cuanto a su complejidad dando lugar a catálogos, a listas, a resúmenes, etc.[DefinicionDe, 2020]

## 2 Conceptos

### 2.1 Jasper Reports

JasperReports es una biblioteca de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML. Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

JasperReports se usa comúnmente con iReport, un front-end gráfico de código abierto para la edición de informes, si bien a partir de la versión 5.5.0 iReport ha sido sustituido por Jaspersoft Studio, un front-end gráfico de código abierto basado en Eclipse. [Wikipedia, 2020]

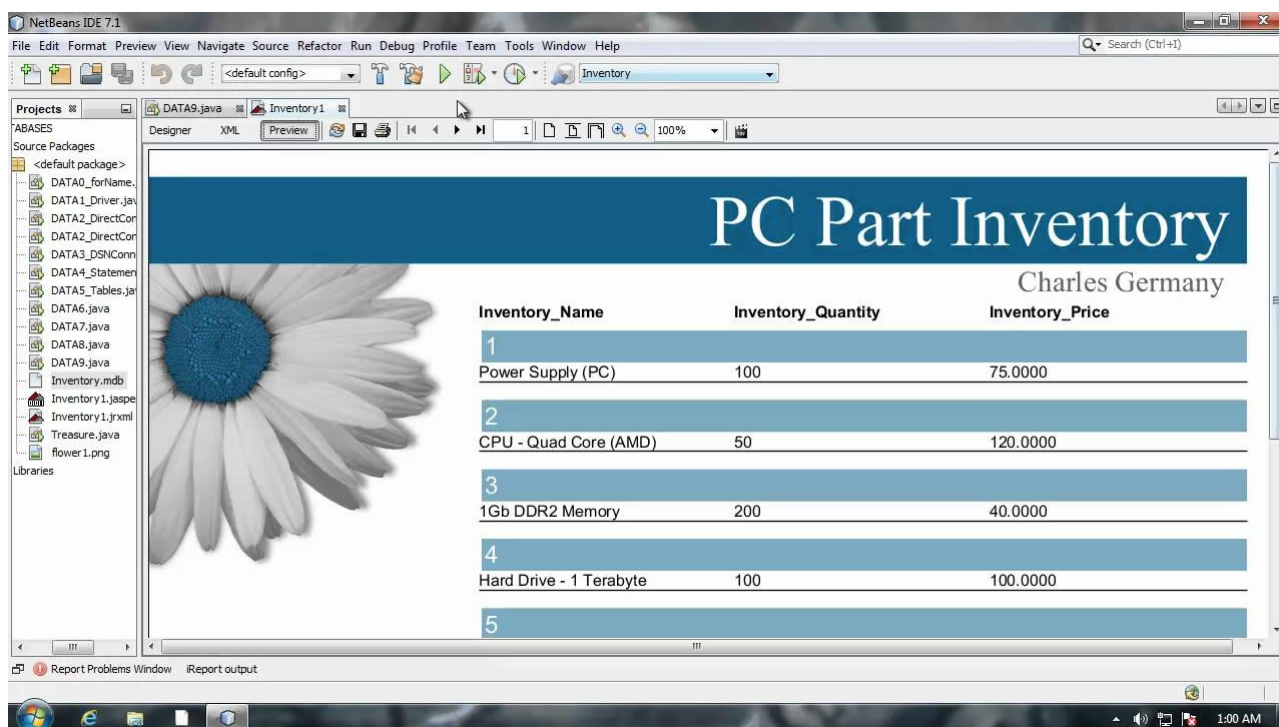


Figura 1: Jasper Report Entorno

### 2.2 jsPDF

jsPDF es una librería para Java Script que permite generar documentos PDF a partir de una plantilla HTML o directamente por programación. Esto nos permite poder diseñar un documento PDF de manera sencilla y adaptable. Una ventaja importante que se obtiene al utilizar esta librería es la de no utilizar recursos en el servidor ya que sólo se ejecuta del lado del cliente (navegador). [UniPython, 2020] La misma librería se proclama como "La solución de cliente HTML5 líder para generar archivos PDF. Perfecto para entradas de eventos, informes, certificados, ¡lo que sea!"



Figura 2: Java Script PDF

### 3 Desarrollo

Para la realización de esta actividad se realizó lo siguiente en ese orden:

- Modificar formulario de registro de usuarios
- Modificar el servlet para prepararse en tema de imágenes
- Modificar el jsp de lista de usuarios

#### 3.1 *Modificar form Usuario.jsp*

Se muestra la sección del formulario donde se adecua para tomar imágenes del lado del cliente.

```
1      <%--
2      Document    : usuarioForm
3      Created on  : Dec 20, 2020, 8:33:48 PM
4      Author     : aarongarcia
5  --%>
6
7  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
8  <%@page errorPage="error.jsp?de=usuarioForm.jsp" %>
9  <%@page contentType="text/html" pageEncoding="UTF-8"%>
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14 <title>P&acute;gina de formulario</title>
15 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
16 </head>
17 <body>
18 <div class="container">
19 <nav class="navbar navbar-expand-lg navbar-light bg-light">
20 <a class="navbar-brand" href="/">Project MVC</a>
21 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent">
22 <span class="navbar-toggler-icon"></span>
23 </button>
24
25 <div class="collapse navbar-collapse" id="navbarSupportedContent">
26 <ul class="navbar-nav mr-auto">
27 <li class="nav-item active">
28 <a class="nav-link" href="/">Inicio </a>
29 </li>
30 <li class="nav-item">
31 <a class="nav-link" href="UserServlet?accion=listaDeUsuarios">Lista de usuarios</a>
32 </li>
33 <li class="nav-item">
34 <a class="nav-link" href="UserServlet?accion=nuevo">Registrar usuario</a>
35 </li>
36 <li class="nav-item">
37 <a class="nav-link" href="UserServlet?accion=verpdf" target="blank">Ver PDF</a>
38 </li>
39 </ul>
40 </div>
41 </nav>
42
43 <div class="card-header bg-primary">
44 <h1>
45 Datos de usuario
46 </h1>
47 </div>
48
49 <div class="card-body">
```

```

50 <form
51     action=" UserServlet?accion=guardar"
52     method="POST"
53     name=" formUsuario"
54     enctype=" multipart/form-data"
55     id=" formUsuario">
56
57     <input
58         class=" form-control"
59         type=" hidden"
60         name=" id"
61         id=" id"
62         value="<c:out value=" ${dto.entidad.idUsuario}"/>" />
63
64     <div class=" form-group row">
65         <div class=" col-sm-2">
66             <label class=" col-form-label">Nombre:</label>
67         </div>
68
69         <div class=" col-sm-10">
70             <input
71                 id=" txtnombre"
72                 type=" text"
73                 name=" txtnombre"
74                 class=" form-control"
75                 required=" required"
76                 placeholder=" Nombre de usuario"
77                 value="<c:out value=" ${dto.entidad.nombre}"/>" />
78             </div>
79         </div>
80
81     <div class=" form-group row">
82         <div class=" col-sm-2">
83             <label class=" col-form-label">Apellido paterno:</label>
84         </div>
85
86         <div class=" col-sm-10">
87             <input
88                 type=" text"
89                 name=" txtpaterno"
90                 id=" txtpaterno"
91                 required=" required"
92                 placeholder=" Apellido paterno"
93                 class=" form-control"
94                 value="<c:out value=" ${dto.entidad.paterno}"/>" />
95             </div>
96         </div>
97
98     <div class=" form-group row">
99         <div class=" col-sm-2">
100             <label class=" col-form-label">Apellido materno:</label>
101         </div>
102
103         <div class=" col-sm-10">
104             <input
105                 type=" text"
106                 name=" txtmaterno"
107                 id=" txtmaterno"
108                 required=" required"
109                 placeholder=" Apellido materno"
110                 class=" form-control"
111                 value="<c:out value=" ${dto.entidad.materno}"/>" />
112             </div>
113         </div>
114
115     <div class=" form-group row">

```



```

116 <div class="col-sm-2">
117     <label class="col-form-label">Correo electr&oacute;nico:</label>
118 </div>
119
120 <div class="col-sm-10">
121     <input
122         type="email"
123         name="txtemail"
124         id="txtemail"
125         required="required"
126         placeholder="Correo electr&oacute;nico"
127         class="form-control"
128         value="<c:out value="{dto.entidad.email}"/>"/>
129     </div>
130 </div>
131
132 <div class="form-group row">
133     <div class="col-sm-2">
134         <label class="col-form-label">Nombre usuario:</label>
135     </div>
136
137     <div class="col-sm-10">
138         <input
139             type="text"
140             name="txtnombreUsuario"
141             id="txtnombreUsuario"
142             required="required"
143             placeholder="Username"
144             class="form-control"
145             value="<c:out value="{dto.entidad.nombreUsuario}"/>"/>
146         </div>
147     </div>
148
149 <div class="form-group row">
150     <div class="col-sm-2">
151         <label class="col-form-label">Clave de usuario:</label>
152     </div>
153
154     <div class="col-sm-10">
155         <input
156             type="password"
157             name="txtclaveUsuario"
158             id="txtclaveUsuario"
159             required="required"
160             placeholder="Password"
161             class="form-control"
162             value="<c:out value="{dto.entidad.claveUsuario}"/>"/>
163         </div>
164     </div>
165
166 <div class="form-group row">
167     <div class="col-sm-2">
168         <label class="col-form-label">Tipo de usuario:</label>
169     </div>
170
171     <div class="col-sm-10">
172         <input
173             type="text"
174             name="txttipoUsuario"
175             id="txttipoUsuario"
176             required="required"
177             placeholder="Rol de usuario"
178             class="form-control"
179             value="<c:out value="{dto.entidad.tipoUsuario}"/>"/>
180         </div>
181     </div>

```

```

182
183
184 <div class="form-group row">
185 <label for="imagen" class="col-sm-2 col-form-label">Imagen:</label>
186 <div class="col-sm-10">
187 <input
188 type="file"
189 class="form-control-file"
190 id="imagen"
191 name="imagen"
192 accept="image/png, image/jpeg"
193 value="<c:out value="{dto.entidad.imagen}"/>" />
194 </div>
195 </div>
196
197
198
199
200 <div class="col-sm-10">
201 <input
202 type="hidden"
203 name="txtimagen"
204 id="txtimagen"
205 required="required"
206 placeholder="Imagen de perfil"
207 class="form-control"
208 value="<c:out value="{dto.entidad.imagen}"/>" />
209 </div>
210
211 <button type="submit" class="form-group row offset-md-0 btn btn-outline-primary">
212 Guardar
213 </button>
214 </form>
215 </div>
216 <br>
217 <a href="javascript:history.back()" class="btn btn-outline-dark btn-sm" role="button">
218 </div>
219
220
221
222
223 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
224 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js">
225 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
226 </body>
227 </html>

```

---

**Algoritmo:** Únicamente se estableció como atributo dentro de la tabla "enctype="multipart/form-data", el cual nos sirve para reconocer que vamos a enviar archivos, en este caso imágenes. Igualmente se agrega el input de imagen donde se aceptan los formatos jpg y png, el campo de texto de imagen se hace no visible, para enviar algo siempre como definición del modelo Usuario.

### 3.2 *Modificar UserServlet.java — crearUsuario*

Se muestra la sección del servlet de usuario donde se adecua para tomar valores de la respuesta del formulario para registrar el nombre de la imagen en el atributo imagen del usuario y almacenar la imagen en el servidor.

```
1      private void guardarUsuario(HttpServletRequest request, HttpServletResponse response)
2      {
3          UsuarioDAO dao = new UsuarioDAO();
4          UsuarioDTO dto = new UsuarioDTO();
5
6          // Atributos en com n
7          dto.getEntidad().setNombre(request.getParameter("txtnombre"));
8          dto.getEntidad().setPaterno(request.getParameter("txtpaterno"));
9          dto.getEntidad().setMaterno(request.getParameter("txtmaterno"));
10         dto.getEntidad().setEmail(request.getParameter("txtemail"));
11         dto.getEntidad().setNombreUsuario(request.getParameter("txtnombreUsuario"));
12         dto.getEntidad().setClaveUsuario(request.getParameter("txtclaveUsuario"));
13         dto.getEntidad().setTipoUsuario(request.getParameter("txttipoUsuario"));
14         dto.getEntidad().setImagen(request.getParameter("txtimagen"));
15
16         try {
17             Part archivo = request.getPart("imagen"); //llamada al par metro foto de mi formulario
18
19             String path_carpeta = request.getServletContext().getRealPath("imagenes_usuario"); //i
20             String imagen_nombre = Paths.get(archivo.getSubmittedFileName()).getFileName().toStrin
21
22             dto.getEntidad().setImagen(imagen_nombre);
23             archivo.write(path_carpeta + File.separator + imagen_nombre);
24         } catch (IOException | ServletException ex) {
25             Logger.getLogger(UserServlet.class.getName()).log(Level.SEVERE, null, ex);
26         }
27
28         try {
29             if (request.getParameter("id") == null || request.getParameter("id").isEmpty()) { // I
30                 dao.create(dto);
31                 listarUsuarios(request, response);
32             } else { // actualizacion
33                 dto.getEntidad().setIdUsuario(Integer.parseInt(request.getParameter("id")));
34                 dao.update(dto);
35                 listarUsuarios(request, response);
36             }
37         } catch (SQLException ex) {
38             Logger.getLogger(CategoriaServlet.class.getName()).log(Level.SEVERE, null, ex);
39         }
40     }
```

**Algoritmo:** Dentro del método guardarUsuario se obtiene la imagen por el identificador establecido en el formulario de registro, se extrae el nombre de la imagen y el contenido de la imagen como tal, de ahí se procede a únicamente almacenar el nombre de la imagen e escribir la imagen con base a un path establecido, en este caso en la carpeta "imagenesusuario".

### 3.3 *Modificar listaUsuario.jsp*

Se muestra la sección de listar usuarios, en donde se hace la adecuación para mostrar imágenes con cierto tamaño y path.

```
1      <%--
2      Document    : listaUsuarios
3      Created on  : Dec 20, 2020, 2:33:59 PM
4      Author     : aarongarcia
5  --%>
6  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
7  <%@page errorPage="error.jsp?de=listaUsuarios.jsp" %>
8  <%@page contentType="text/html" pageEncoding="UTF-8"%>
9  <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <title>Listado de usuarios</title>
14 <link href='https://unpkg.com/boxicons@2.0.7/css/boxicons.min.css' rel='stylesheet '>
15 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap"
16 </head>
17 <body>
18 <div class="container">
19 <nav class="navbar navbar-expand-lg navbar-light bg-light">
20 <a class="navbar-brand" href="/">Proyecto MVC</a>
21 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#
22 <span class="navbar-toggler-icon"></span>
23 </button>
24
25 <div class="collapse navbar-collapse" id="navbarSupportedContent">
26 <ul class="navbar-nav mr-auto">
27 <li class="nav-item active">
28 <a class="nav-link" href="/">Inicio </a>
29 </li>
30 <li class="nav-item">
31 <a class="nav-link" href="UserServlet?accion=listaDeUsuarios">Lista de
32 </li>
33 <li class="nav-item">
34 <a class="nav-link" href="UserServlet?accion=nuevo">Registrar usuario
35 </li>
36 <li class="nav-item">
37 <a class="nav-link" href="UserServlet?accion=verpdf" target="blank">V
38 </li>
39 </ul>
40 </div>
41 </nav>
42 <table class="table table-sm table-bordered table-hover ">
43 <thead class="thead-dark">
44 <tr>
45 <th scope="col" class="text-center">ID</th>
46 <th scope="col" class="text-center">Nombre</th>
47 <th scope="col" class="text-center">Paterno</th>
48 <th scope="col" class="text-center">Materno</th>
49 <th scope="col" class="text-center">Email</th>
50 <th scope="col" class="text-center">Usuario</th>
51 <th scope="col" class="text-center">Clave</th>
52 <th scope="col" class="text-center">Tipo</th>
53 <th scope="col" class="text-center">Imagen</th>
54 <th scope="col" class="text-center" colspan="3">Acci&oacute;n</th>
55 </tr>
56 </thead>
57 <tbody>
58 <c:forEach
59 var="usuarios"
60 items="{listaDeUsuarios}">
61
```

```

62 <tr>
63 <td class="text-center">
64 <a class="btn"
65 href="UserServlet?accion=ver&id=<c:out value="\${usuarios.entidad
66 <c:out value="\${usuarios.entidad.idUsuario}"/>"/>
67 </a>
68 </td>
69 <td>
70 <c:out value="\${usuarios.entidad.nombre}"/> <!-- -->
71 </td>
72 <td>
73 <c:out value="\${usuarios.entidad.paterno}"/> <!-- -->
74 </td>
75 <td>
76 <c:out value="\${usuarios.entidad.materno}"/> <!-- -->
77 </td>
78 <td>
79 <c:out value="\${usuarios.entidad.email}"/> <!-- -->
80 </td>
81 <td>
82 <c:out value="\${usuarios.entidad.nombreUsuario}"/> <!-- -->
83 </td>
84 <td>
85 <c:out value="\${usuarios.entidad.claveUsuario}"/> <!-- -->
86 </td>
87 <td>
88 <c:out value="\${usuarios.entidad.tipoUsuario}"/> <!-- -->
89 </td>
90 <td>
91 
93 class="img-rounded"
94 width="80"
95 height="80">
96 </td>
97 <td class="text-center">
98 <a class="btn btn-danger btn-xs"
99 href="UserServlet?accion=eliminar&id=<c:out value="\${usuarios.entidad.id}"/>"/>
100 <i class='bx bx-trash'></i>
101 </a>
102 </td>
103 <td class="text-center">
104 <a class="btn btn-warning btn-xs"
105 href="UserServlet?accion=actualizar&id=<c:out value="\${usuarios.entidad.id}"/>"/>
106 <i class='bx bx-pencil'></i>
107 </a>
108 </td>
109 <td class="text-center">
110 <a class="btn btn-info btn-xs"
111 href="UserServlet?accion=reporteUsuario&id=<c:out value="\${usuarios.entidad.id}"/>"/>
112 <i class='bx bxs-report'></i>
113 </a>
114 </td>
115 </td>
116 </tr>
117 </c:forEach>
118 </tbody>
119 </table>
120 </div>
121
122
123 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
124 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
125 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
126 </body>
127 </html>

```

*Algoritmo:* Dentro del jsp de listar usuarios, el campo de texto se cambia a una imagen, en la cual establecemos el largo y ancho en que se mostrara, en este caso de 80x80 pixeles, el path y la imagen.

## 4 Resultados

Este ejercicio estuvo bastante sencillo, si tuve un tropezón, el cual fue intentar querer subir imágenes a heroku esperando que fuera igual que en el servidor local, lo que resulta es que heroku no almacena imágenes, entonces por mas que trate de hacerlo, no pude, por lo que al final opte por realizarlo de manera local.

### 4.1 Ejecución del Programa

#### 4.1.1 Capturas de funcionamiento

The screenshot shows a web application interface for user registration. At the top, there is a navigation bar with links: "Project MVC", "Inicio", "Lista de usuarios", "Registrar usuario", and "Ver pdf". Below this is a blue header section with the title "Datos de usuario". The form consists of several input fields with labels on the left: "Nombre:" (input: "Nombre de usuario"), "Apellido paterno:" (input: "Apellido paterno"), "Apellido materno:" (input: "Apellido materno"), "Correo electrónico:" (input: "Correo electrónico"), "Nombre usuario:" (input: "Username"), "Clave de usuario:" (input: "Password"), and "Tipo de usuario:" (input: "Rol de usuario"). Below these is a file upload section for "Imagen:" with a "Choose File" button and the text "No file chosen". At the bottom of the form is a large button labeled "Guardar". Below the form is a "Regresar" button.

Figura 3: Formulario de registro de usuario

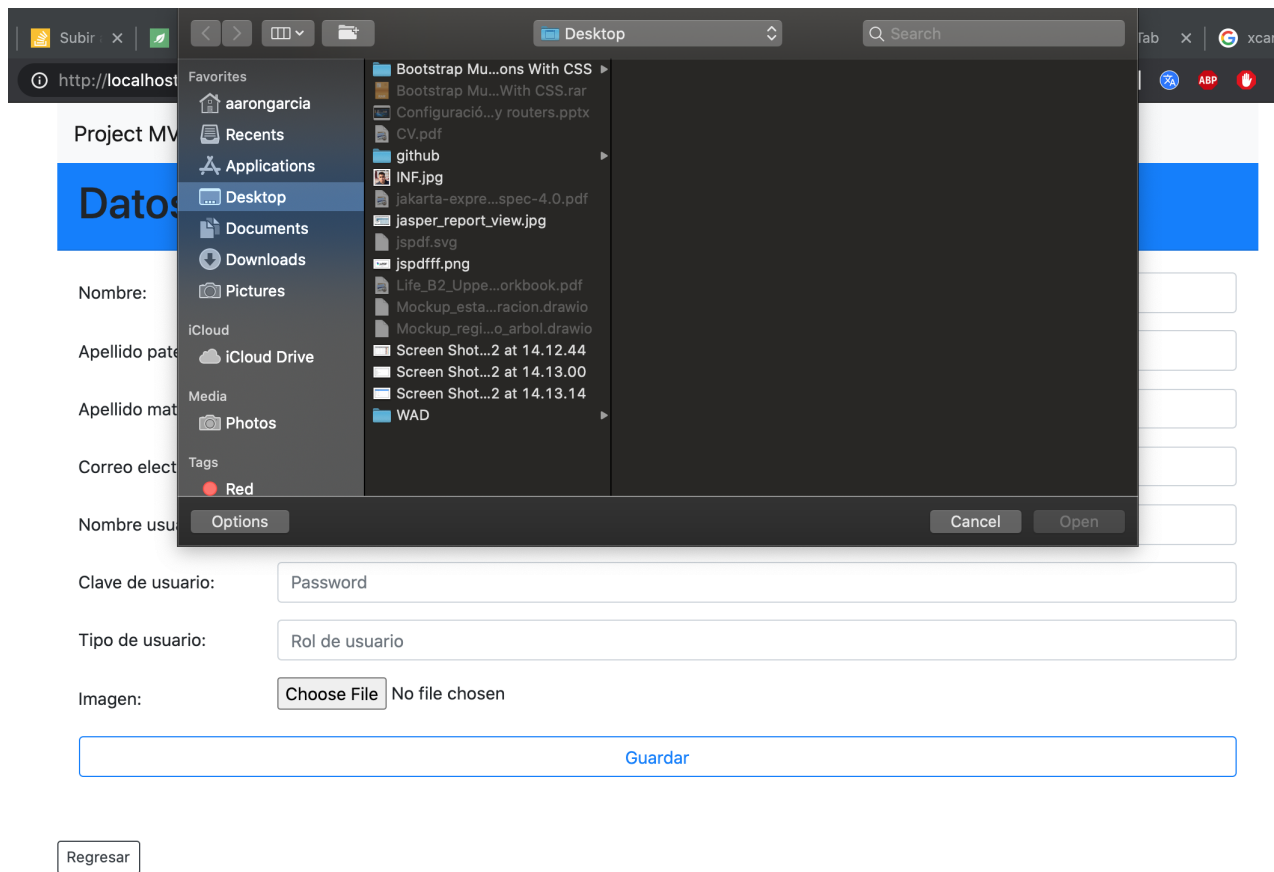


Figura 4: Selección de imagen



Proyecto MVC

Inicio

Lista de usuarios

Registrar usuario

Ver pdf





























| ID | Nombre         | Paterno   | Materno   | Email                           | Usuario     | Clave       | Tipo          | Imagen  | Acción  |   |   |
|----|----------------|-----------|-----------|---------------------------------|-------------|-------------|---------------|---|---|---|---|
| 1  | Aaron Antonio  | Garcia    | Gonzalez  | aarongarcia.ipn.escom@gmail.com | AaronGG11   | AaronGG11   | administrador |  |  |  |  |
| 2  | Gerardo        | Gallegos  | Garcia    | geragg@gbrtelecom.com           | GerardoGall | GerardoGall | Empleado      |  |  |  |  |
| 3  | Luisa          | Maldonado | Flor      | MariLu@cocina.mx                | LuisaMal    | LuisaMal    | empleado      |  |  |  |  |
| 4  | Josue          | Gallegos  | Corona    | josCor@apps.com.mx              | JosCor      | JosCor      | empleado      |  |  |  |  |
| 5  | Lucia          | Del mar   | Salgado   | LuciMar@outlook.com             | MissLuci    | MissLuci    | empleado      |  |  |  |  |
| 6  | Miriam         | Velazquez | Corona    | MiriamVelazquez98@gmail.com     | MiriamVC    | MiriamVC    | empleado      |  |  |  |  |
| 7  | Rodolfo Ramses | Rivera    | Hernandez | RodoRH@xcaret.com.mx            | RodoRH      | RodoRH      | administrador |  |  |  |  |

Figura 5: Lista de usuarios

|   |
|---|
| Proyecto MVC Inicio Lista de usuarios Registrar usuario Ver pdf |
|---|

Datos de usuario

Id: 1

Nombre: Aaron Antonio

Apellido Paterno: Garcia

Apellido Materno: Gonzalez

Correo electrónico: aarongarcia.ipn.escom@gmail.com

Nombre de usuario: AaronGG11

Clave de usuario: AaronGG11

Tipo de usuario: administrador

Imagen: INF.jpg

Regresar

Figura 6: Información de usuarios

## 5 Conclusión

Fue un ejercicio sencillo, jamás había utilizado JasperReports, y el pasar argumentos como parámetros fue muy fácil, pero comenzaba a complicarme la vida con base a un tutorial que vi en youtube, lo mejor fue revisar la documentación y todo salio bien, igualmente prefiero utilizar reportes en blanco para después personalizar cada uno.

*Aarón Antonio García González*

## 6 Referencias Bibliográficas

### Referencias

[Wikipedia, 2020] Wikipedia. *JasperReports* Wikipedia.

[DefinicionDe, 2020] DefinicionDe. *Definición de reporte* DefinicionDe.

[UniPython, 2020] UniPython. *Crear un PDF en Java Script* UniPython.