



JasperReports

1. JasperReports

Una de las herramientas utilizadas para crear informes desde Java sobre base de datos es **JasperReports** que es una librería open source para el desarrollo de informes con Java sobre diferentes tipos de fuentes de datos.

JasperReports está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo JEE o aplicaciones web, para generar contenido dinámico.

Con JasperReports podemos crear informes complejos que incluyan gráficos, imágenes, subinformes, etc. Podemos acceder a nuestros datos a través de JDBC, JavaBeans, XML, Hibernate, CSV, NSQL, etc.

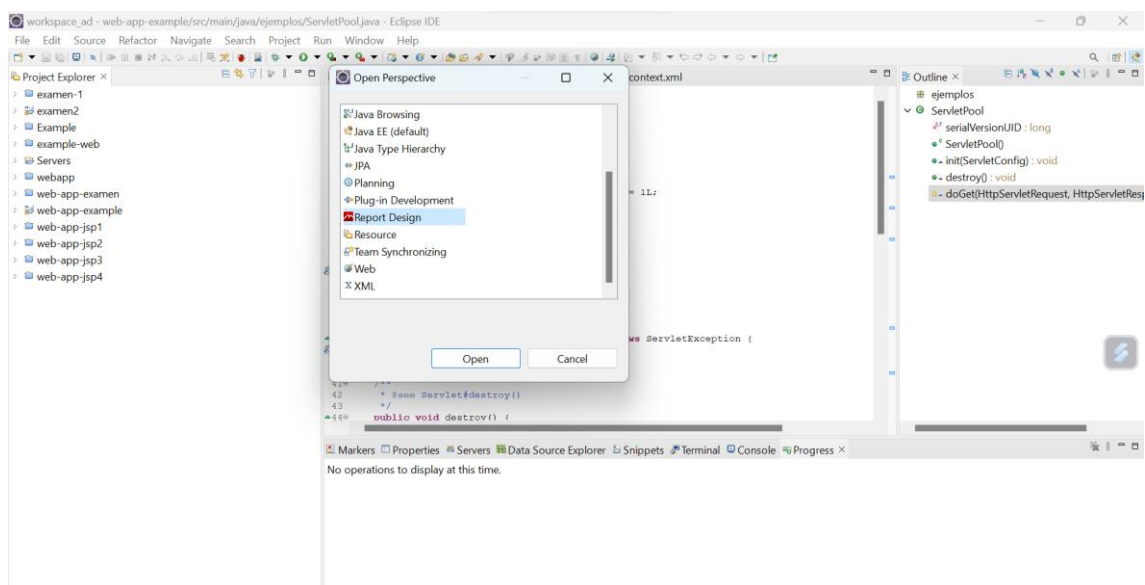
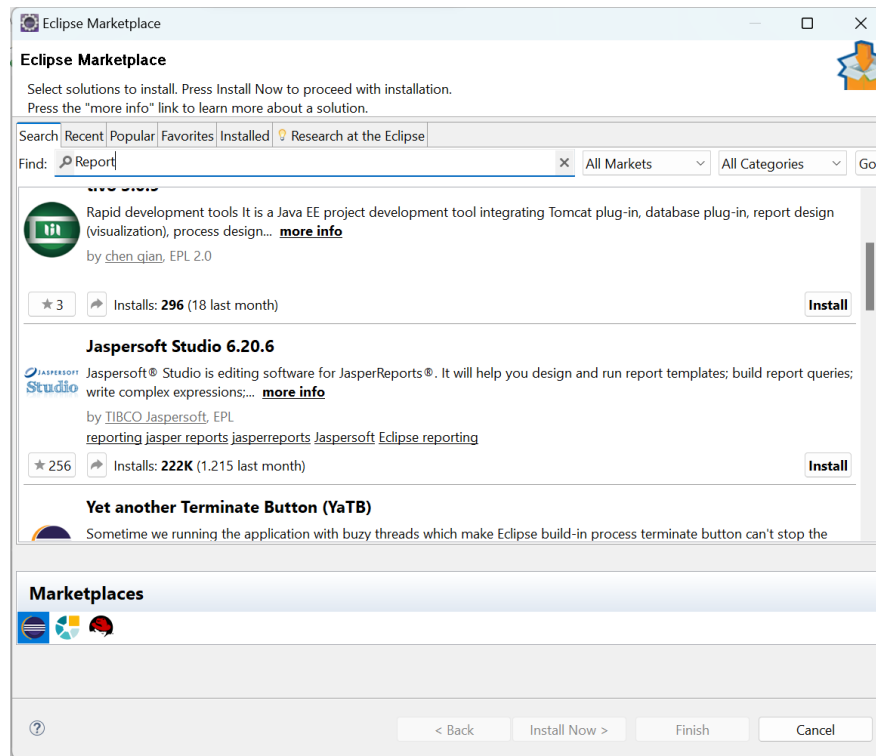
La publicación y exportación de los informes pueden ser a gran variedad de formatos: PDF, RTF, XML, XLS, CSV, HTML, XHTML, text, DOCX, OpenOffice.

Nosotros vamos a utilizar la herramienta **Jaspersoft Studio** preparada para la plataforma de Eclipse. Aunque también existen otro tipo de herramientas como iReport (Netbeans), Oracle Report, etc.

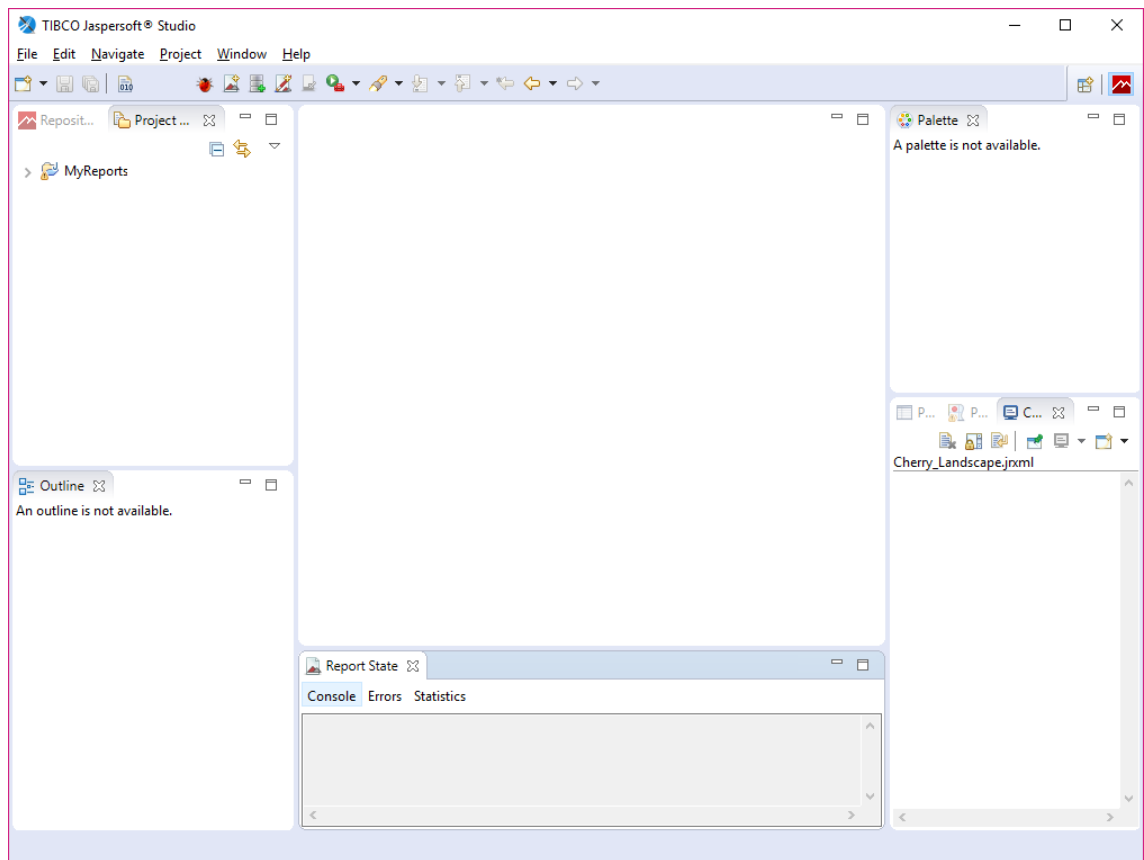
Aunque JasperReports se utiliza principalmente para capacidades de reporte a las aplicaciones basadas en la Web a través de la API de Servlet o cualquier otra biblioteca de Java EE, se podría utilizar también en la creación de aplicaciones de escritorio Java para generar informes con JasperReports.

2. Instalación y configuración de JasperSoft

Vamos a instalar **Jaspersoft Studio** en nuestro equipo, lo podemos descargar de <http://community.jaspersoft.com/project/jaspersoft-studio/releases>, solo nos pedirán un registro de usuario. La versión que descargaremos es la **TIBCOJaspersoftStudio-6.20** o desde el **MarketPlace** de Eclipse se puede hacer la descarga e integrarla en el IDE.

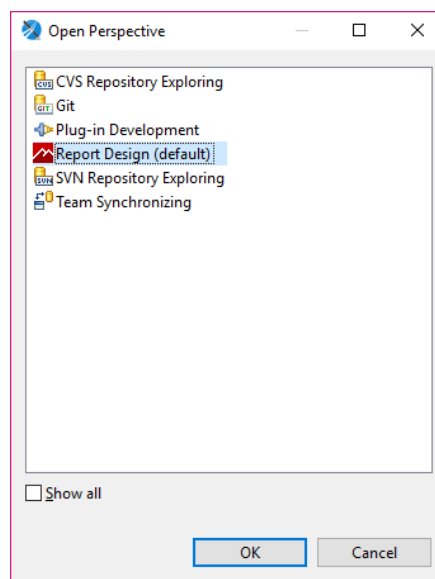


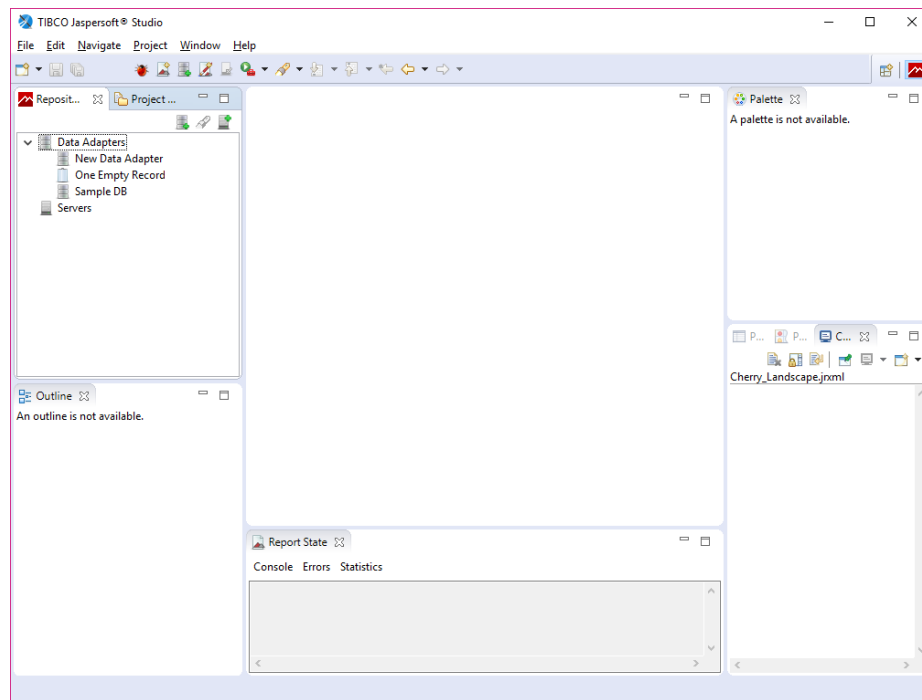
El aspecto de la interfaz una vez instalado es el siguiente:



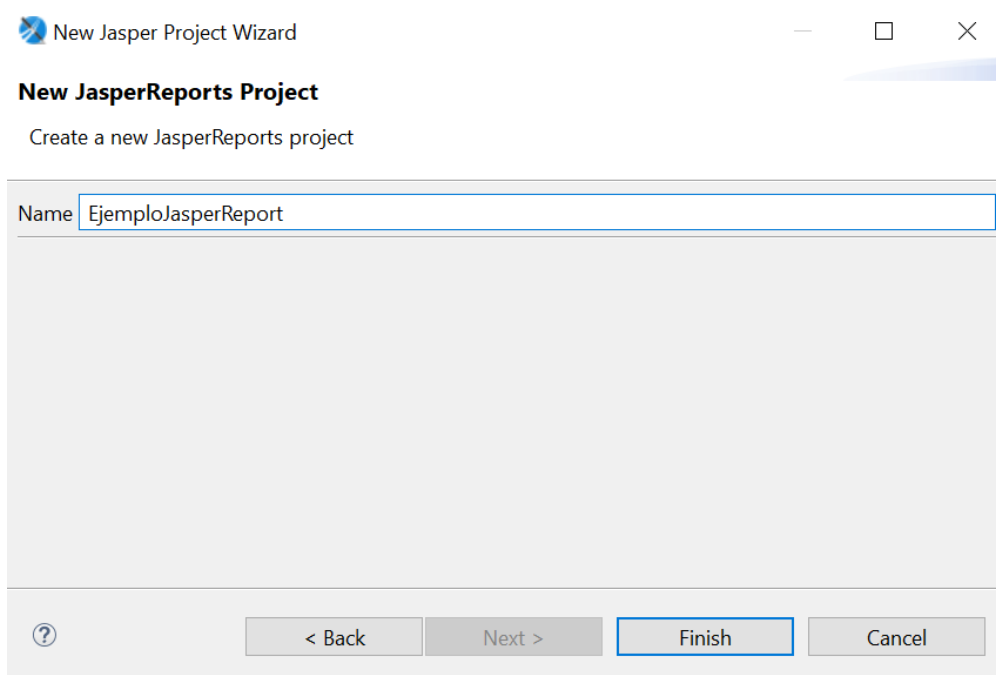
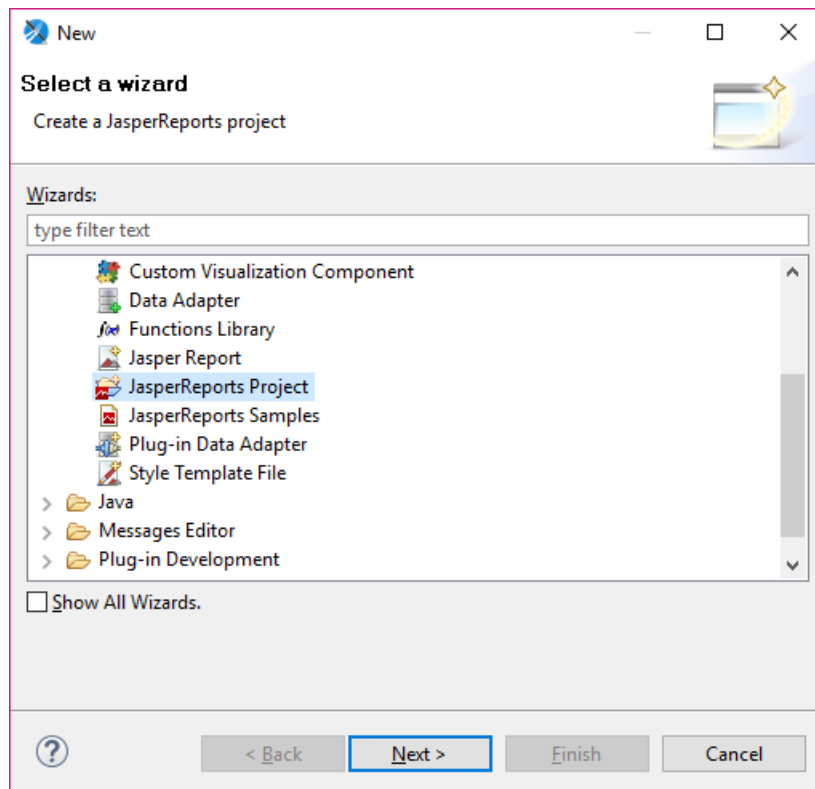
2.1 Crear proyecto JasperReport

- ✓ Primeramente vamos a abrir la perspectiva de **Reports** (**Window->Open Perspectiva->Other**)

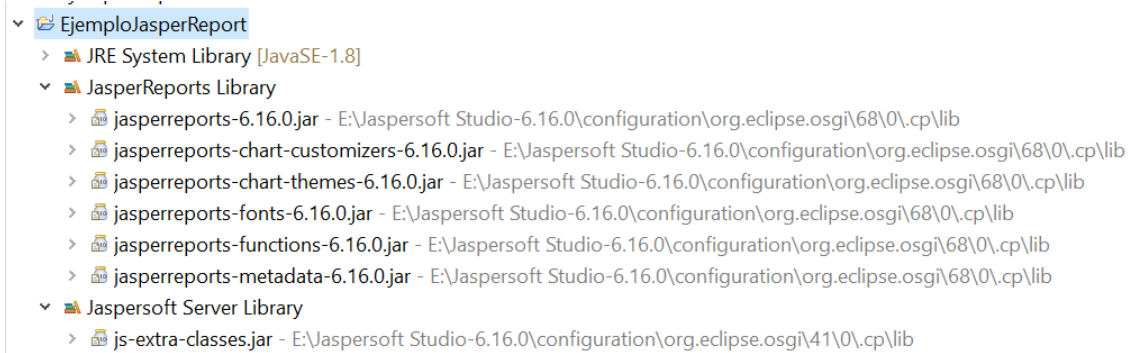




✓ A continuación creamos un proyecto nuevo (**File->New->Project**) e indicamos el nombre del proyecto a crear.

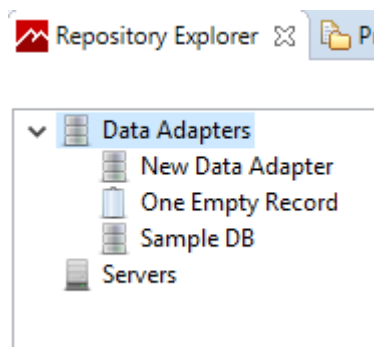


La estructura del proyecto quedaría como se ve en la imagen, cargado con sus librerías correspondientes.

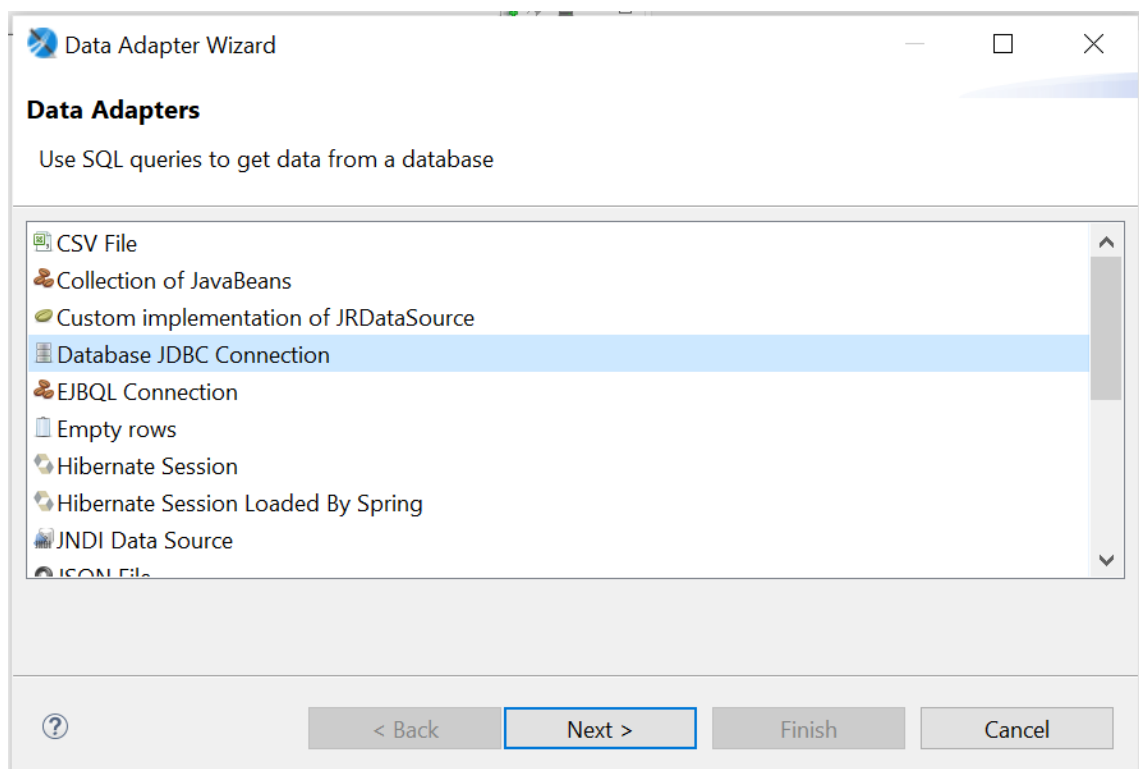


2.2 Crear fuente de datos

✓ Vamos a conectar a una base de datos con MYSQL a través del conector JDBC. Elegimos la opción **Create Date Adapter**. A continuación se elige la opción **Database JDBC Connection**.



✓ Completamos los datos de la base de datos a la que vamos a conectar



Data Adapter Wizard

Data Adapter

Database JDBC Connection

Name: New Data Adapter

JDBC Driver: com.mysql.jdbc.Driver

JDBC URL: jdbc:mysql://localhost/empresa?useTimezone=true&serverTimezone=UTC

Username: root

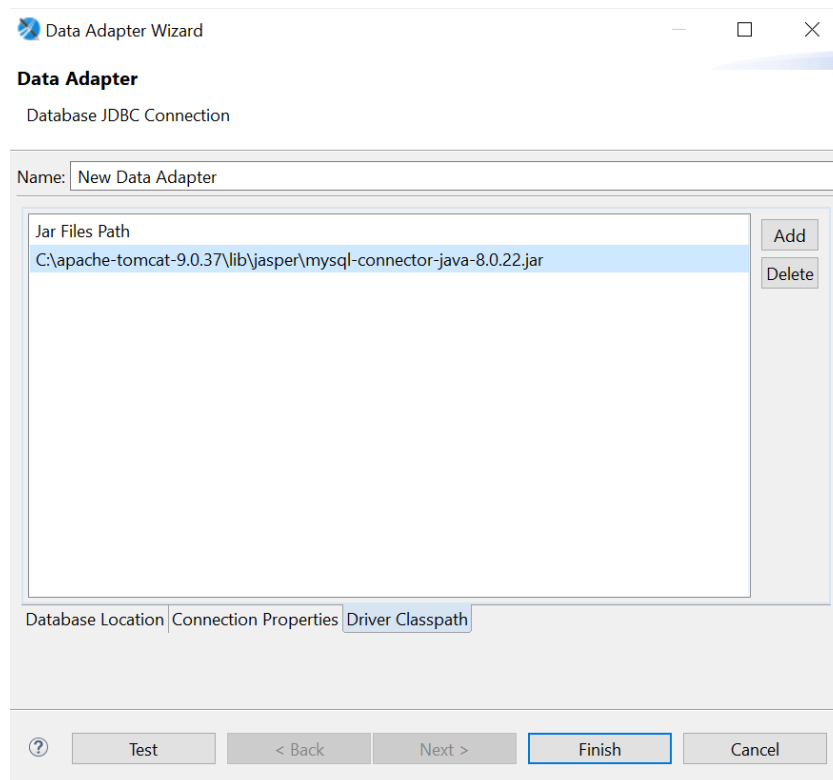
Password: ●●●●

Attention! Passwords are saved in clear text

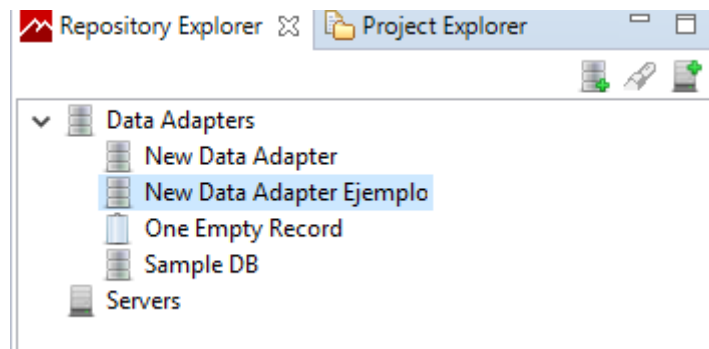
Database Location | Connection Properties | Driver Classpath

? Test < Back Next > Finish Cancel

- ✓ Si salta una excepción como al testear la conexión a la bbdd referente a la zona horaria se debe de añadir en la url-jdbc los siguientes atributos: serverTimezone y useTiemzone
- ✓ Añadimos el fichero .jar de localización de la librería del driver JDBC que vamos a utilizar para nuestra base de datos.

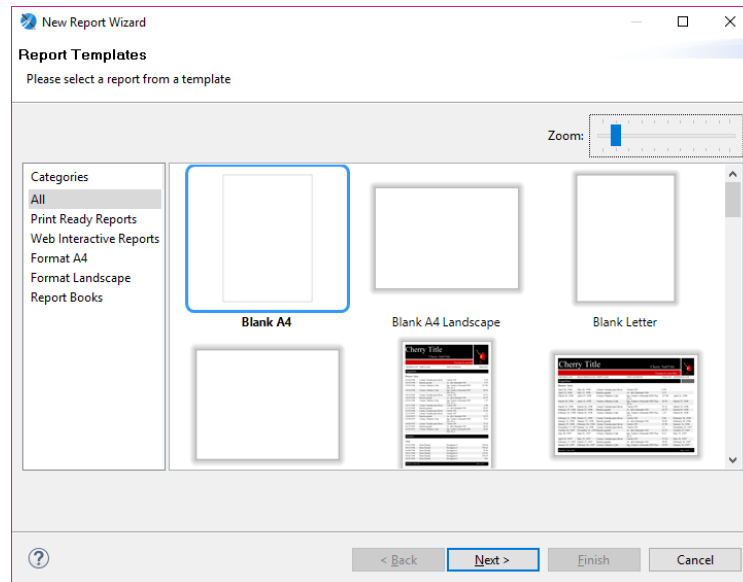


✓ Una vez finalizado este último paso ya tenemos configurada una fuente de datos de la que podremos utilizar sus datos para la creación diferentes informes que realizamos sobre esa conexión.

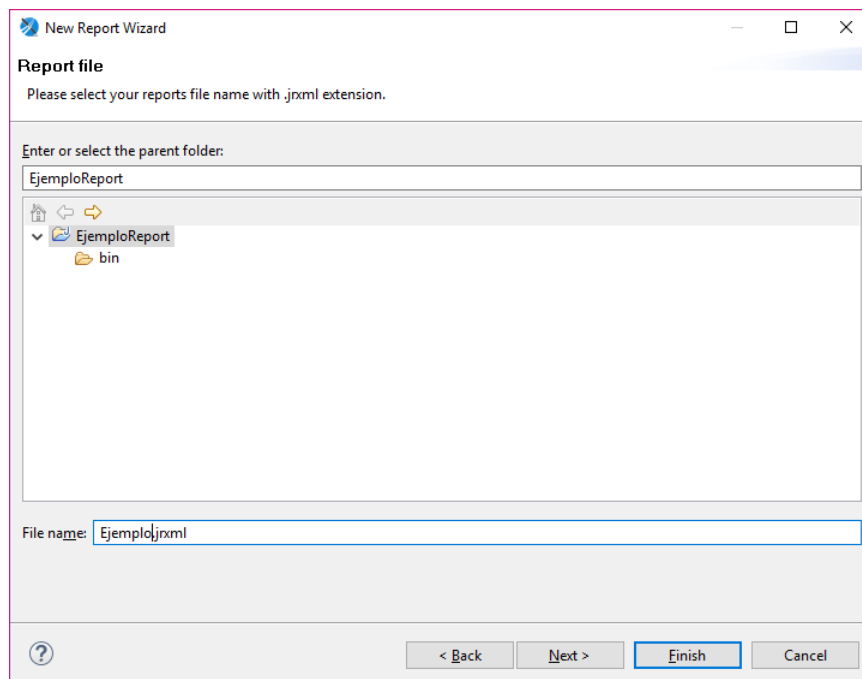


2.3 Crear un report

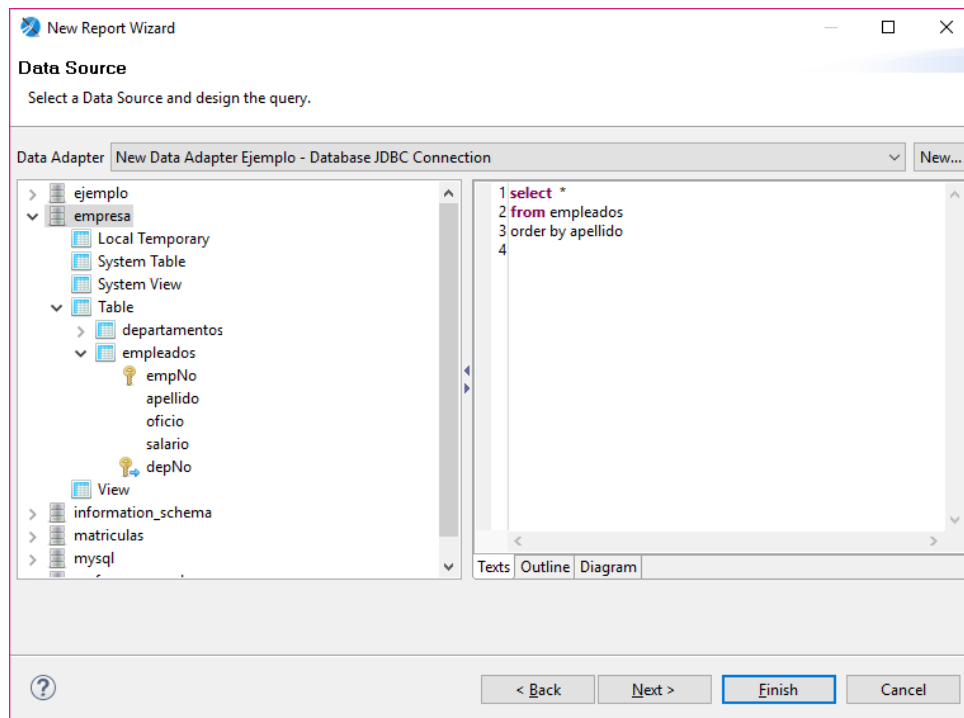
- ✓ Para crear un report desde el proyecto que acabamos de crear vamos a **New -> JasperReport..** Elegimos un report en blanco.



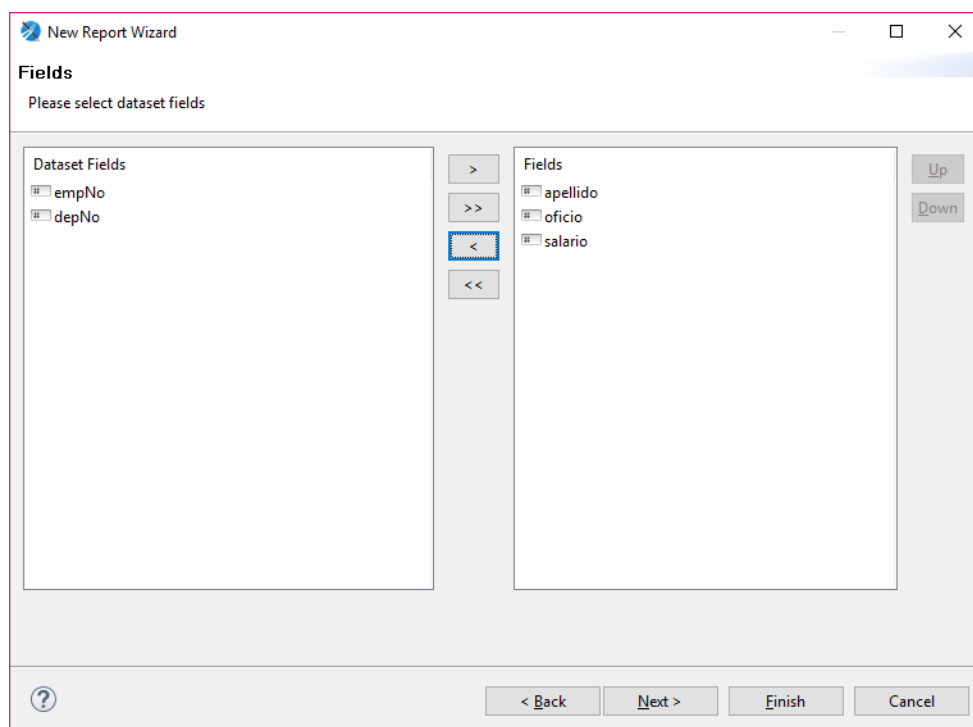
- ✓ En el siguiente paso se nos indica la carpeta donde se guardará nuestro informe que tendrá un formato jrxml, de momento dejamos la configuración por defecto e introducimos el nombre de nuestro informe, en nuestro caso ejemplo.jrxml



- ✓ Ahora seleccionamos el Datasource creado anteriormente. Añadimos la consulta que deseemos utilizar para obtener datos.



- ✓ A continuación seleccionamos los campos que queremos mostrar en el informe, ya que hay que campos que puedes necesitar para mostrar resultados en el informe pero que no quieres que se muestren directamente en el informe.

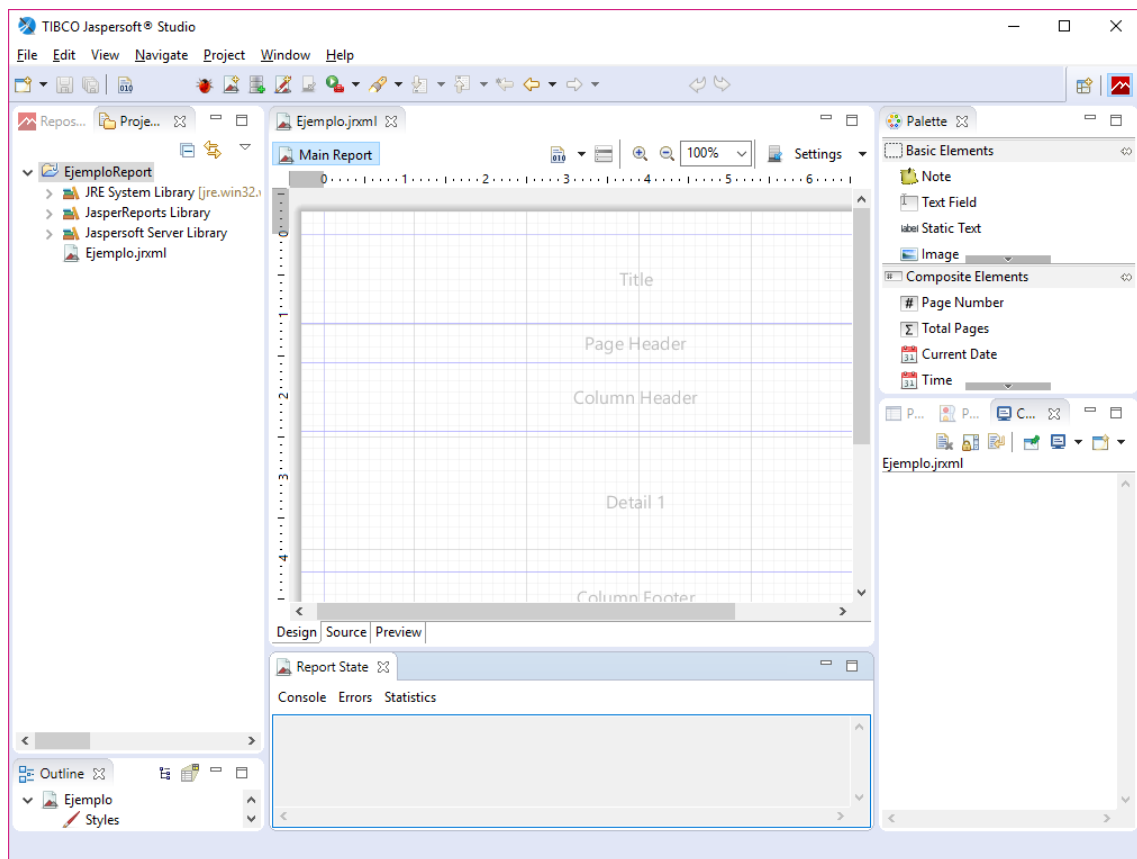


- ✓ En esta ventana podríamos crear grupos para agrupar resultados por una de las columnas.

The screenshot shows the 'New Report Wizard' window at the 'Group By' step. The title bar reads 'New Report Wizard'. Below the title bar, the text 'Group By' is followed by the instruction 'Please select fields to Group By'. The main area is divided into two panes: 'Dataset Fields' on the left and 'Fields' on the right. The 'Dataset Fields' pane contains a list of fields: 'apellido', 'oficio', and 'salario'. Between the panes are four arrow buttons: '>', '>>', '<', and '<<'. The 'Fields' pane is currently empty. To the right of the 'Fields' pane are 'Up' and 'Down' buttons. Below the panes, 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 window, there is a navigation bar with a help icon, '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel' buttons.

The screenshot shows the 'New Report Wizard' window at the 'Finish' step. The title bar reads 'New Report Wizard'. Below the title bar, the text 'Finish' is followed by the instruction 'We are ready to create your report'. The main area contains the text 'Congratulations!' in a large font, followed by 'All the information to create your new report have been successfully acquired.' and 'Press Finish to generate the report.' At the bottom of the window, there is a navigation bar with a help icon, '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel' buttons.

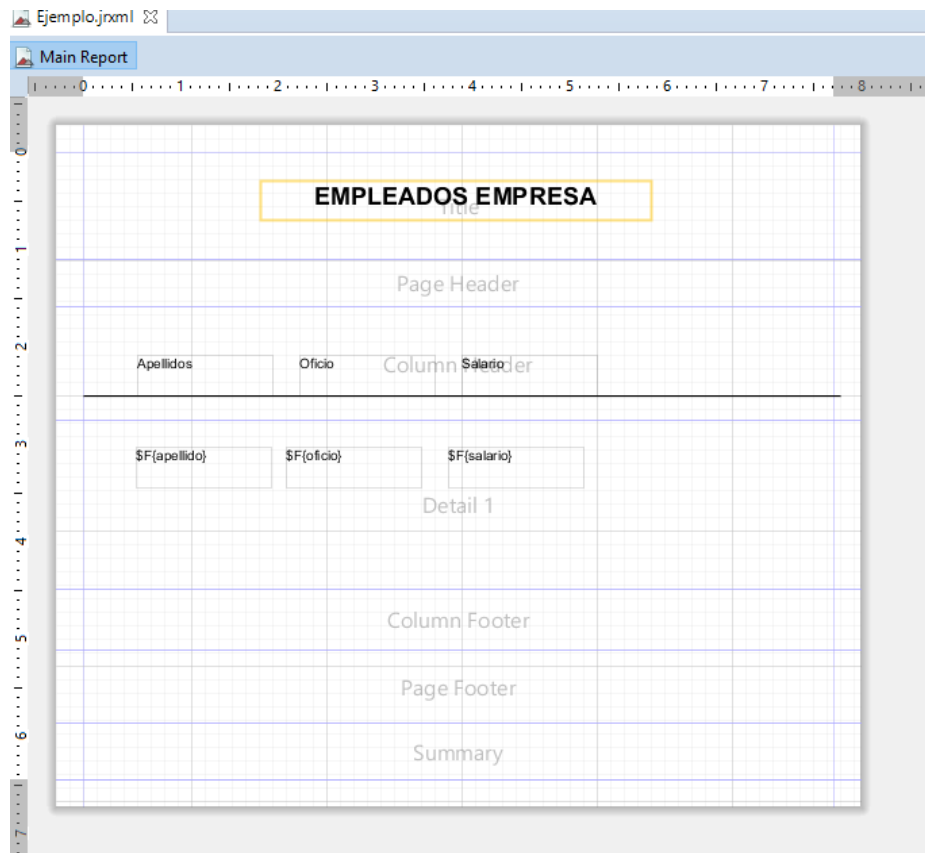
✓ Por último, quedaría generado el informe



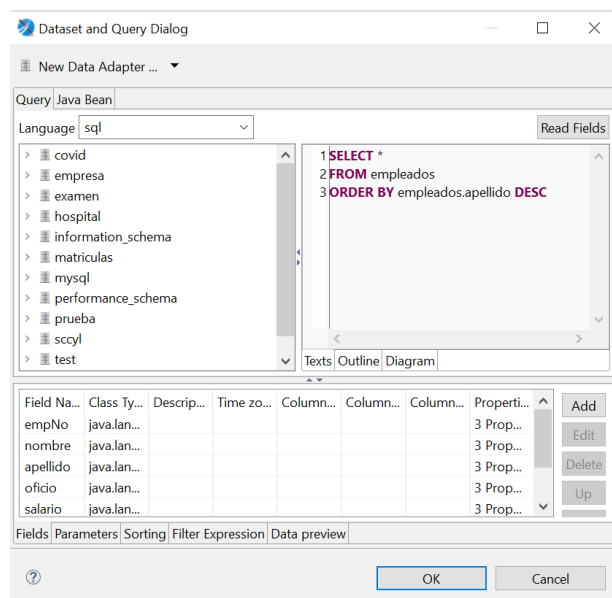
JasperReports nos permite separar los datos en **secciones**. Estas secciones incluyen:

- El **título del informe**, que aparece una vez en la parte superior del informe.
- Un **encabezado de página**, que aparece en la parte superior de cada página.
- Una **sección de detalle**, que normalmente contiene los datos de los informes primarios.
- A **pie de página**, que aparece en la parte inferior de cada página.
- Una **sección de resumen**, que aparece al final del informe.

- ✓ Ahora ya podríamos comenzar a construir el diseño del informe para nuestro ejemplo

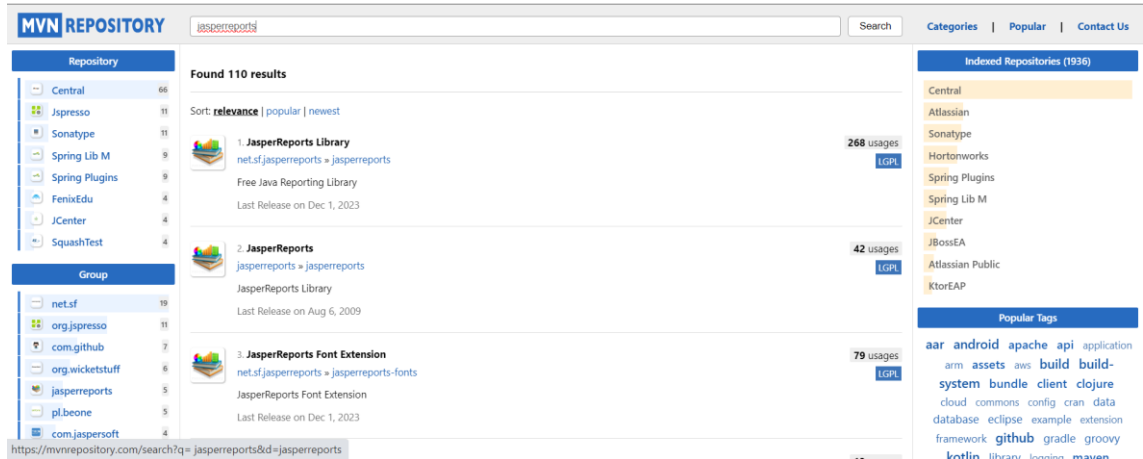


Para cargar en el diseño los campos de texto con los valores de las consultas realizadas sobre las tablas de la base de datos de empleados si no han sido cargados previamente se deben actualizar desde la siguiente opción de menú **“DataSet and Query Dialog”**.



3. Utilizar el JasperReport desde la aplicación JEE

- ✓ Lo primero que tenemos que hacer es cargar en el proyecto Maven las dependencias necesarias de JasperReport.



- ✓ Implementamos la generación del documento en código Java

```
JasperReport jasperReport = null;
JasperPrint jasperPrint = null;
Connection conn = null;

try {
    conn = (Connection) ConexionBD.getConnection();
    // Compile jrxml file.
    jasperReport = JasperCompileManager.compileReport("F:/report/Blank_A4.jrxml");
    //Imprime report
    Map<String, Object> parameters = new HashMap<String, Object>();
    jasperPrint = JasperFillManager.fillReport(jasperReport, parameters, conn);

    File outDir = new File("F:/jasperoutput");
    outDir.mkdirs();
    //Exportar a PDF
    JasperExportManager.exportReportToPdfFile(jasperPrint, "F:/jasperoutput\\Blank_A4.pdf");
    System.out.println("Documento creado!");
    // se muestra en una ventana aparte para su descarga
    JasperPrint jasperPrintWindow = JasperFillManager.fillReport(
        "F:/report/Blank_A4.jasper", null,
        conn);
    JasperViewer jasperViewer = new JasperViewer(jasperPrintWindow);
    jasperViewer.setVisible(true);

} catch (JRException e) {
    e.printStackTrace();
} catch (Exception e) {}
// TODO Auto-generated catch block
e.printStackTrace();

} finally {
    if (conn != null)
        try {
            conn.close();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
}
```

