

	<p style="text-align: center;">UNIVERSIDAD DON BOSCO ESCUELA DE COMPUTACION</p>
<p style="text-align: center;">CICLO I</p>	<p style="text-align: center;">PROGRAMACIÓN ORIENTADA A OBJETOS CREACIÓN DE REPORTES EN JAVA CON JASPER REPORTS</p>

I.OBJETIVOS

Que el estudiante:

- Genere reportes con parámetros y sin parámetros en aplicaciones web utilizando Jasper Reports.

II. INTRODUCCIÓN

JasperReports es un framework para desarrollar reportes tanto web como desktop en Java.

Aunque el formato fuente de un reporte en JasperReports es un archivo XML, existe una herramienta que permite crear un reporte de manera visual, su nombre es JasperSoft Studio.

La herramienta JasperSoft Studio es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con imágenes, subinformes, etc. JasperSoft Studio está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, etc.

Características de JasperSoft Studio

La lista siguiente describe algunas de las características importantes de JasperSoft Studio:

- 100% escrito en Java, es Open Source y gratuito.
- Maneja el 98% de las etiquetas de JasperReports.
- Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports (subreportes).
- Soporta internacionalización nativamente.
- Navegador de la estructura del documento.
- Recopilador y exportador integrados.
- Soporta JDBC.
- Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
- Incluye Wizard's (asistentes) para crear automáticamente informes.
- Tiene asistentes para generar los subreportes.
- Tiene asistentes para las plantillas.

- Facilidad de instalación.

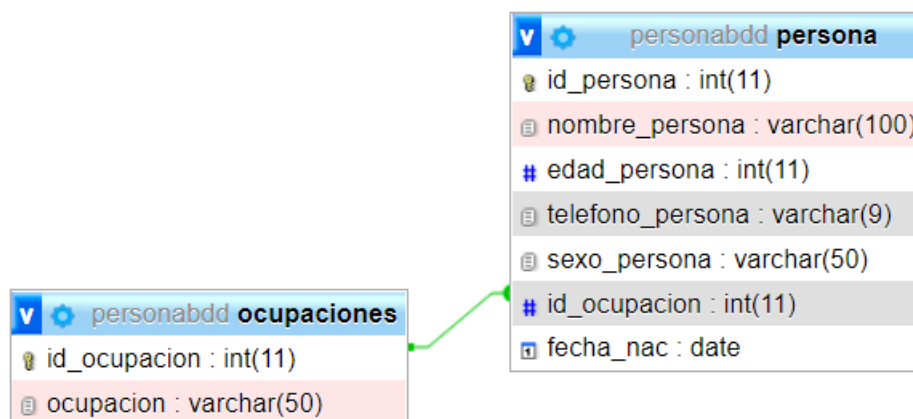
III. PROCEDIMIENTO

En esta guía se mostrará cómo generar reportes para una aplicación web hecha con Java Server Pages, utilizando como gestor de base de datos a MySQL. Para empezar, estableceremos nuestro entorno de trabajo (se indica la versión mínima de cada herramienta con la que debe disponer):

- IntelliJ IDEA 2020.3.4.
- Tomcat 10.0 o superior que será nuestro servidor web.
- Aplicación para leer PDF.
- Mysql 5.0 o superior (en cargado de alojar la base de datos).
- JasperSoft Studio y JasperReports: el primero será el entorno gráfico que nos ayudará a crear el reporte, mientras que del segundo utilizaremos las librerías que servirán para compilar y ejecutar el reporte.

Base de datos

Antes de iniciar el diseño y la creación del reporte es importante tener una base de datos lista para crear el reporte, para lo cual usaremos la base de datos proporcionada en la guía #4 (**personabdd**).



A cada tabla deberá introducir valores los cuales quedaran de la siguiente manera

id_persona	nombre_persona	edad_persona	telefono_persona	sexo_persona	id_ocupacion	fecha_nac	id_ocupacion	ocupacion
1	Alejandro Pineda Saravia	45	7722-4455	Masculino	1	1976-02-05	1	Doctor
2	Fernando Calderón	21	7667-7890	Masculino	2	2001-05-07	2	Emprendedor
3	Juan Machado	25	7724-5678	Masculino	3	1997-05-11	3	Profesor
4	Alejandra Saravia	28	7724-0678	Femenino	2	1994-03-08		
5	Jocelyn Martinez	21	7894-1239	Femenino	1	2000-07-01		
6	Pamela Quinteros	22	6689-8912	Femenino	3	1999-01-05		

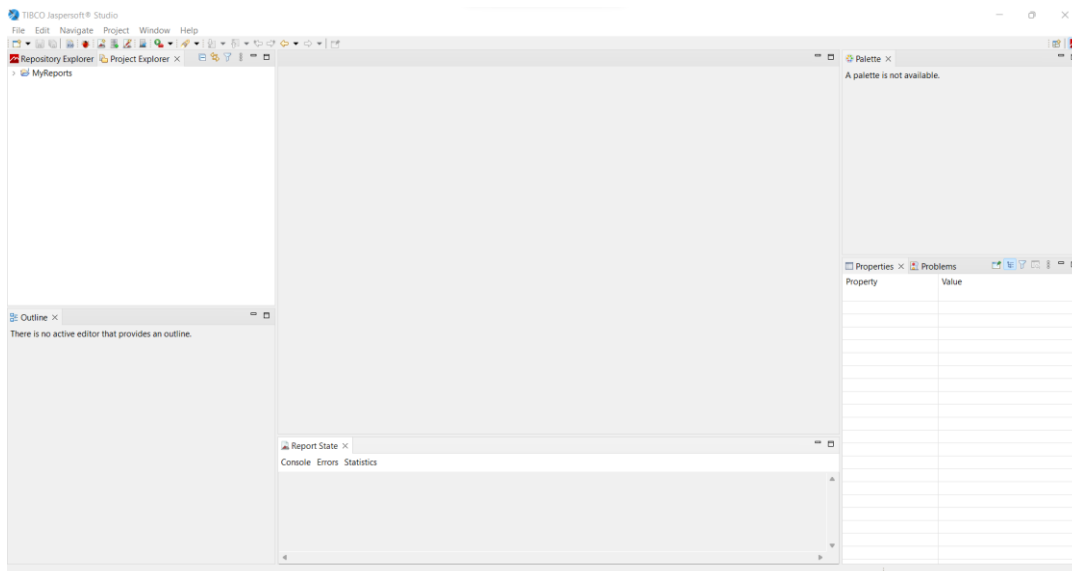
JasperSoft Studio

El siguiente paso consiste en crear reportes a partir de la base de datos persona, para ello deberá ejecutar el programa **JasperSoft Studio** proporcionado por el docente y con el cual se van a realizar los pasos de esta guía.

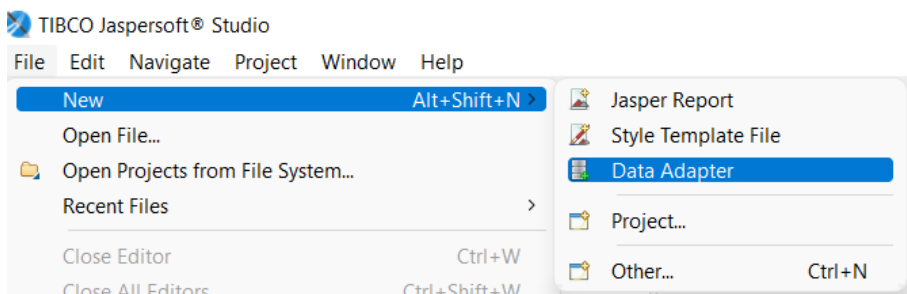
1. Instalar el software dentro de su computadora y proceder a su ejecución.



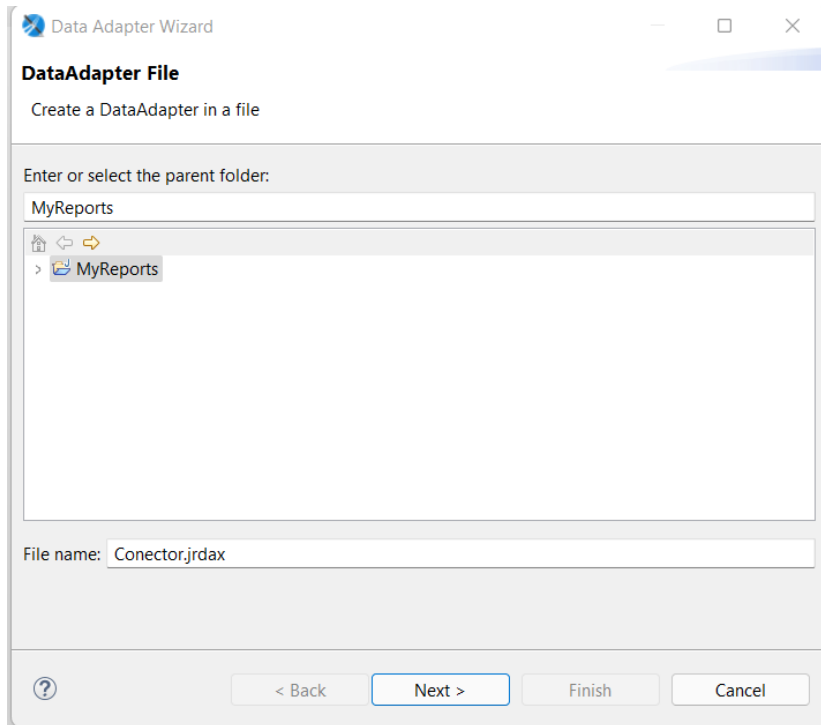
2. Al terminar deberá mostrar una ventana como la siguiente.



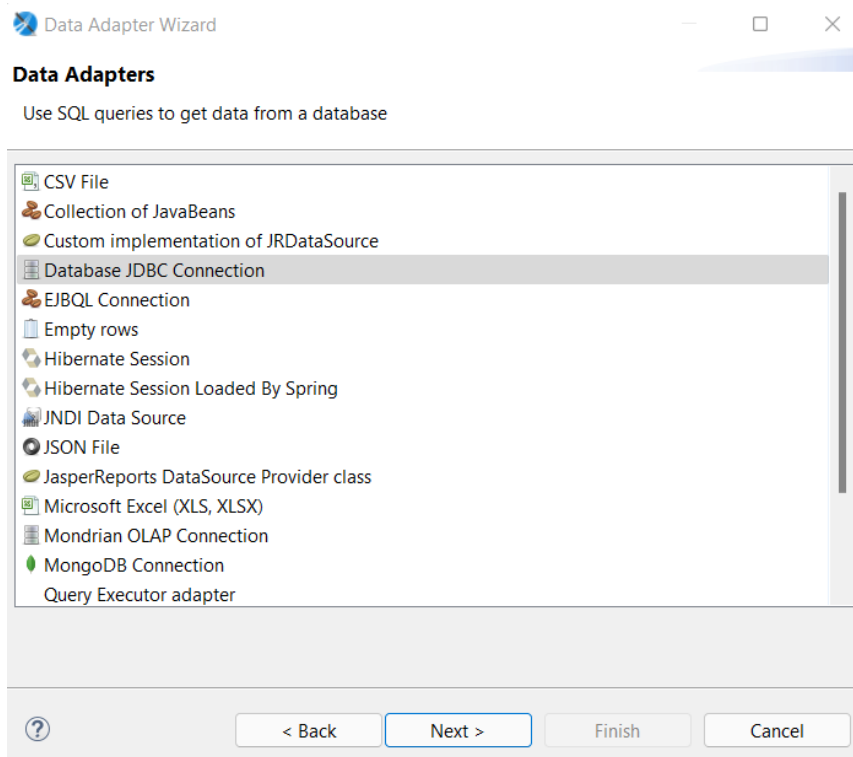
3. Una vez estando en la ventana principal de JasperSoft Studio deberemos establecer una conexión con la base de datos, para realizar este paso dar clic sobre file>new>Data Adapter.



4. Ahora deberá dar un nombre a este componente, preferiblemente **Conector.jrdax**.



5. Luego seleccionará un adaptador de datos, para este caso será **DataBase JDBC Connection**.



6. Deberá llenar los campos como se presentan a continuación.

The screenshot shows the 'Data Adapter Wizard' window, specifically the 'Database JDBC Connection' step. The 'Name' field is set to 'Adaptador 1'. The 'JDBC Driver' dropdown is set to 'MySQL (com.mysql.jdbc.Driver)'. The 'JDBC Url' is 'jdbc:mysql://localhost/personabdd'. The 'Username' is 'root' and the 'Password' field is empty. An 'Attention!' message states: 'Passwords are saved in clear text'. At the bottom, there are tabs for 'Database Location', 'Connection Properties', and 'Driver Classpath'. The 'Finish' button is highlighted.

Data Adapter Wizard
Database JDBC Connection

Name: Adaptador 1

JDBC Driver: MySQL (com.mysql.jdbc.Driver)

JDBC Url: jdbc:mysql://localhost/personabdd

Username: root

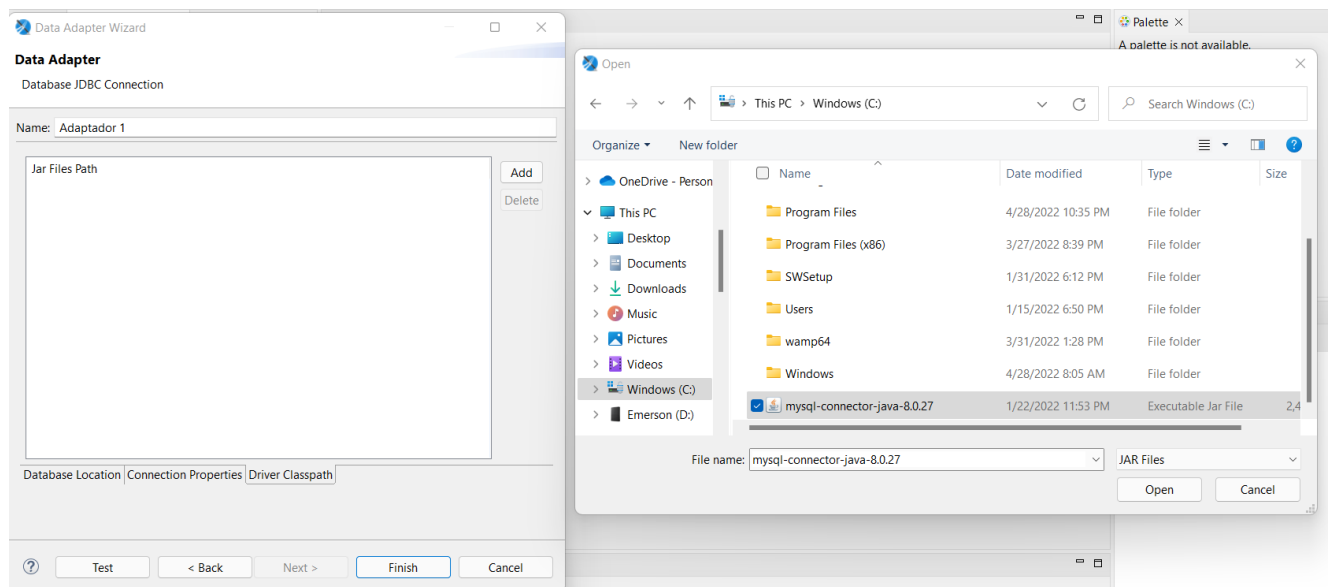
Password:

Attention! Passwords are saved in clear text

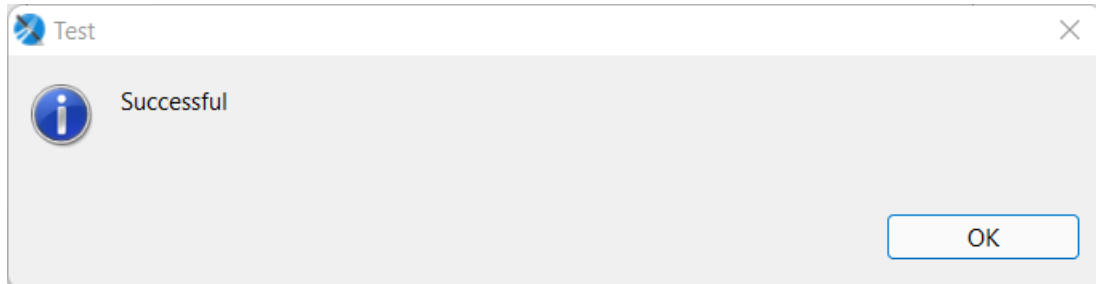
Database Location | Connection Properties | Driver Classpath

Test < Back Next > Finish Cancel

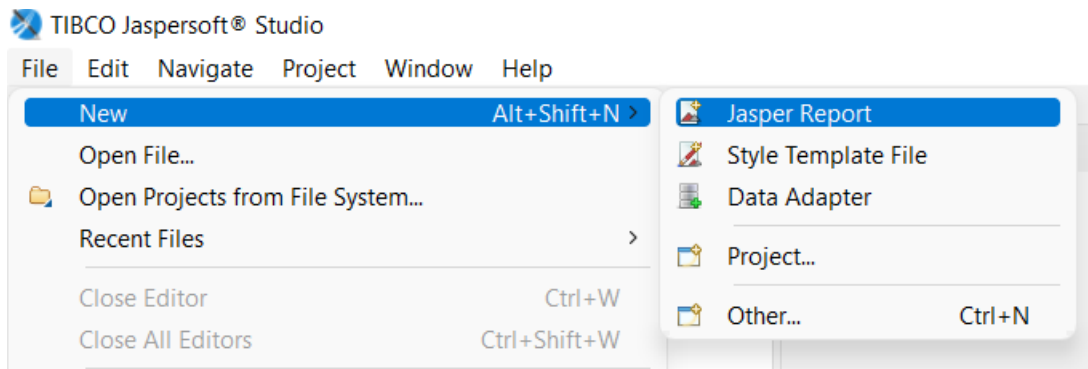
7. En la misma ventana ir a DriverClasspath y seleccionar **Add** y buscar el driver de mysql usado en las primeras prácticas de la asignatura.



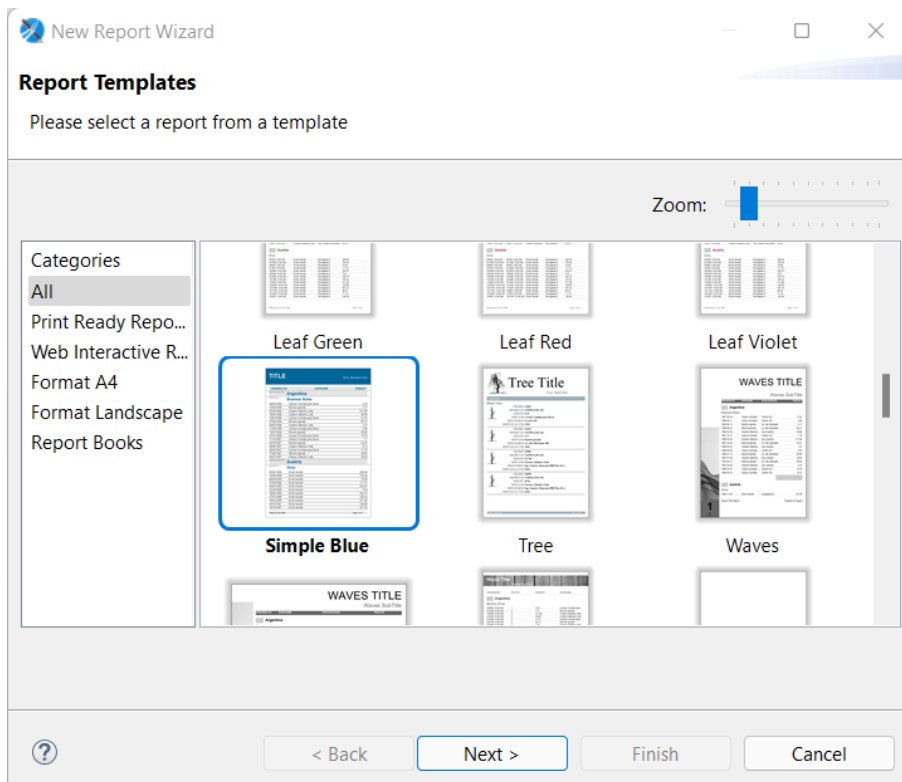
8. Dar click en test y verificar que la conexión sea exitosa, finalmente dar click en **Finish**.



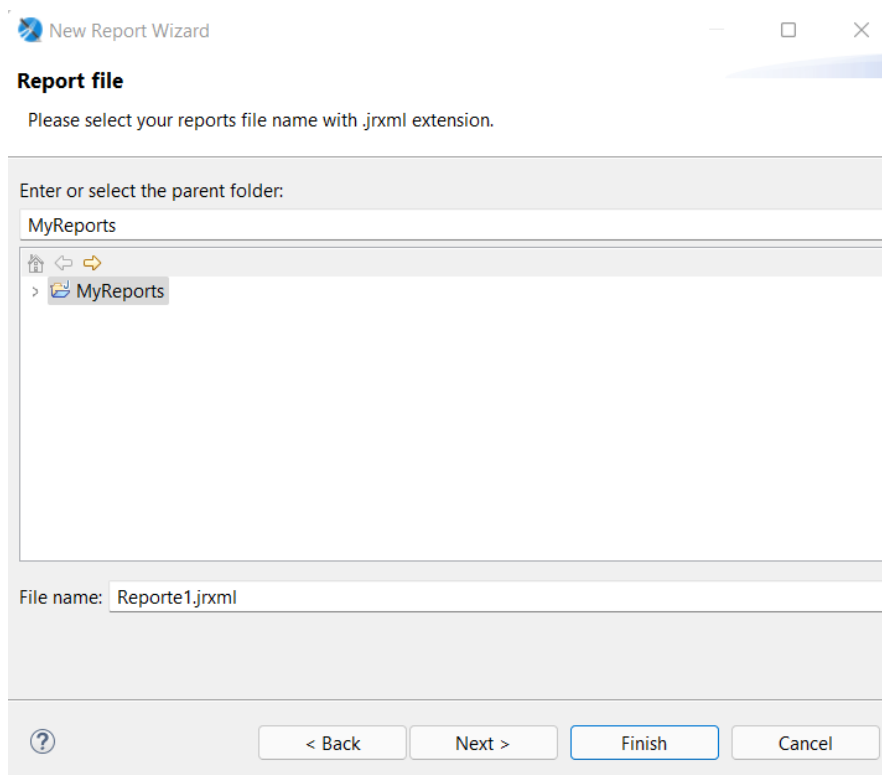
9. Ahora crearemos un nuevo reporte, para lo cual deberá ir a file>new>Jasper Report



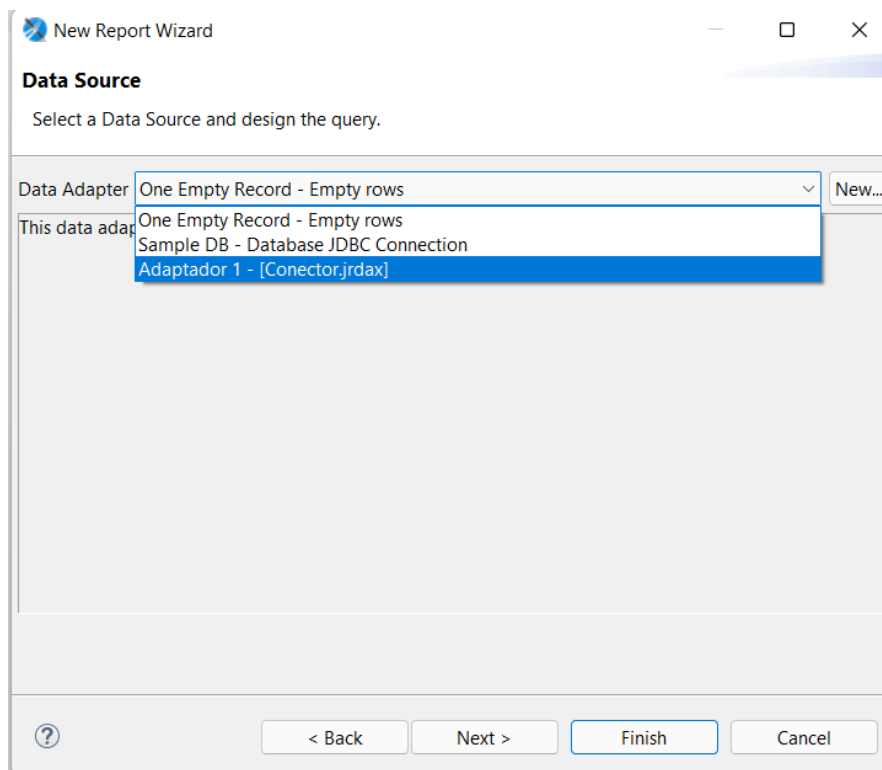
10. Deberá escoger una plantilla para dar estilo a su reporte.



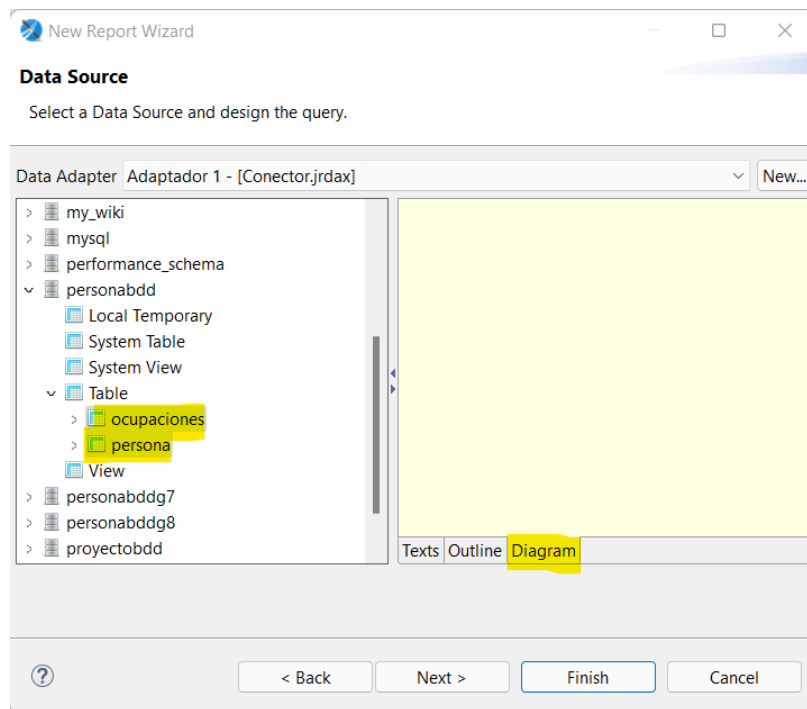
11. Asignaremos un nombre a nuestro reporte.



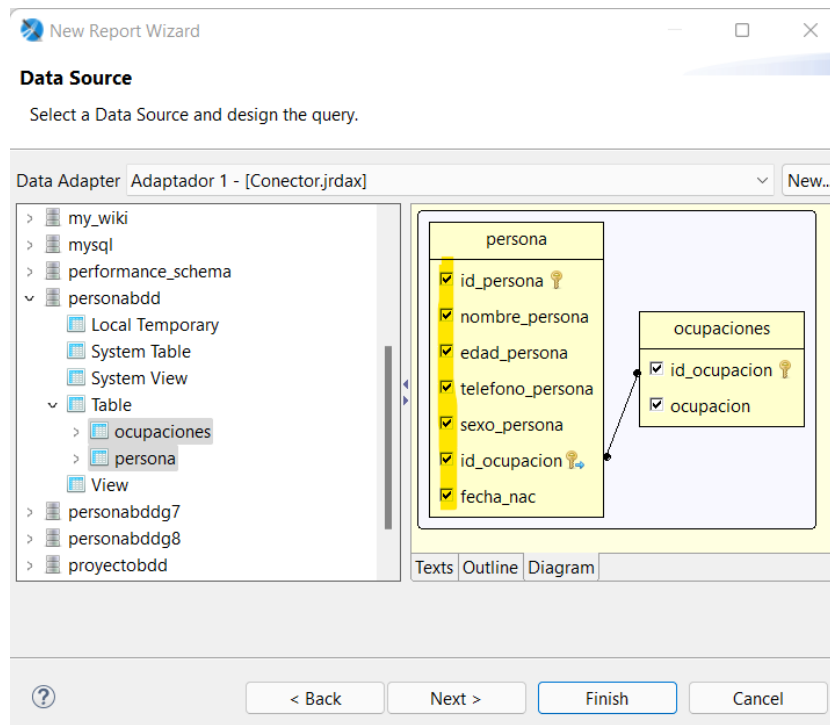
12. Seleccionaremos el adaptador creado en los pasos anteriores.



13. Inmediatamente podrá ver todas las bases de datos alojadas en su servicio de mysql, para este caso recordar que estaremos trabajando con **personabdd**. Ir la pestaña **diagram** y seleccionar las tablas de las cuales desea obtener su información.



14. Es importante chequear los campos que desea tomar en cuenta en su reporte. Para este ejemplo los tomaremos todos.



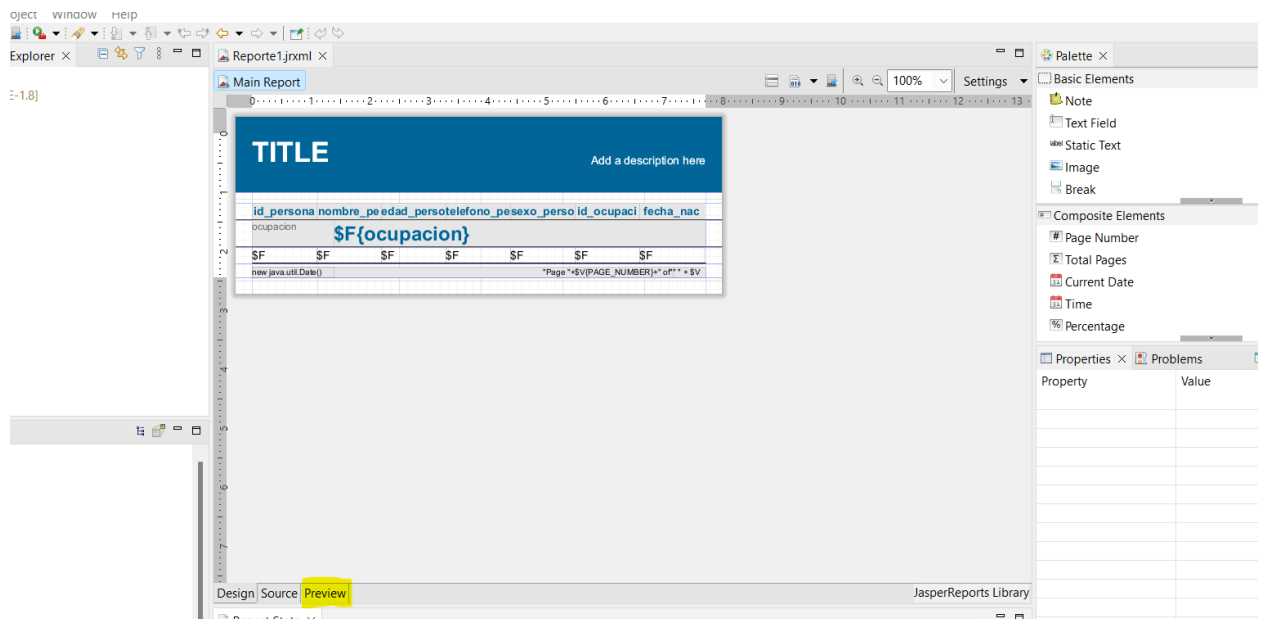
15. Aquí podrá arrastrar todos los campos que serán impresos en el reporte, podrá usar las flechas (<, >, <<, >>) para mover los campos de un lado al otro. Finalmente, su selección debe quedar así.

The screenshot shows the 'New Report Wizard' dialog box, specifically the 'Fields' step. The title bar reads 'New Report Wizard'. Below the title, the text 'Please select dataset fields' is displayed. The dialog is divided into two main sections. On the left, under the heading 'Dataset Fields', there is a list containing 'COLUMN_8'. On the right, under the heading 'Fields', there is a list containing 'id_persona', 'nombre_persona', 'edad_persona', 'telefono_persona', 'sexo_persona', 'id_ocupacion', 'fecha_nac', and 'ocupacion'. Between these two lists are four arrow buttons: '>', '>>', '<', and '<<'. To the right of the 'Fields' list are two buttons: 'Up' and 'Down'. At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), and 'Cancel'. A help icon (?) is located at the bottom left.

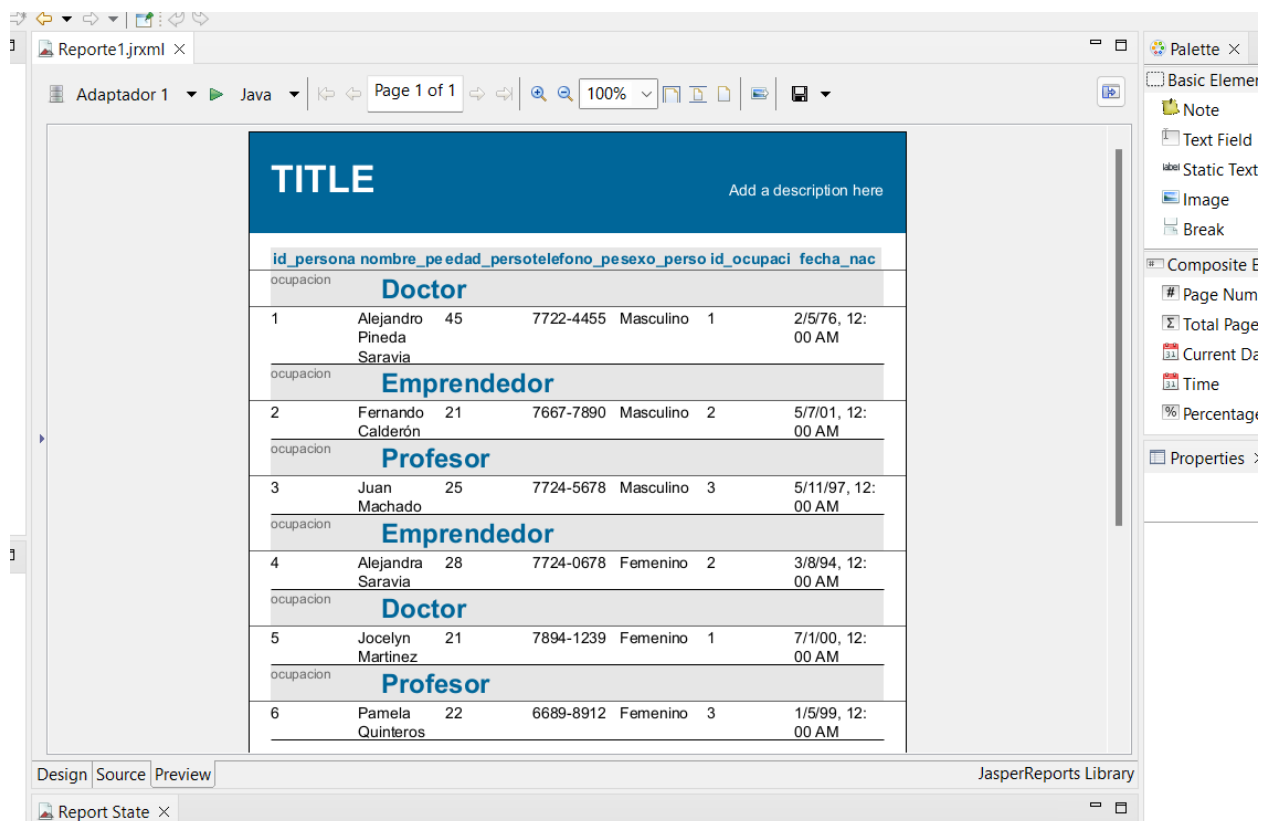
16. En este paso decidiremos en que orden se agruparan los elementos, esto es como hacer una consulta **group by** en mysql. Para este caso agruparemos por el campo ocupación. Dar click en **finish**.

The screenshot shows the 'New Report Wizard' dialog box, specifically the 'Group By' step. The title bar reads 'New Report Wizard'. Below the title, the text 'Please select fields to Group By' is displayed. The dialog is divided into two main sections. On the left, under the heading 'Dataset Fields', there is a list containing 'id_persona', 'nombre_persona', 'edad_persona', 'telefono_persona', 'sexo_persona', 'id_ocupacion', and 'fecha_nac'. On the right, under the heading 'Fields', there is a list containing 'ocupacion'. Between these two lists are four arrow buttons: '>', '>>', '<', and '<<'. To the right of the 'Fields' list are two buttons: 'Up' and 'Down'. Below the 'Dataset Fields' list, there is a checkbox labeled 'Use the group fields as sort fields. Select this option if you want to aggregate all the group fields with the same value and not only the consecutive ones'. At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), and 'Cancel'. A help icon (?) is located at the bottom left.

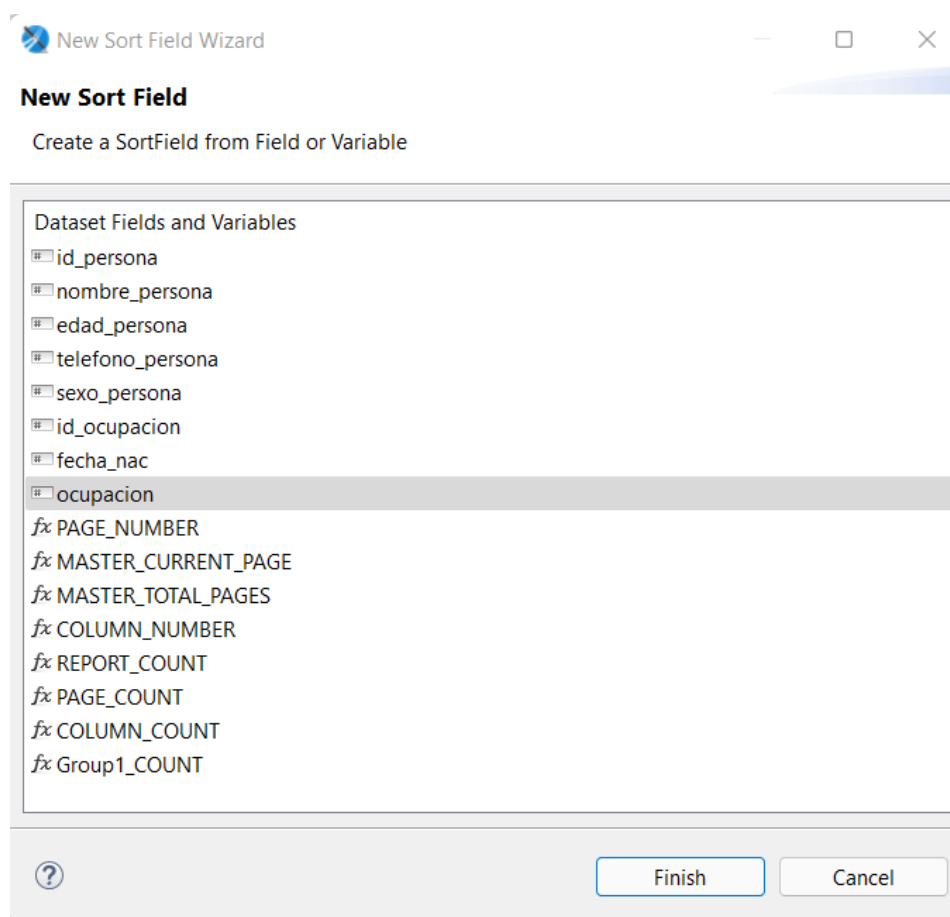
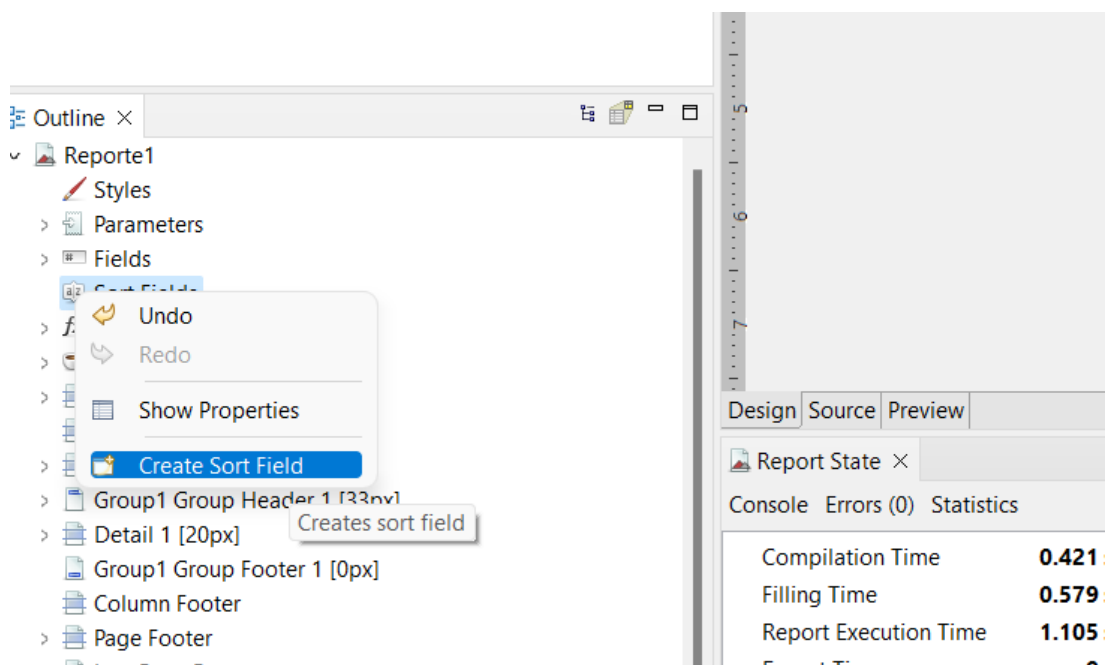
17. Su reporte fue creado con éxito, para visualizar bastara dar click en preview.



18. Pueda que los campos aun no se vean correctamente agrupados, esto es porque aún falta agregar un sort, podrá ver en la siguiente imagen la impresión.



19. En la barra de propiedades se encuentra la opción sort fields, dar click derecho y agregaremos un nuevo sort. El cual lo haremos por el campo ocupación.

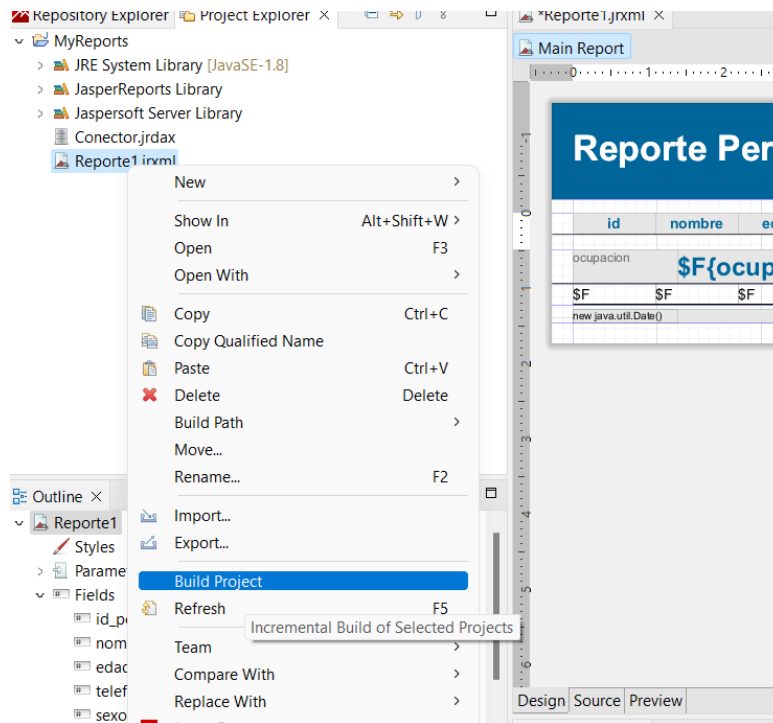


20. Dar click en preview y ver el cambio.

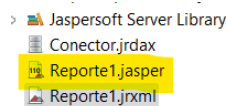
id_persona	nombre_persona	edad_persona	telefono_persona	sexo_persona	id_ocupacion	fecha_nacimiento
Doctor						
1	Alejandro Pineda Saravia	45	7722-4455	Masculino	1	2/5/76, 12:00 AM
5	Jocelyn Martinez	21	7894-1239	Femenino	1	7/1/00, 12:00 AM
Emprendedor						
2	Fernando Calderón	21	7667-7890	Masculino	2	5/7/01, 12:00 AM
4	Alejandra Saravia	28	7724-0678	Femenino	2	3/8/94, 12:00 AM
Profesor						
3	Juan Machado	25	7724-5678	Masculino	3	5/11/97, 12:00 AM
6	Pamela Quinteros	22	6689-8912	Femenino	3	1/5/99, 12:00 AM

Para esta parte como puede observar el reporte fue creado, pero las columnas no poseen valores descriptivos. Volver a la pestaña **Designer** cambiar los nombres a las columnas y ponerlo un título que haga referencia al resultado mostrando en el reporte.

21. Finalmente crearemos el archivo .jasper el cual nos servirá para generar el reporte de forma dinámica en la jsp. Para ello daremos click derecho sobre el archivo .jspx y **build project**.



22. Este será el archivo que nos llevaremos a la base de nuestro proyecto, es decir dentro de la carpeta **webapp**. Para ello bastara con dar click derecho, copiar y pegar dentro de nuestro directorio en el ide o buscando webapp desde los directorios de Windows.



Llamando al reporte desde una página JSP

Para esta parte deberemos crear un proyecto web llamado **JasperReport**.

1. Agregar las librerías necesarias proporcionadas por el docente (ver la siguiente figura).
Recordar que esto se hace mediante la herramienta pom.xml.

```
<!-- https://mvnrepository.com/artifact/org.glassfish.web/jakarta.servlet.jsp.jstl -->

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

    <groupId>mysql</groupId>

    <artifactId>mysql-connector-java</artifactId>
```

```

    <version>8.0.27</version>

</dependency>

<!-- https://mvnrepository.com/artifact/net.sf.jasperreports/jasperreports -->

<dependency>

    <groupId>net.sf.jasperreports</groupId>

    <artifactId>jasperreports</artifactId>

    <version>6.19.1</version>

</dependency>

```

2. En la página **index.jsp** digitar el siguiente código.

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

<!DOCTYPE html>

<html>

<head>

    <title>JasperReport</title>

    <link          href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"                                     integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

    <link          rel="stylesheet"                    href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

</head>

<body >

<section class="container">

    <h1 class="text-center">Mi Primer Reporte con JasperReport</h1>

<a href="JaspertReport" class="text-center"><i class="bi bi-filetype-pdf"></i> Generar
Reporte Personas</a>

```

```
</section>
```

```
</body>
```

```
</html>
```

3. Crearemos el paquete llamado **sv.edu.udb** y crear dentro el servlet llamado JaspertReport y copiar el siguiente código.

```
package sv.edu.udb;

import jakarta.servlet.ServletException;
import jakarta.servlet.ServletOutputStream;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.export.JRPdfExporter;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.export.SimpleExporterInput;
import net.sf.jasperreports.export.SimpleOutputStreamExporterOutput;

import java.io.IOException;
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

@WebServlet(name = "JaspertReport", value = "/JaspertReport")
public class JaspertReport extends HttpServlet {

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

        Connection conexion = null;

        try {

            response.setContentType("application/pdf");

            //Nombre del reporte

            response.setHeader("Content-Disposition","attachment;
filename=\"ReportePersonas.pdf\";");

            //Devuelve un objeto de la clase ServletOutputStream que modela un flujo de salida que
            permite la escritura de datos a nivel de bytes

            ServletOutputStream out = response.getOutputStream();

            String pathReporte;

            //Conexion BD

            /*Context init = new InitialContext();
```



```
Context context = (Context) init.lookup("java:comp/env");

DataSource dataSource =(DataSource)context.lookup("jdbc/mysql");

conexion = dataSource.getConnection();*/

try
{
//obtenemos el driver de para mysql

    Class.forName("com.mysql.jdbc.Driver");
// Se obtiene una conexión con la base de datos. 2

    conexion = DriverManager.getConnection

        ("jdbc:mysql://localhost/personabdd","root", "");
// Permite ejecutar sentencias SQL sin parámetros

}

catch (ClassNotFoundException e1) {
//Error si no puedo leer el driver de MySQL

    System.out.println("ERROR:No encuentro el driver de la BD: "+e1.getMessage());

}


//Ruta del archivo .jasper

pathReporte=getServletContext().getRealPath("/Reporte1.jasper");

//Pasar de parametro la ruta del .jasper

JasperReport reporte = (JasperReport) JRLoader.loadObjectFromFile(pathReporte);
```

//Se necesitan 3 parametros el reporte,los parametros si el sql los necesitara y la conexion a la BD

```
JasperPrint jasperPrint = JasperFillManager.fillReport(reporte,null, conexion);
```

```
JRPdfExporter exporter = new JRPdfExporter();
```

```
exporter.setExporterInput(new SimpleExporterInput(jasperPrint));
```

```
exporter.setExporterOutput(new SimpleOutputStreamExporterOutput(out));
```

```
exporter.exportReport();
```

```
}
```

```
catch (SQLException | JRException ex) {
```

```
    Logger.getLogger(JaspertReport.class.getName()).log(Level.SEVERE, null, ex);
```

```
}finally{
```

```
    try {
```

```
        conexion.close();
```

```
    } catch (SQLException ex) {
```

```
        Logger.getLogger(JaspertReport.class.getName()).log(Level.SEVERE, null, ex);
```

```
    }
```

```
}
```

```
}
```

```
@Override
```

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

```
    processRequest(request,response);
```

```
}
```

```
@Override
```

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

```

ServletException, IOException {

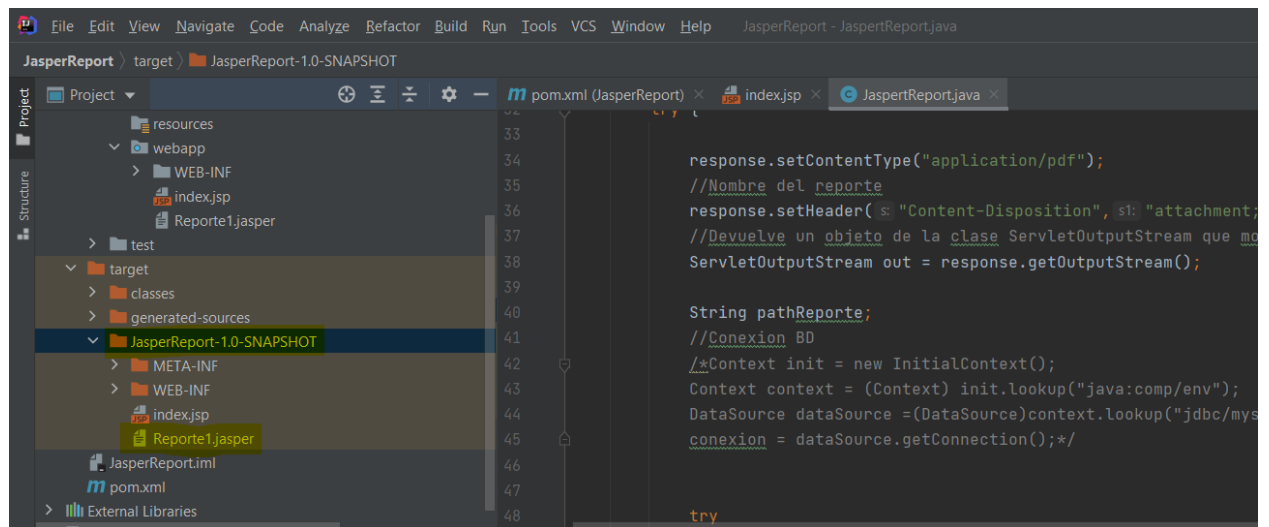
    processRequest(request,response);

}

}

```

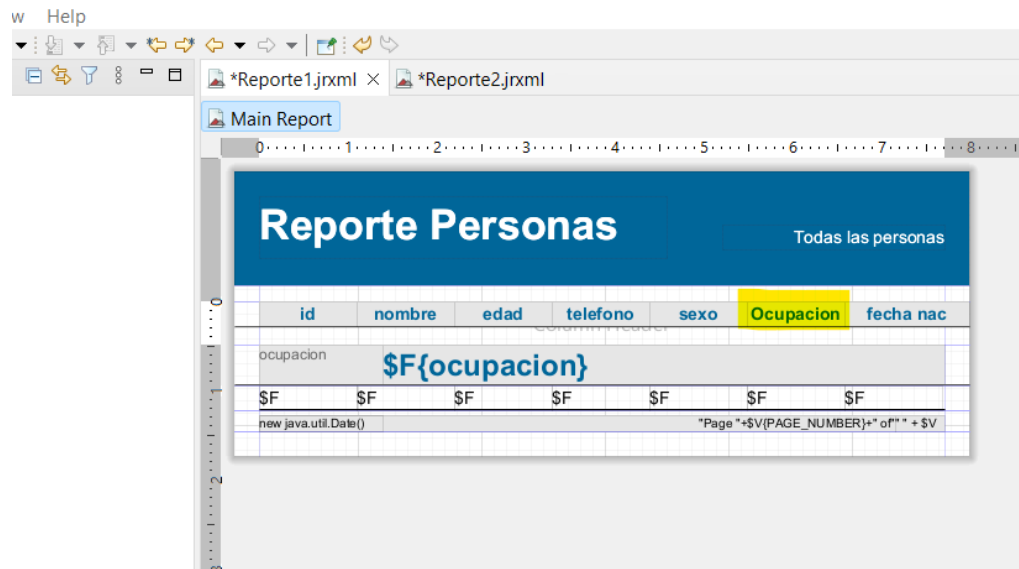
4. Correr la aplicación web y probar en el navegador de su preferencia.
5. **Advertencia:** Si el archivo pdf da un error, es porque el archivo..jasper no ha sido integrado en el despliegue del proyecto, bastara con hacer una copia de este y llevarlo a la carpeta target y buscar dentro JasperReport-1.0-SNAPSHOT, como se muestra en la figura. Esto para todos los recursos que se integren al proyecto.



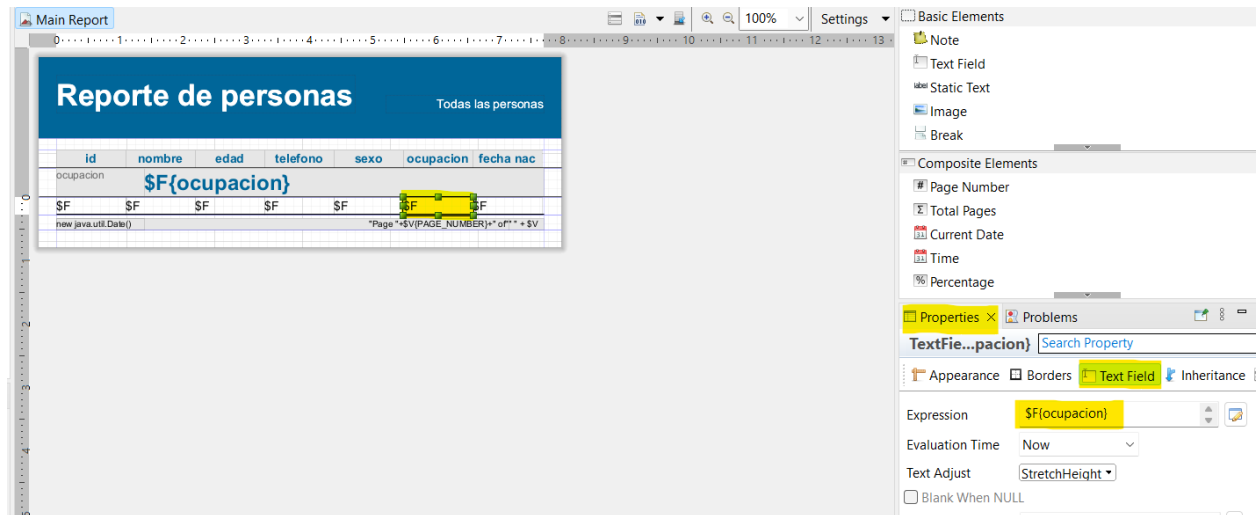
Reportes con parámetros

Para esta parte crearemos reportes enviándoles parámetros desde la JSP, de tal manera que podamos seleccionar los datos que necesitamos ver en el reporte.

1. Para comenzar a ir a JasperSoft Studio, seleccionamos en el menú **file** la opción **new** y le colocamos el nombre **“Reporte2.jrxml”** siguiendo los mismos pasos del reporte anterior. Tampoco olvidar agrupar por **ocupacion** y hacer el sort también usando el parámetro ocupación, la idea es generar un reporte similar al anterior. ahora no pintemos el id de la ocupación, si no el texto (doctor, emprendedor y profesor) correspondientemente.



2. Para ello deberá seleccionar el parámetro \$F y posteriormente cambiar en las propiedades de ese control el text fiel, como se muestra en la figura.

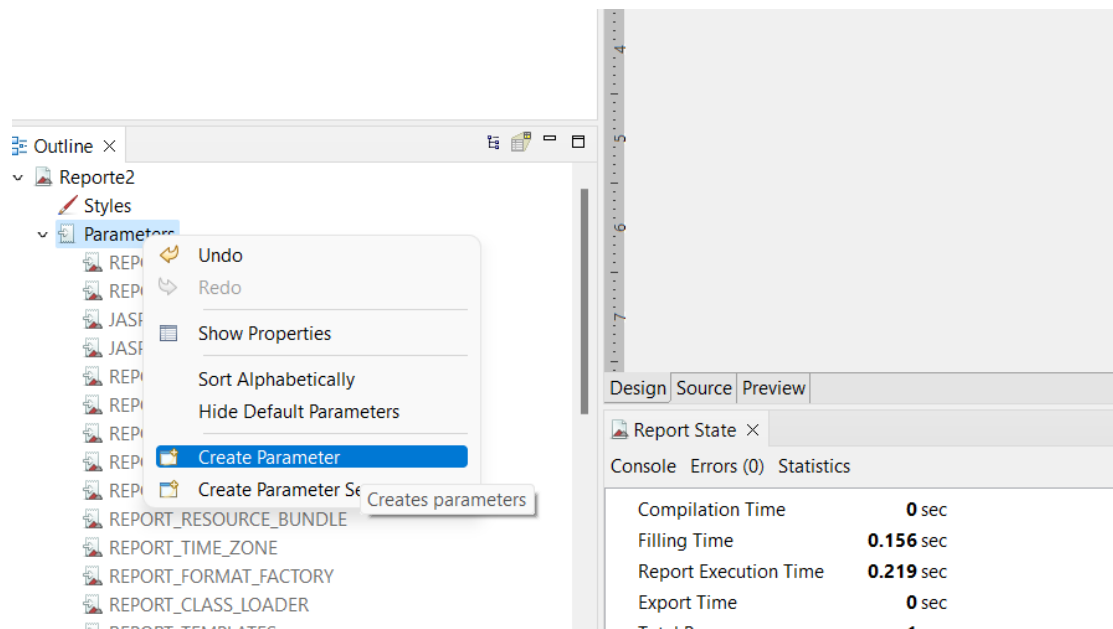


3. Dar click en previsualizar y ver los cambios en el documento.

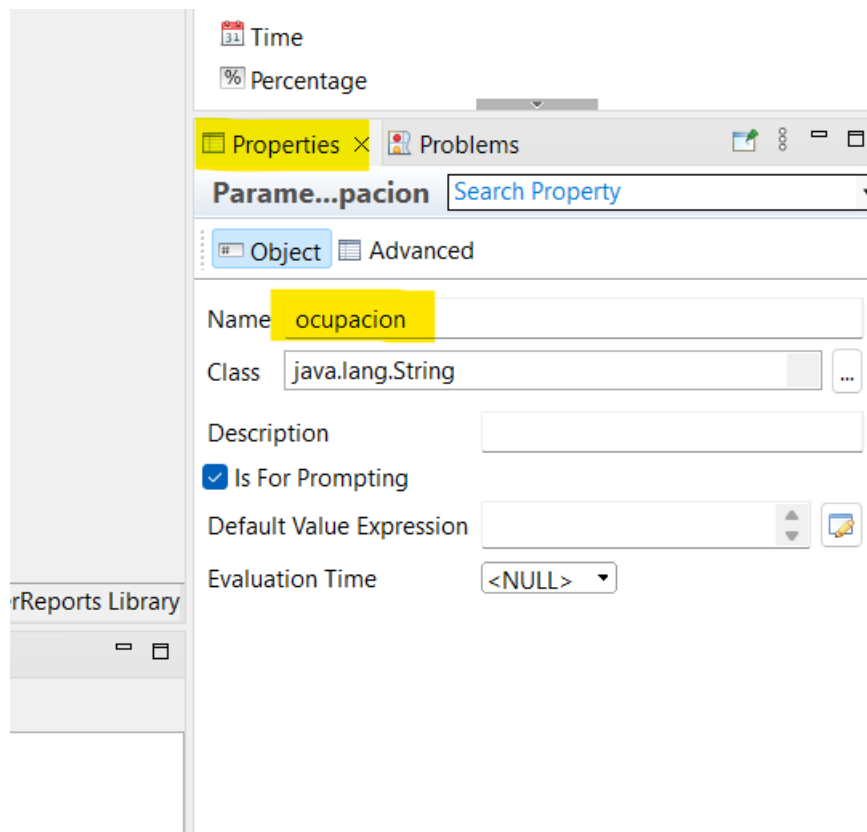
The screenshot shows a report viewer interface with two tabs: '*Reporte1.jrxml' and '*Reporte2.jrxml'. The main content area displays a report titled 'Reporte de personas' with a subtitle 'Todas las personas'. The report contains a table with the following data:

id	nombre	edad	telefono	sexo	ocupacion	fecha nac
Doctor						
1	Alejandro Pineda Saravia	45	7722-4455	Masculino	Doctor	2/5/76, 12:00 AM
5	Jocelyn Martinez	21	7894-1239	Femenino	Doctor	7/1/00, 12:00 AM
Emprendedor						
2	Fernando Calderón	21	7667-7890	Masculino	Emprendedor	5/7/01, 12:00 AM
4	Alejandra Saravia	28	7724-0678	Femenino	Emprendedor	3/8/94, 12:00 AM
Profesor						
3	Juan Machado	25	7724-5678	Masculino	Profesor	5/11/97, 12:00 AM
6	Pamela Quinteros	22	6689-8912	Femenino	Profesor	1/5/99, 12:00 AM

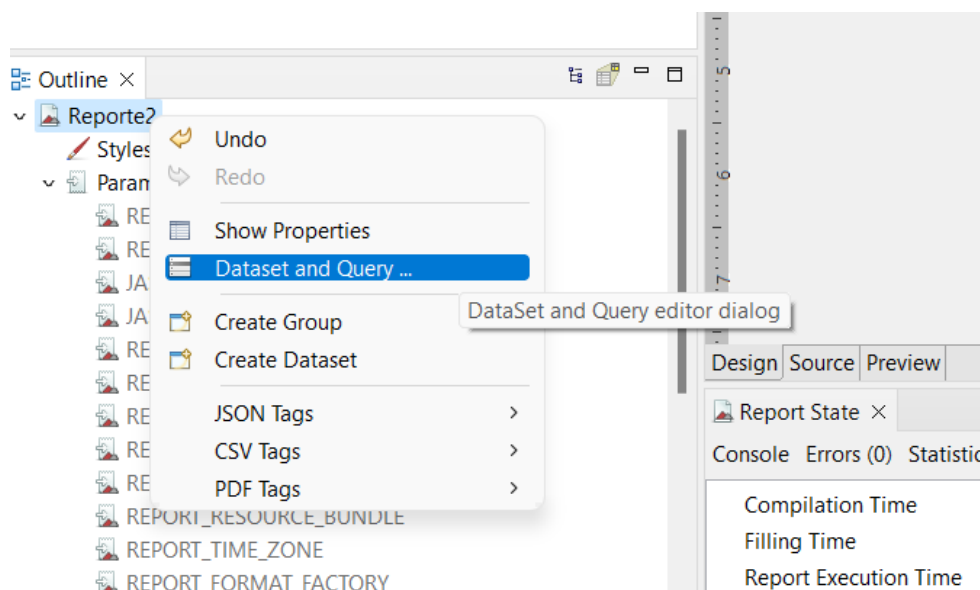
4. Finalmente agregaremos parámetros a nuestro documento dinámico, para ello dar click en **parameters > create parameter**.



5. Dar click derecho sobre el parámetro y **show properties**, finalmente cambiar el name como se muestra en la figura.



6. Finalmente seleccionaremos el reporte, daremos click derecho y daremos a la opción **Dataset and Query**.



7. Ya allí bastara con modificar la consulta y arrastrar el parámetro creado, este se encuentra en la pestaña **parameters**.

Dataset and Query Dialog

Adaptador 1

Query Java Bean

Language sql

Read Fields

```

1 SELECT personabdd.persona.id_persona,
2 personabdd.persona.nombre_persona,
3 personabdd.persona.edad_persona,
4 personabdd.persona.telefono_persona,
5 personabdd.persona.sexo_persona,
6 personabdd.persona.id_ocupacion,
7 personabdd.persona.fecha_nac,
8 personabdd.ocupaciones.id_ocupacion,
9 personabdd.ocupaciones.ocupacion
10 FROM personabdd.persona
11 INNER JOIN personabdd.ocupaciones ON
12 personabdd.persona.id_ocupacion = personabdd.ocupaciones.id_ocupacion
13 WHERE personabdd.ocupaciones.ocupacion = ${P(ocupacion)}
  
```

Parameter Name	Is For Prompt	Class Type	Description	Default Value E...	Evaluation Time	Properties
SORT_FIELDS		java.util.List				
FILTER		net.sf.jasperrep...				
REPORT_VIRT...		net.sf.jasperrep...				
IS_IGNORE_P...		java.lang.Bool...				
ocupacion	<input checked="" type="checkbox"/>	java.lang.String				

Fields Parameters Sorting Filter Expression Data preview

OK Cancel

8. Ahora proceda dar preview al reporte y veras que se solicita el parámetro creado. Una vez ingresado, dar click al botón para su ejecución.

*Reporte1.jrxml *Reporte2.jrxml

Adaptador 1 Java Page 1 of 1 100%

Input Parameters

ocupacion

Emprendedor

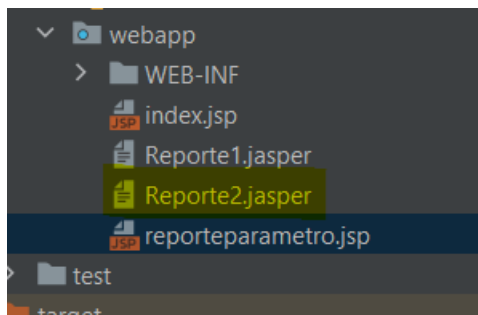
The report requires that you specify the value for some parameters before running it.

9 . Correr y ver el resultado.



id	nombre	edad	telefono	sexo	ocupacion	fecha nac
2	Fernando Calderón	21	7667-7890	Masculino	Emprendedor	5/7/01, 12:00 AM
4	Alejandra Saravia	28	7724-0678	Femenino	Emprendedor	3/8/94, 12:00 AM

10 . Para finalizar daremos **Buil Project** para crear el archivo. jasper y lo llevaremos a la carpeta **webapp**.



11 . Crearemos la página **reporteparametro.jsp**.

```
<%--
Created by IntelliJ IDEA.
User: Rafael
Date: 30/4/2022
Time: 04:16
To change this template use File | Settings | File Templates.
```



```
--%>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <title>Reporte con parametros</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">
</head>
<body>
<h1 class="text-center">Mi Primer Reporte con iReport</h1>
<h2>Generar Reporte Parametro</h2>

  <form action="JaspertReportParameter" method="POST">
  <select name="ocupacion" class="form-control">
    <option value="Doctor">Doctor</option>
    <option value="Emprendedor">Emprendedor</option>
    <option value="Profesor">Profesor</option>
  </select>
  <input type="submit" value="Enviar" class="btn btn-success">
</form>
</body>
</html>
```

12 . Finalmente crearemos el servlet **JaspertReportParameter** dentro del paquete **sv.edu.ldb**.

```
package sv.edu.ldb;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.export.JRPdfExporter;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.export.SimpleExporterInput;
import net.sf.jasperreports.export.SimpleOutputStreamExporterOutput;

import java.io.IOException;
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import java.util.HashMap;
import java.util.Map;

@WebServlet(name = "JaspertReportParameter", value = "/JaspertReportParameter")
public class JaspertReportParameter extends HttpServlet {
    protected void processRequest (HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        Connection conexion = null;
        try {

            Map parameters      = new HashMap();

            response.setContentType("application/pdf");
            //Nombre del reporte
            response.setHeader("Content-Disposition", "attachment;
filename=\"ReportePersonasParametros.pdf\";");
            //Devuelve un objeto de la clase ServletOutputStream que modela un flujo de salida que
permite la escritura de datos a nivel de bytes
            ServletOutputStream out = response.getOutputStream();

            String pathReporte;
            //Conexion BD
            /*Context init = new InitialContext();
            Context context = (Context) init.lookup("java:comp/env");
            DataSource dataSource = (DataSource)context.lookup("jdbc/mysql");
            conexion = dataSource.getConnection();*/

            try
            {
                //obtenemos el driver de para mysql
                Class.forName("com.mysql.jdbc.Driver");
                // Se obtiene una conexión con la base de datos. 2
                conexion = DriverManager.getConnection
                ("jdbc:mysql://localhost/personabdd","root", "");
                // Permite ejecutar sentencias SQL sin parámetros

            }
            catch (ClassNotFoundException e1) {
                //Error si no puedo leer el driver de MySQL
            }
        }
    }
}

```

```

        System.out.println("ERROR:No encuentro el driver de la BD: "+e1.getMessage());
    }

    String parametro = "";
    parametro=request.getParameter("ocupacion");
    //Cargamos parametros del reporte (si tiene).
    parameters.put("ocupacion", parametro);

    //Ruta del archivo .jasper
    pathReporte=getServletContext().getRealPath("/personabdd_parametro.jasper");
    //Pasar de parametro la ruta del .jasper
    JasperReport reporte = (JasperReport) JRLoader.loadObjectFromFile(pathReporte);
    //Se necesitan 3 parametros el reporte,los parametros si el sql los necesitara y la conexion
a la BD
    JasperPrint jasperPrint = JasperFillManager.fillReport(reporte,parameters, conexion);
    JRPdfExporter exporter = new JRPdfExporter();
    exporter.setExporterInput(new SimpleExporterInput(jasperPrint));
    exporter.setExporterOutput(new SimpleOutputStreamExporterOutput(out));
    exporter.exportReport();
}
catch (SQLException | JRException ex) {
    Logger.getLogger(JaspertReport.class.getName()).log(Level.SEVERE, null, ex);
}finally{
    try {
        conexion.close();
    } catch (SQLException ex) {
        Logger.getLogger(JaspertReport.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request,response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    processRequest(request,response);
}
}

```

13 . Finalmete corra su Proyecto y visualice los resultados.

IV. REFERENCIA BIBLIOGRÁFICA

- <http://www.javatutoriales.com/2009/02/creacion-de-reportes-con-jasperrepots-y.html>
Última fecha de Visita: 11/06/2013

Hoja de cotejo.

HOJA DE EVALUACIÓN

Carnet:	Alumno:
Fecha:	Docente:
No.:	Título de la guía:

Actividad a evaluar	Criterio a evaluar	Cumplió		Puntaje
		SI	NO	
Desarrollo	Realizó los ejemplos de guía de práctica (40%)			
	Presentó todos los problemas resueltos (20%)			
	Funcionan todos correctamente y sin errores (30%)			
	Envío la carpeta comprimida y organizada adecuadamente en subcarpetas de acuerdo al tipo de recurso (10%)			
	PROMEDIO:			
Investigación complementaria	Entregó la investigación complementaria en la fecha indicada (20%)			
	Resolvió todos los ejercicios planteados en la investigación (40%)			

	Funcionaron correctamente y sin ningún mensaje de error a nivel de consola o ejecución (40%)			
	PROMEDIO:			