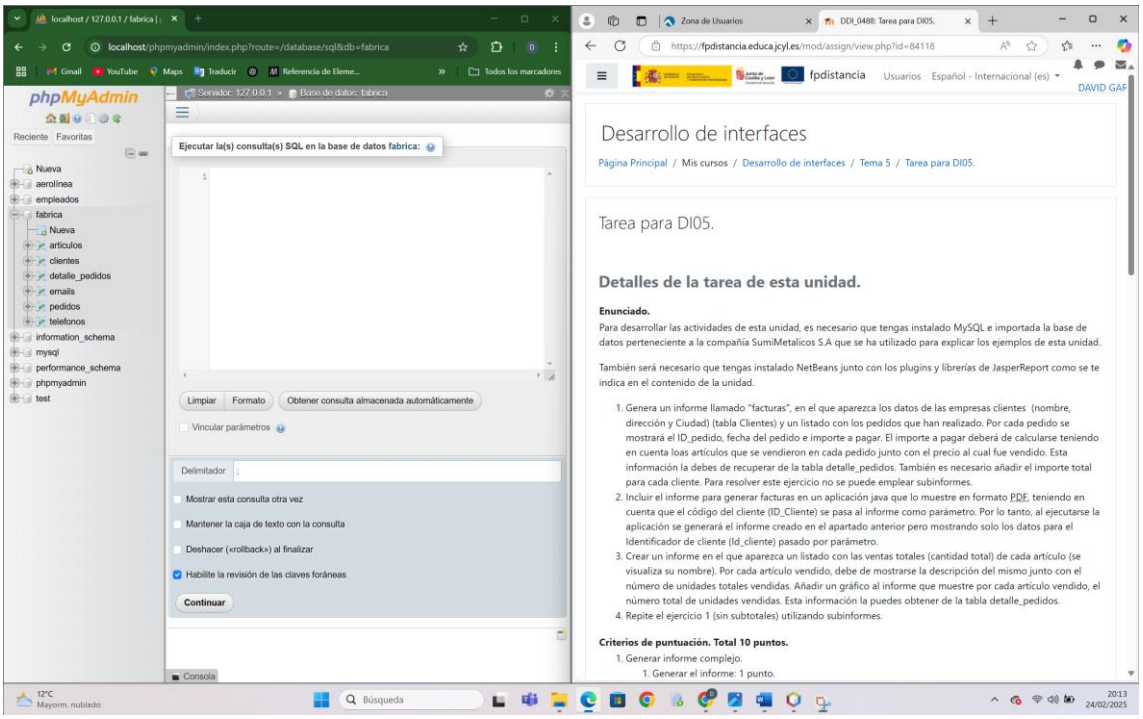
ENUNCIADO

Para desarrollar las actividades de esta unidad, es necesario que tengas instalado MySQL e importada la base de datos perteneciente a la compañía SumiMetalicos S.A que se ha utilizado para explicar los ejemplos de esta unidad.

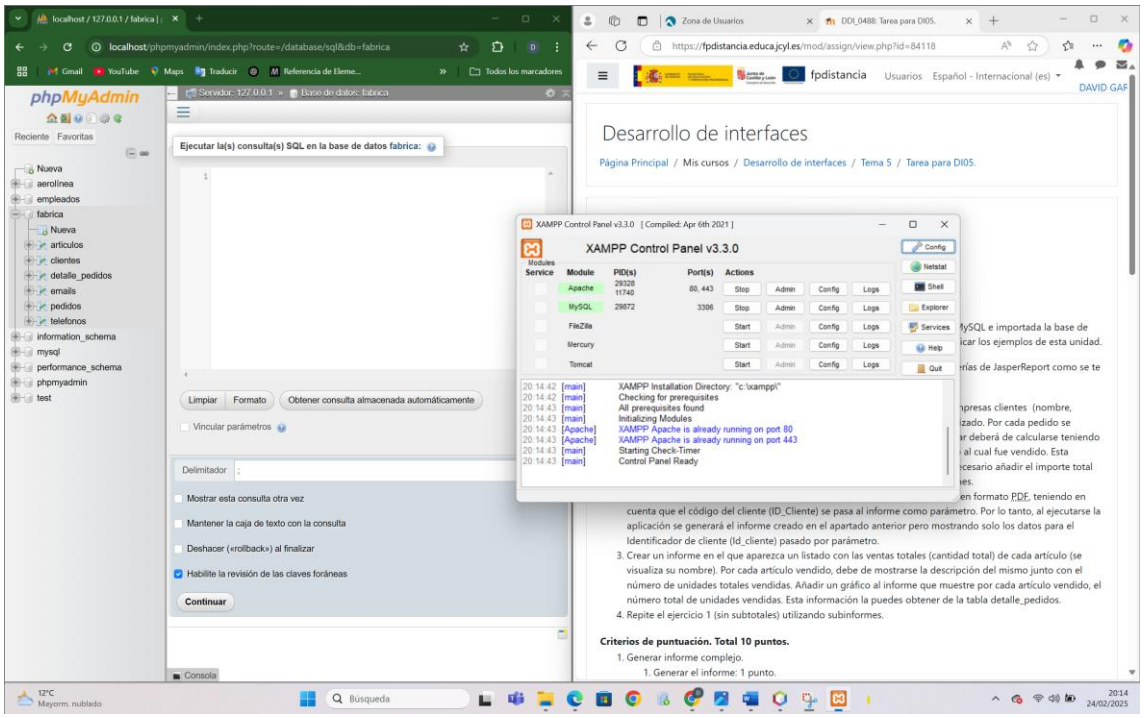
También será necesario que tengas instalado NetBeans junto con los plugins y librerías de JasperReport como se te indica en el contenido de la unidad.

Preparación de la base de datos y conexión con Jreport en NetBeans.

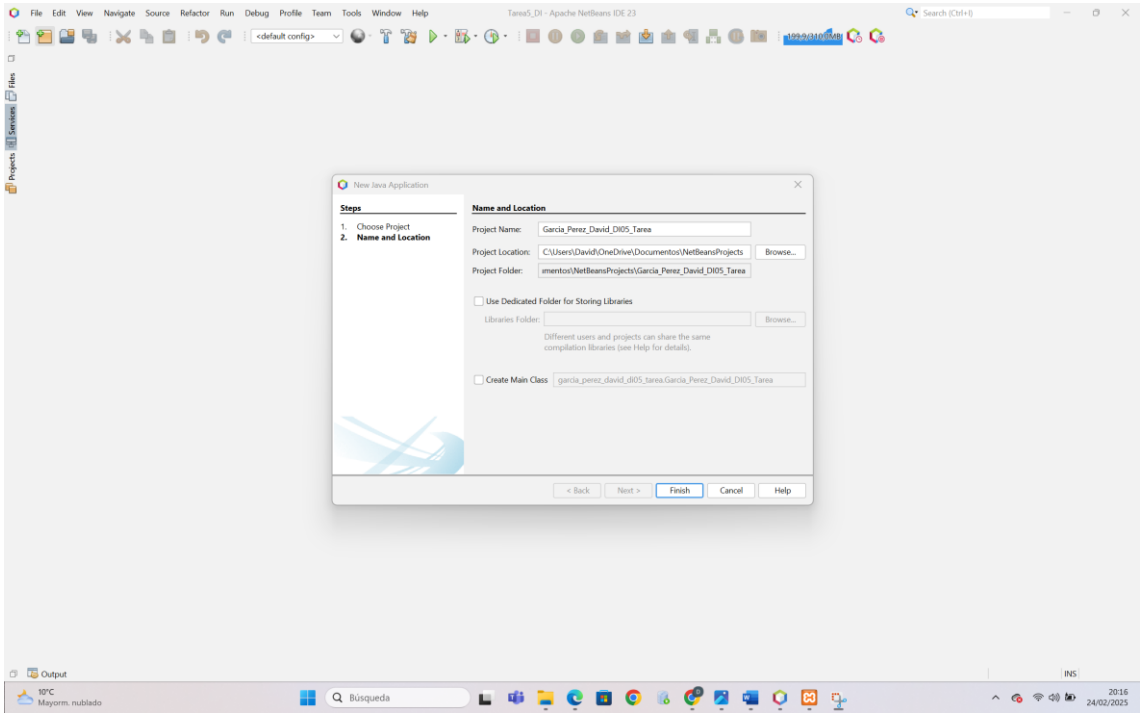
Para realizar este ejercicio, en primer lugar, procederemos a crear la base de datos. Esta la haremos en PHP, y crearemos la conexión con Apache utilizando XAMPP.



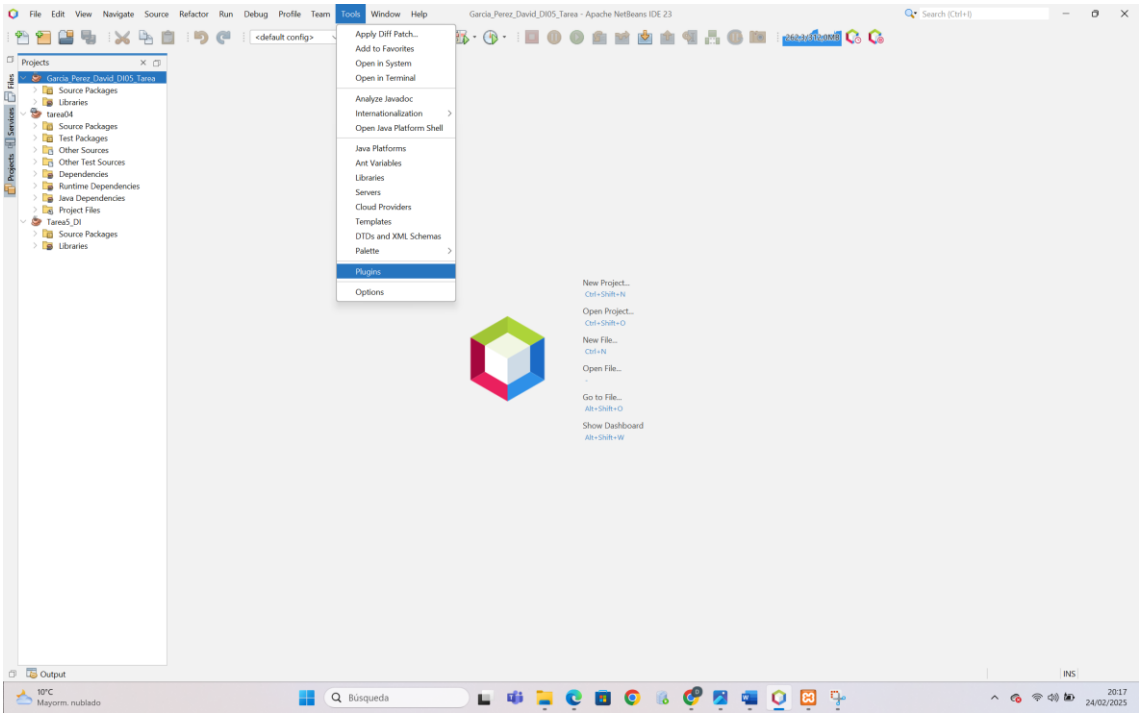
Una vez creada la base de datos procedemos a realizar la conexión en XAMPP con MySQL y Apache:



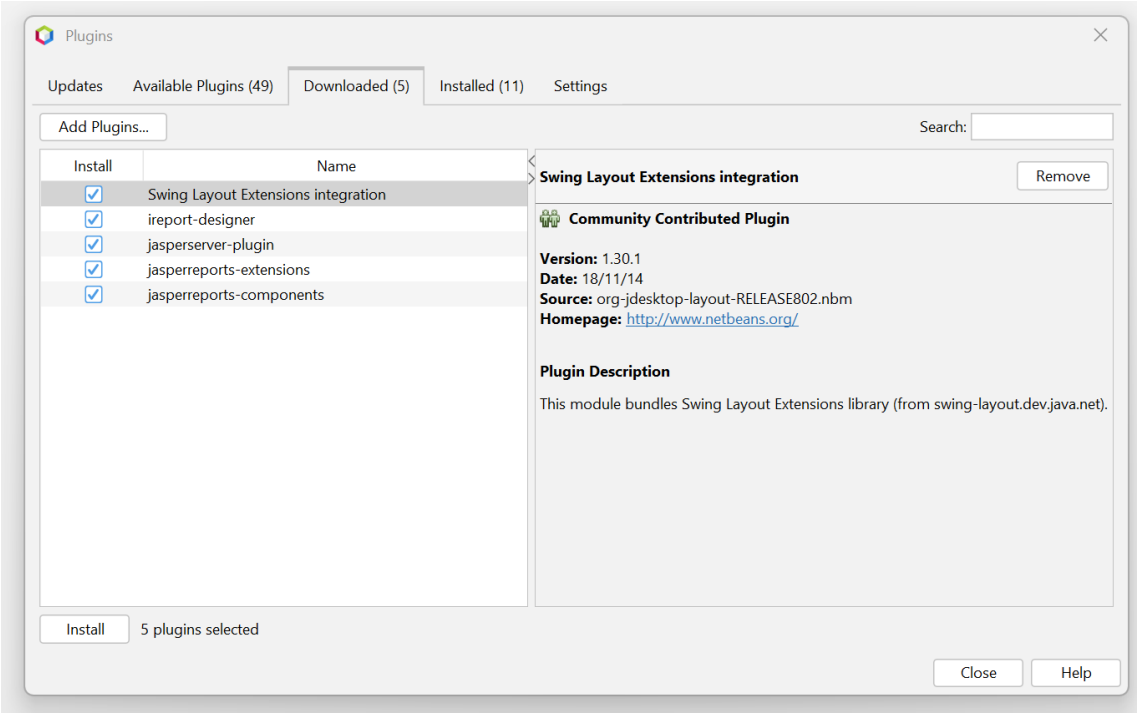
Una vez creada la base de datos procedemos a crear el proyecto en NetBeans, sin el método Main:



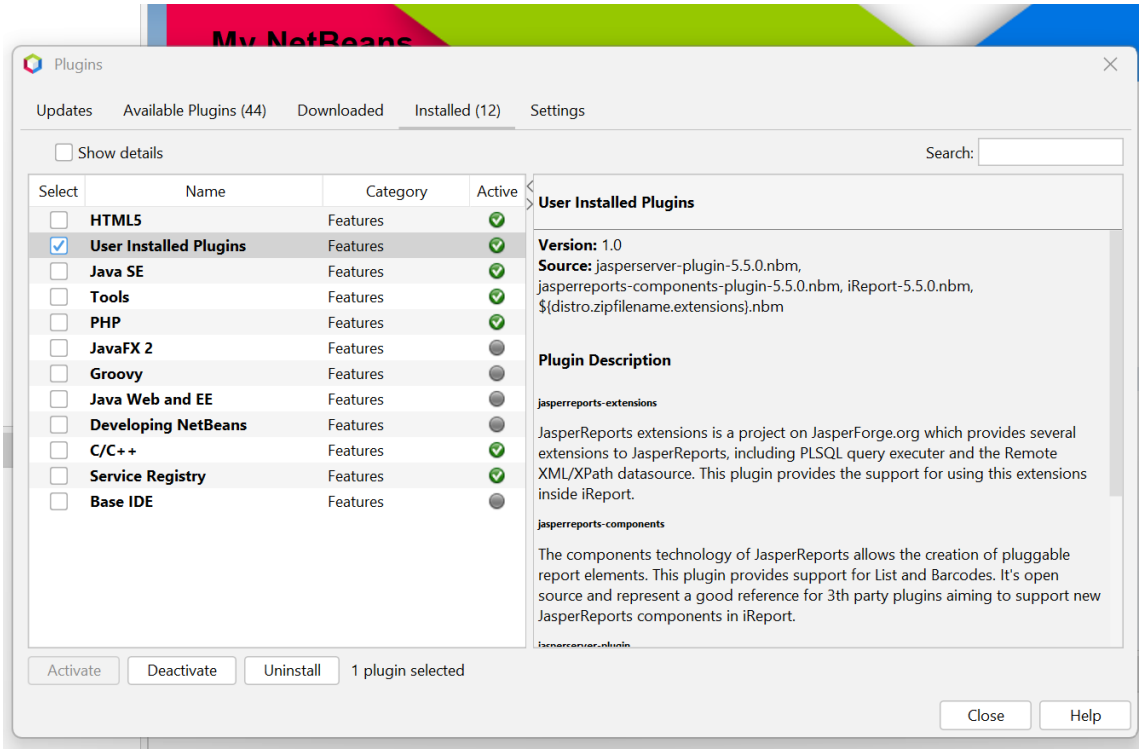
A continuación, instalamos el plugin de Jreport. Para ello vamos a Tools/Plugins



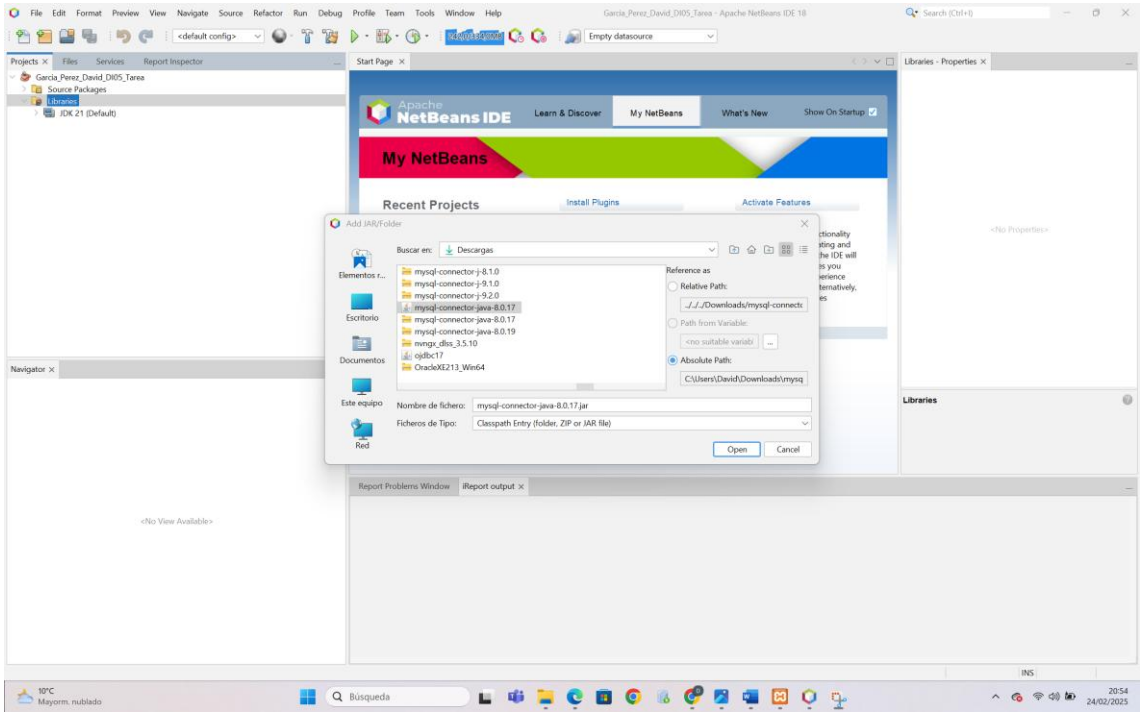
Añadimos los plugins descargados junto a la extensión Swing layouts y procedemos a instalarlos:



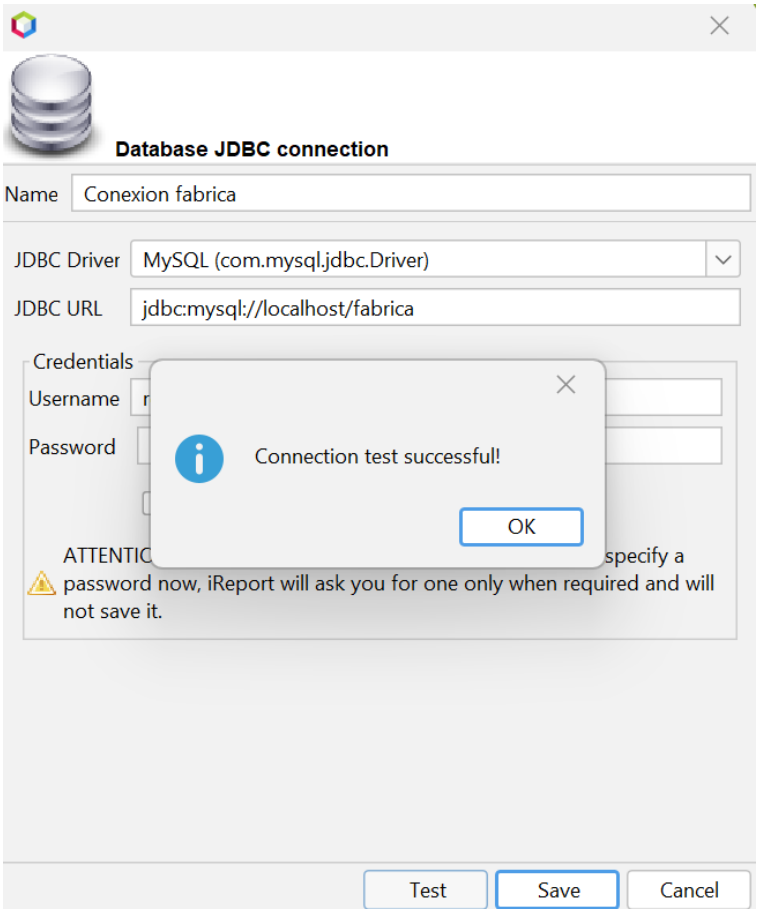
Como daba error debido a que este plugin ya no es compatible con las nuevas versiones de Netbeans y el JDK instalado, procedemos a instalar Netbeans 18. En este caso, ya dejó instalarlo:



Para poder conectarnos a la base de datos, deberemos agregar el archivo .jar del conector de MySQL a las librerías:

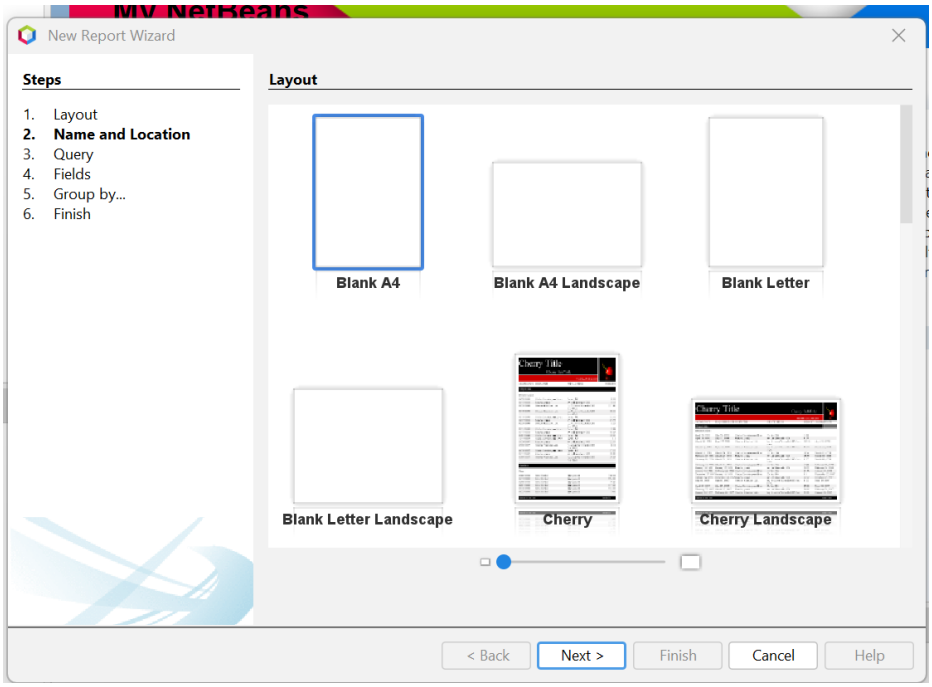


Seguidamente procedemos a crear la conexión:

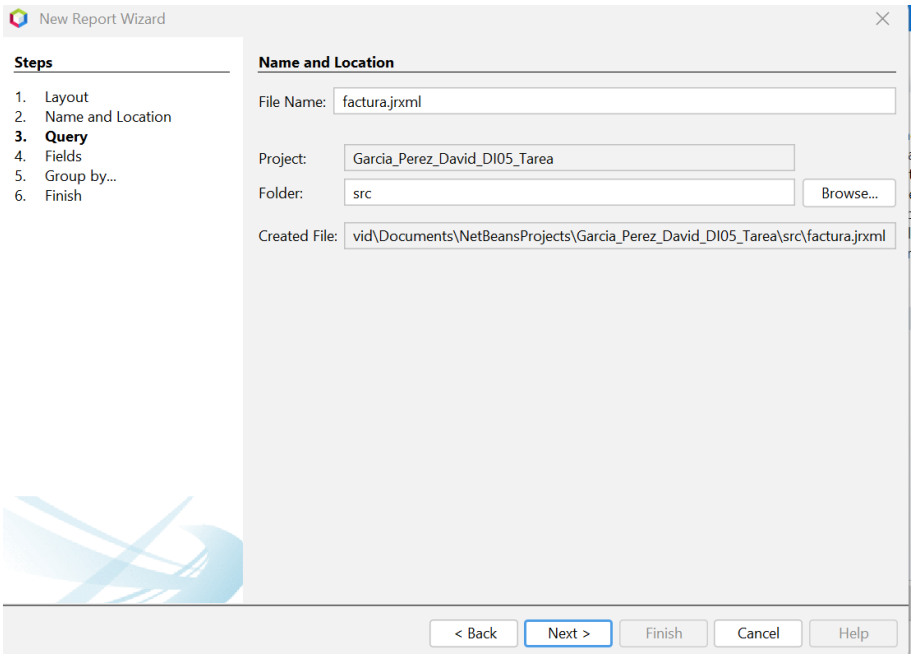


1. **Genera un informe llamado "facturas", en el que aparezca los datos de las empresas clientes (nombre, dirección y Ciudad) (tabla Clientes) y un listado con los pedidos que han realizado. Por cada pedido se mostrará el ID_pedido, fecha del pedido e importe a pagar. El importe a pagar deberá de calcularse teniendo en cuenta loas artículos que se vendieron en cada pedido junto con el precio al cual fue vendido. Esta información la debes de recuperar de la tabla detalle_pedidos. También es necesario añadir el importe total para cada cliente. Para resolver este ejercicio no se puede emplear subinformes.**

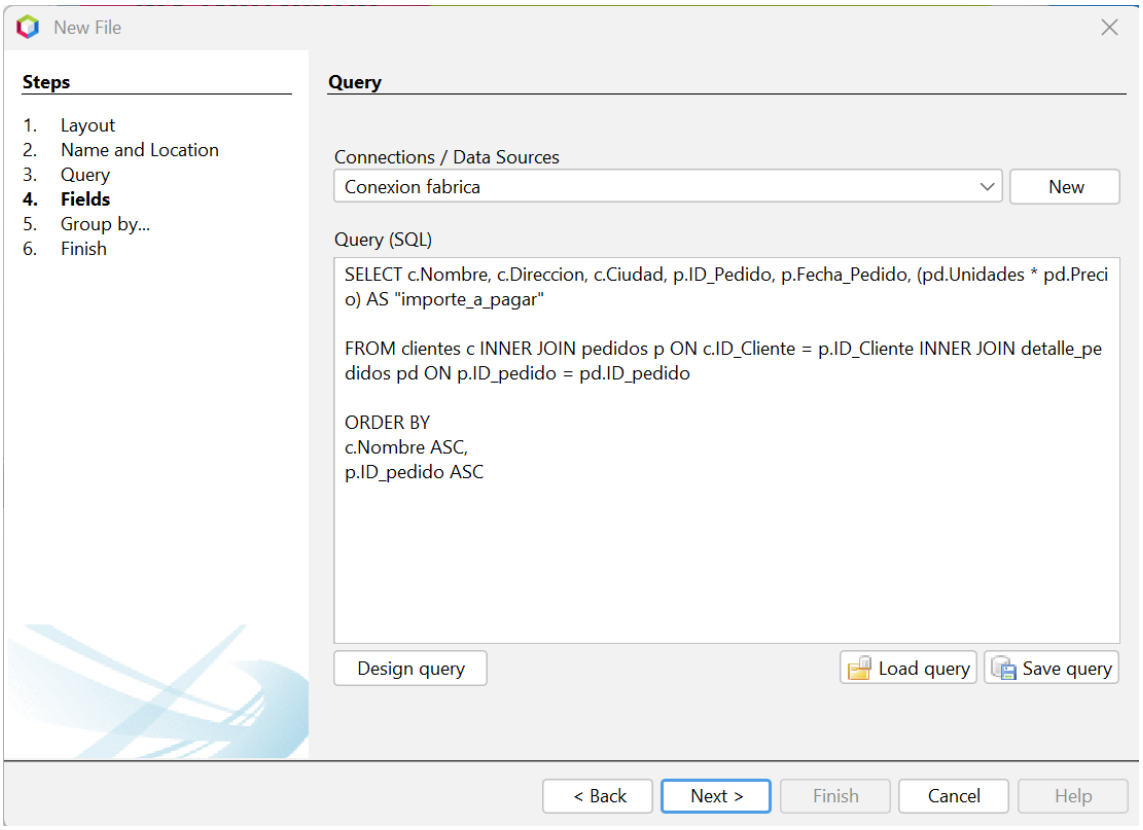
Para realizar este ejercicio será necesario crear un report Wizard. Para este caso, seleccionaremos un informe en blanco en formato A4:



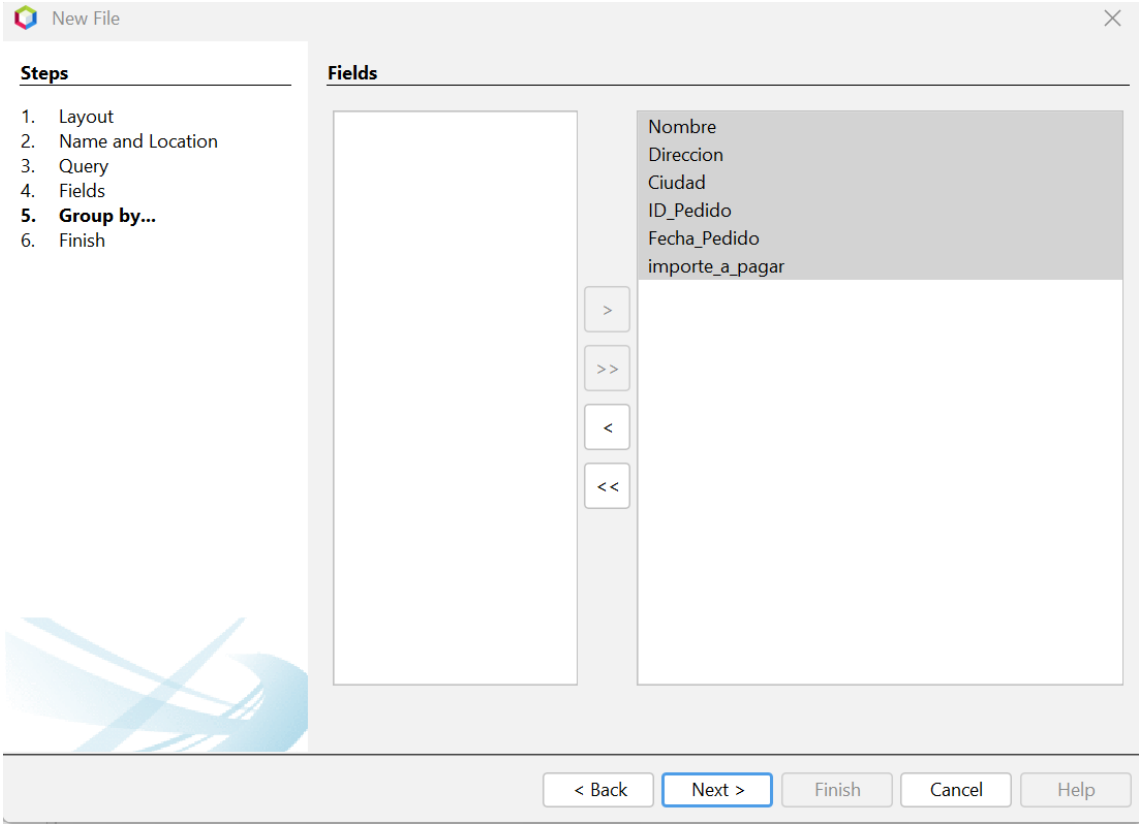
Establecemos el nombre de nuestro documento .jrxml como “factura”:



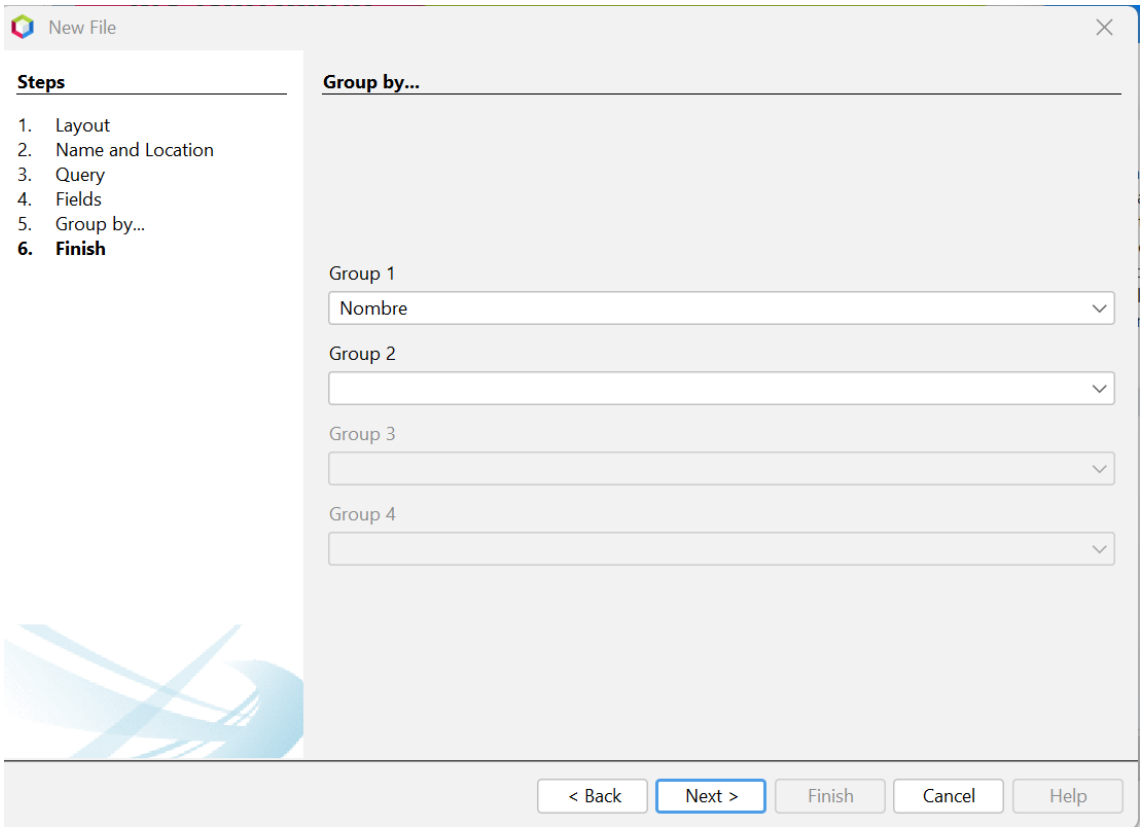
A continuación, seleccionamos la conexión creada anteriormente (conexión fabrica), y creamos el Query SQL incluyendo los datos con los que necesitamos trabajar:



Seguidamente pasamos los campos a consultar en la columna derecha:

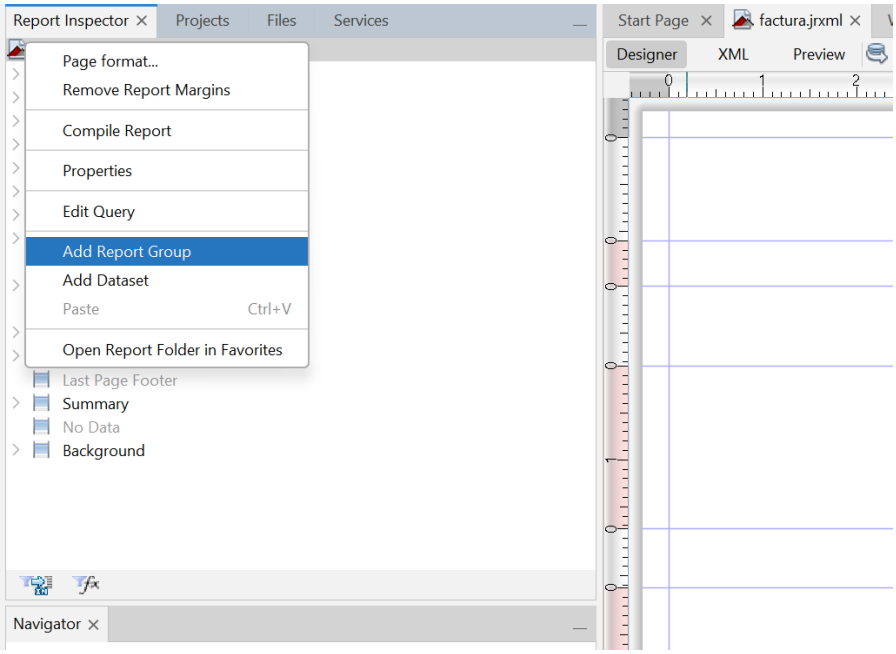


A continuación agrupamos por el nombre del cliente, en este caso queremos un informe en el que aparececan por cada empresa, su nombre, dirección y ciudad y sus pedidos. Lo ideal es agrupar por la clave primarias (ID_Cliente) pero en este caso he agrupado por nombre porque todas las empresas tienen nombres distintos:

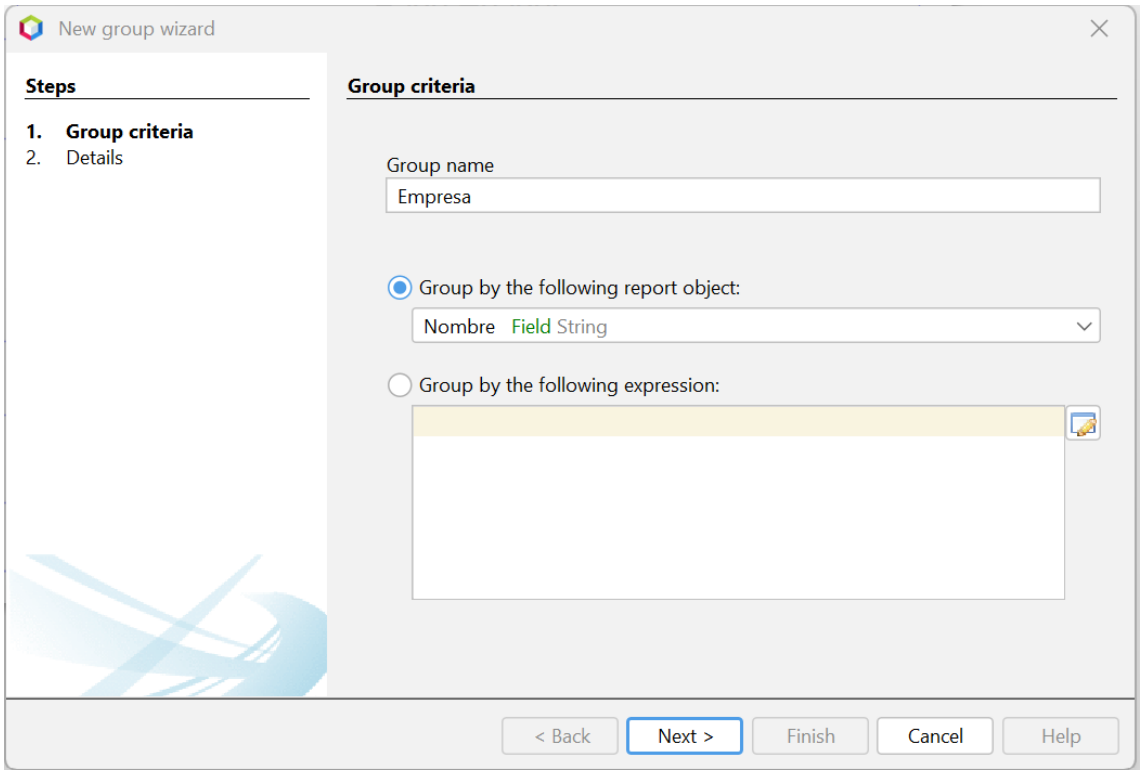


A continuación le damos a finalizar y ya tendríamos nuestro archivo factura.jrxml creado, pero con la plantilla en blanco que pasamos a modificar ahora:

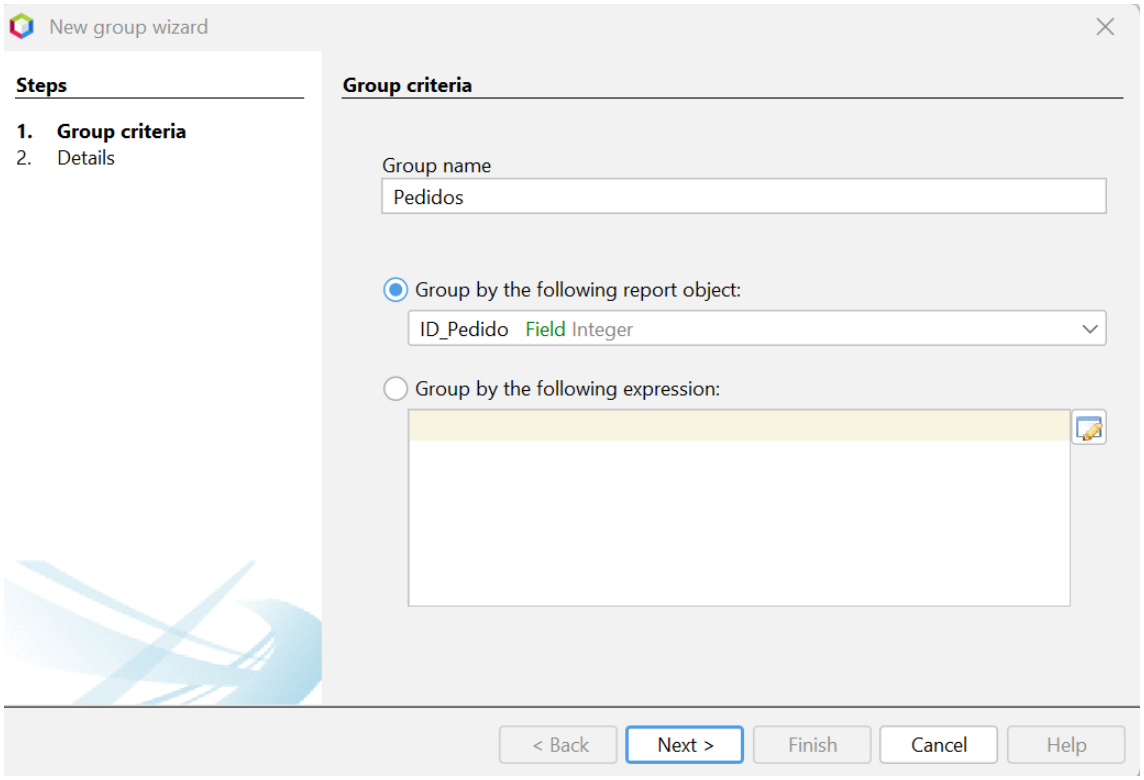
Para crear nuestro informe, en primer lugar, crearemos los grupos de reporte de empresa y pedido:



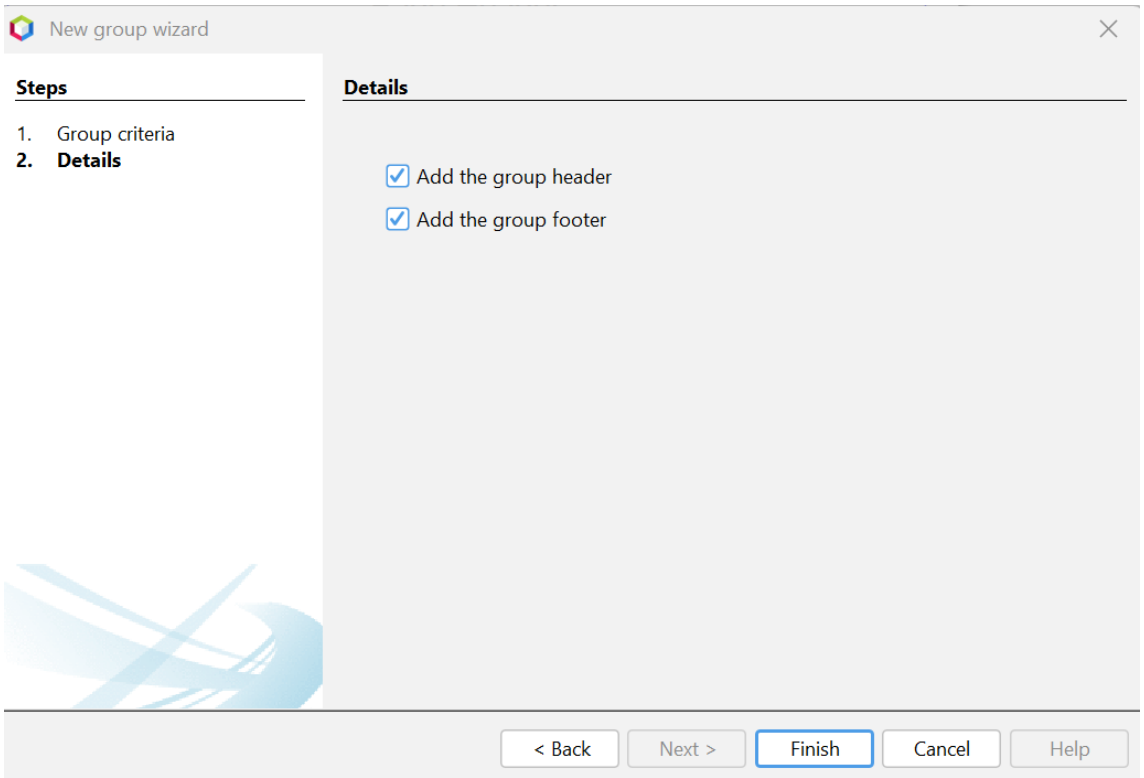
En el grupo empresa agrupamos por Nombre:



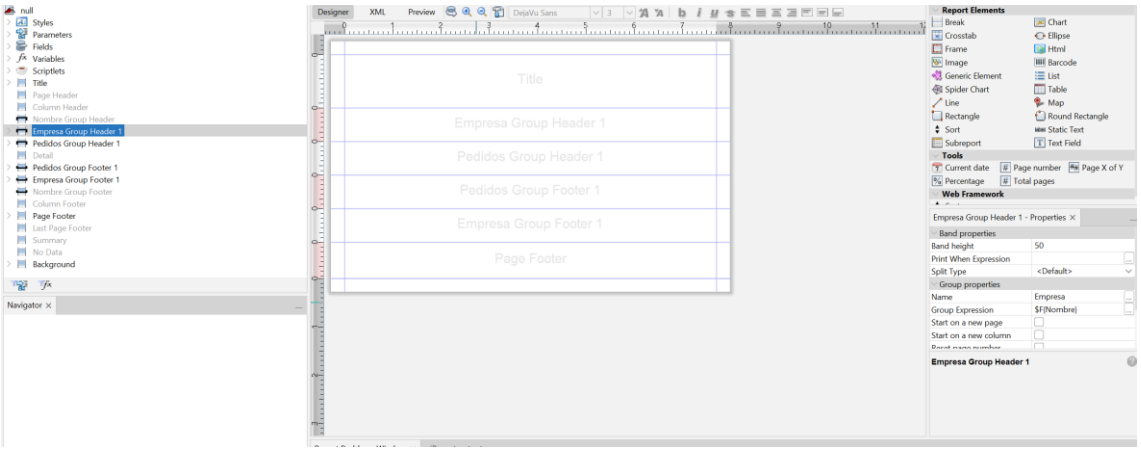
En el grupo Pedidos agrupamos por ID_Pedido:



Añadimos en ambos grupos tanto la cabecera como el pie de página:

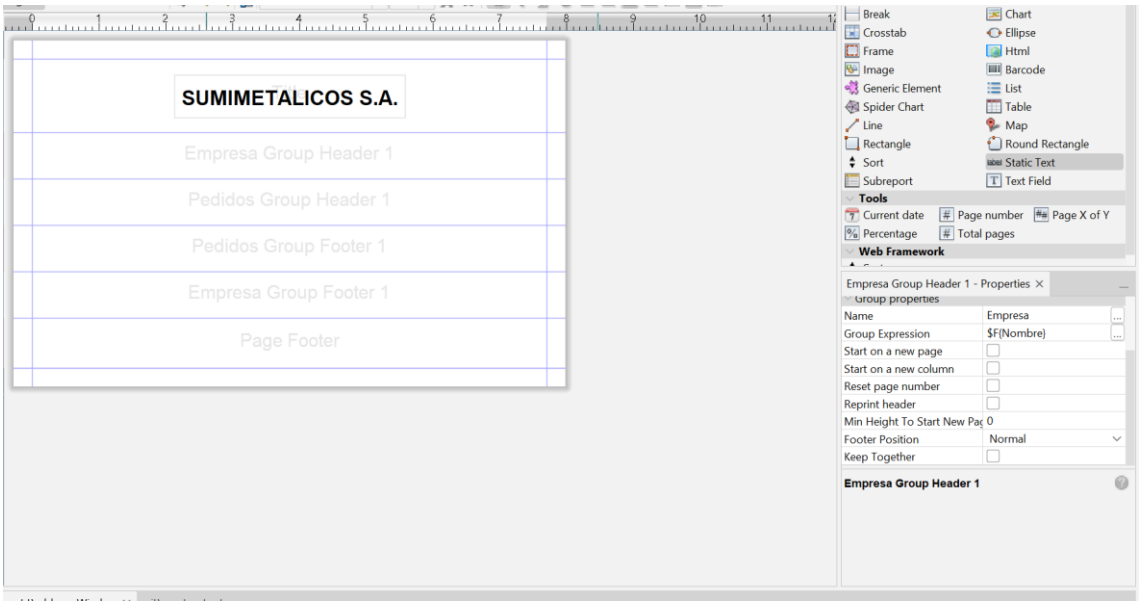


Eliminamos del diseño los campos que no vamos a utilizar y nos quedamos con el Título, el encabezado y pie de página de los grupos Empresa y Pedidos, y el pie de página:

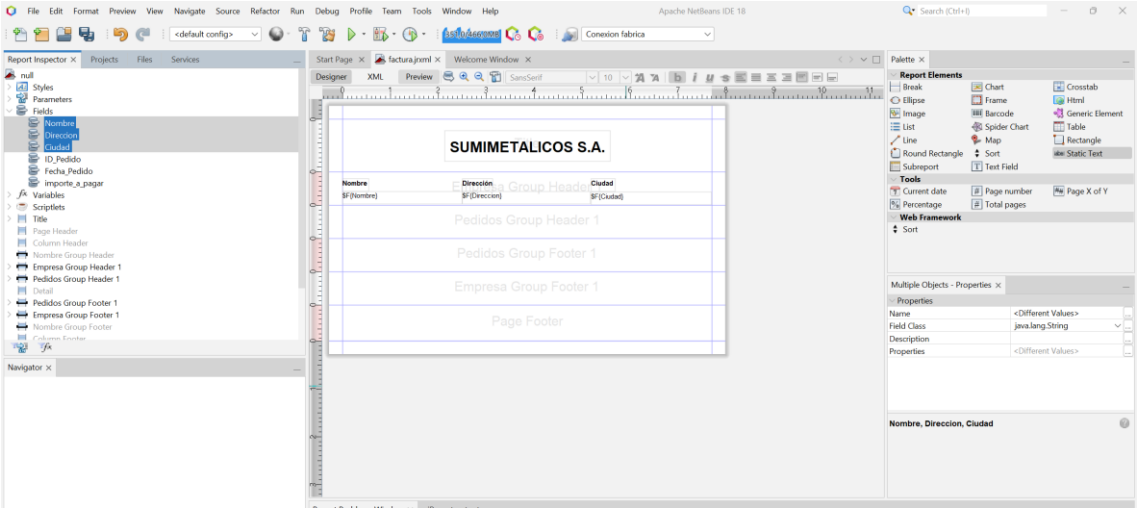


Una vez establecidos los campos, pasamos a ncluir los datos que se mostrarán en el informe:

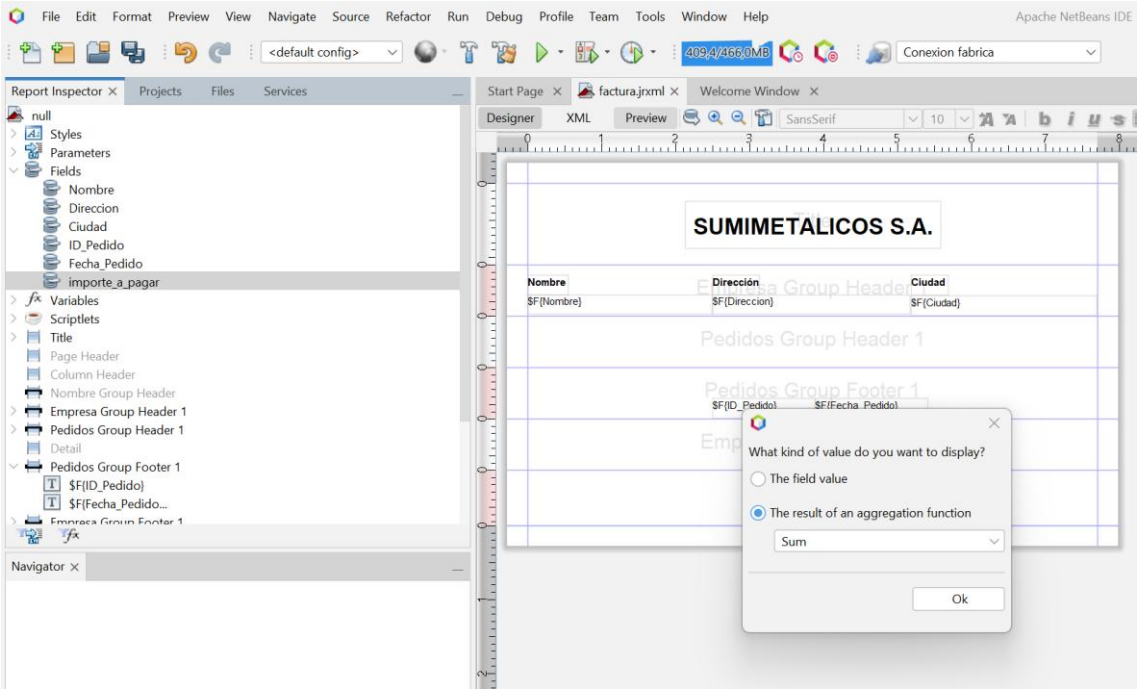
- Título→ Incluimos un Static Text con el nombre de la empresa: “SUMIMETALICOS S.A.”



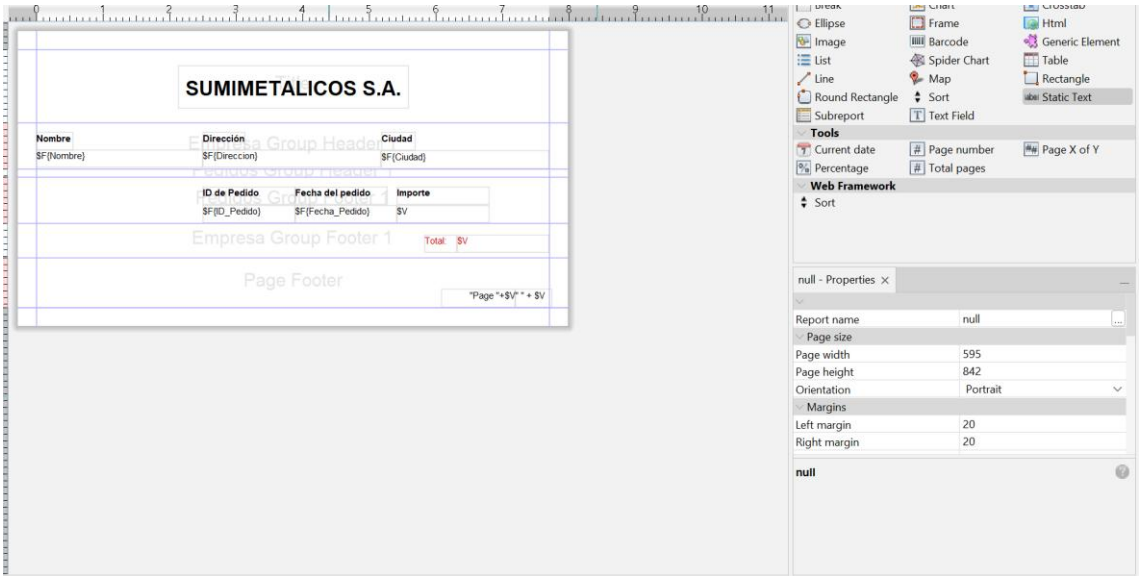
- **Empresa Group Header 1 →** Añadimos los datos de los clientes: Nombre, Dirección y Ciudad.



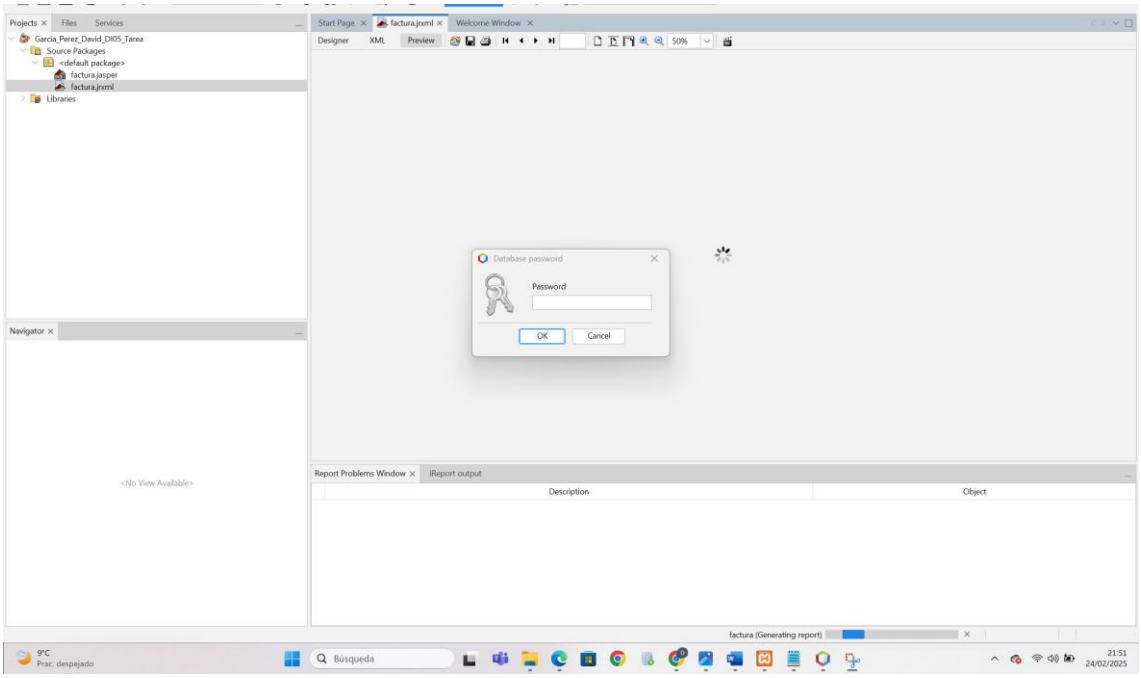
- **Pedidos Group Footer 1 →** Añadimos la información de cada pedido. Los campos: ID_Pedido, Fecha_Pedido y la suma de importe_a_pagar(SUM):

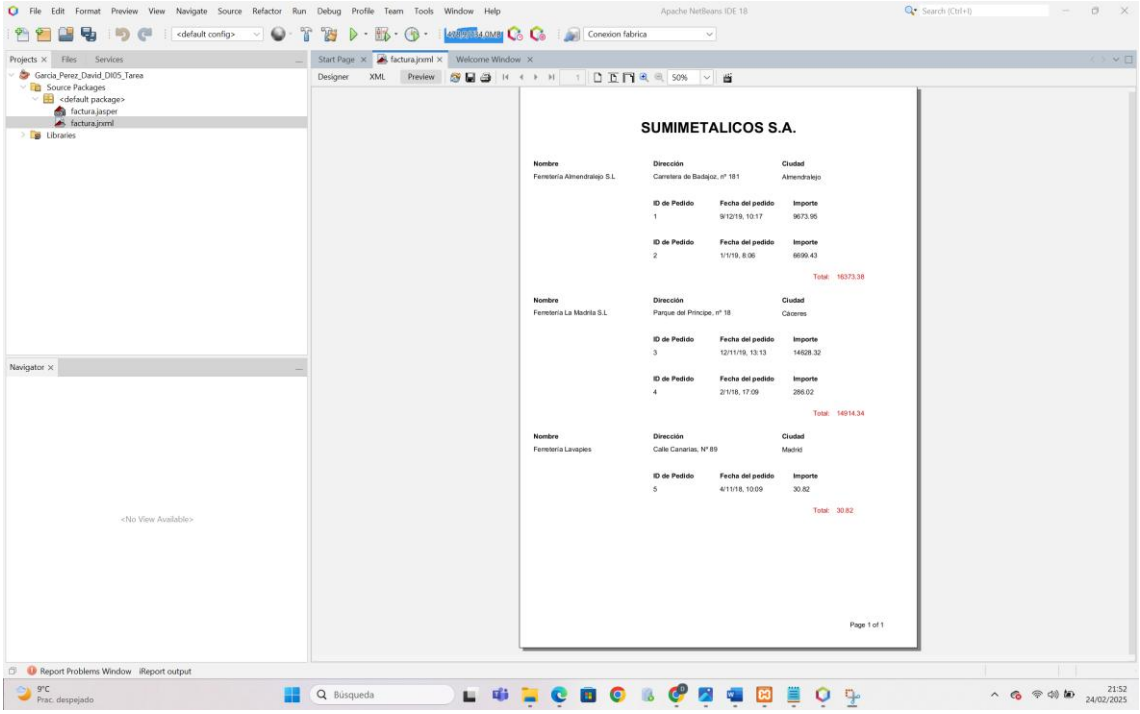


- **Empresa Group Footer 1 →** Añadimos los importes totales a pagar del total de pedidos de cada empresa, para ello añadimos el sumatorio del campo importe_a_pagar(SUM):

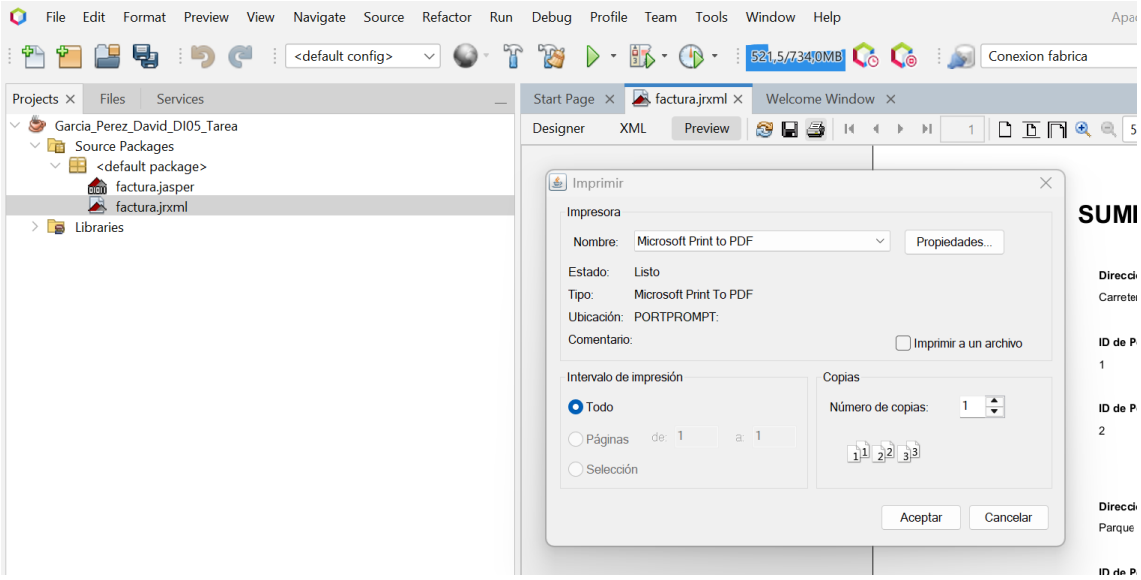


Para ver una vista previa del informe damos a preview:



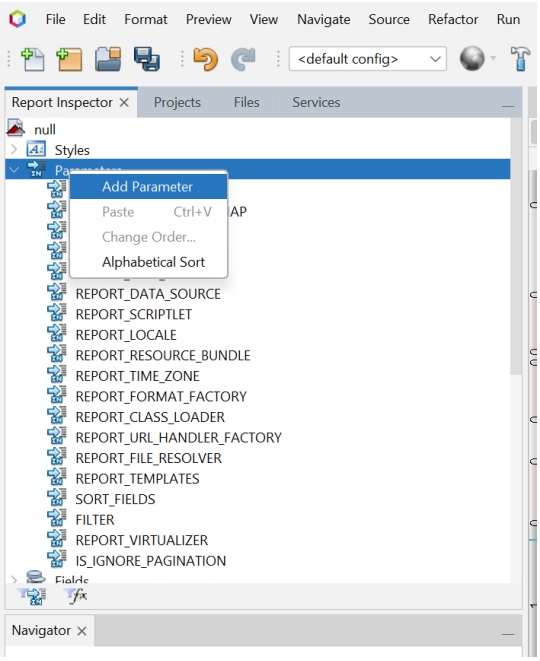


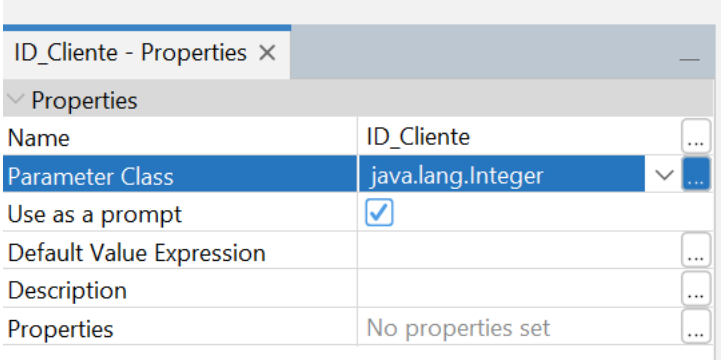
Para descargar el PDF, pulsamos en imprimir y lo guardamos con el nombre que queramos:



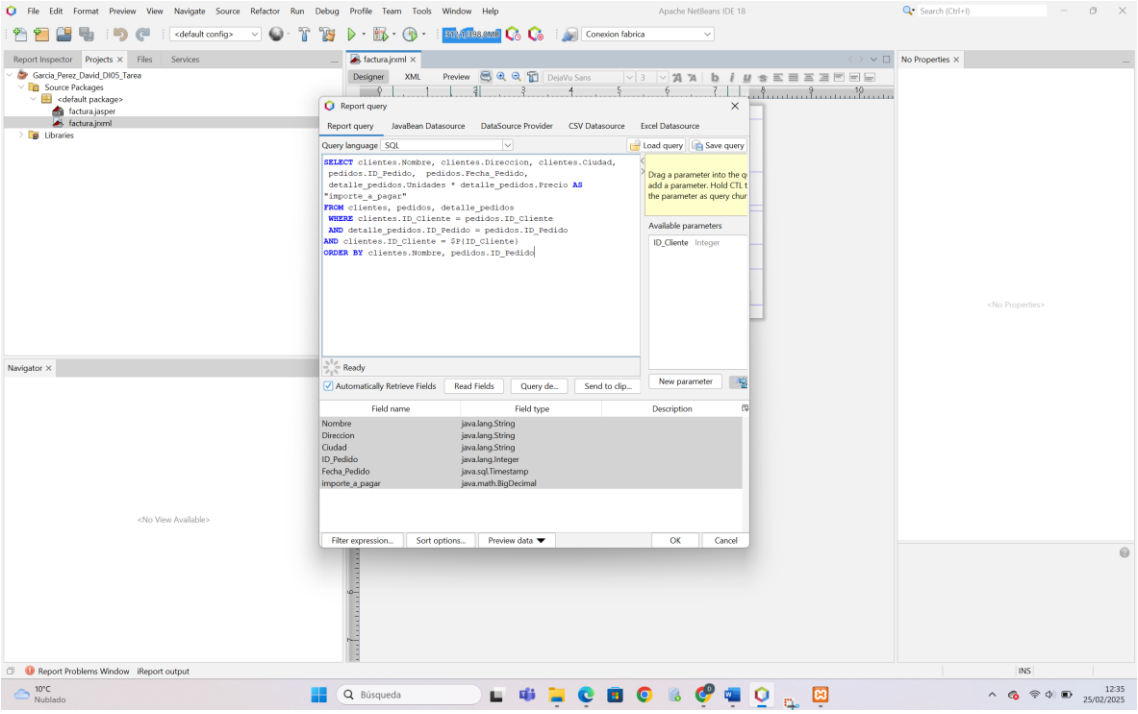
2. **Incluir el informe para generar facturas en un aplicación java que lo muestre en formato PDF, teniendo en cuenta que el código del cliente (ID_Cliente) se pasa al informe como parámetro. Por lo tanto, al ejecutarse la aplicación se generará el informe creado en el apartado anterior pero mostrando solo los datos para el Identificador de cliente (Id_cliente) pasado por parámetro.**

Este ejercicio requiere de una consulta con un parámetro ID_Cliente. Por lo que, en primer lugar procedo a crear este parámetro:

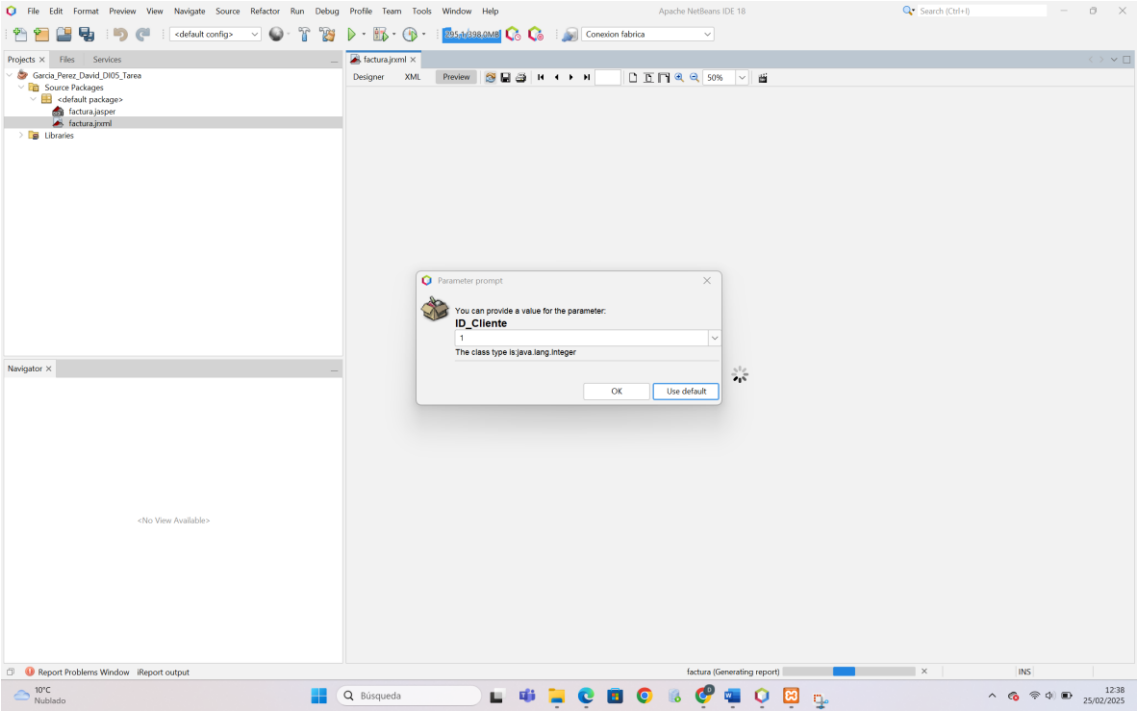


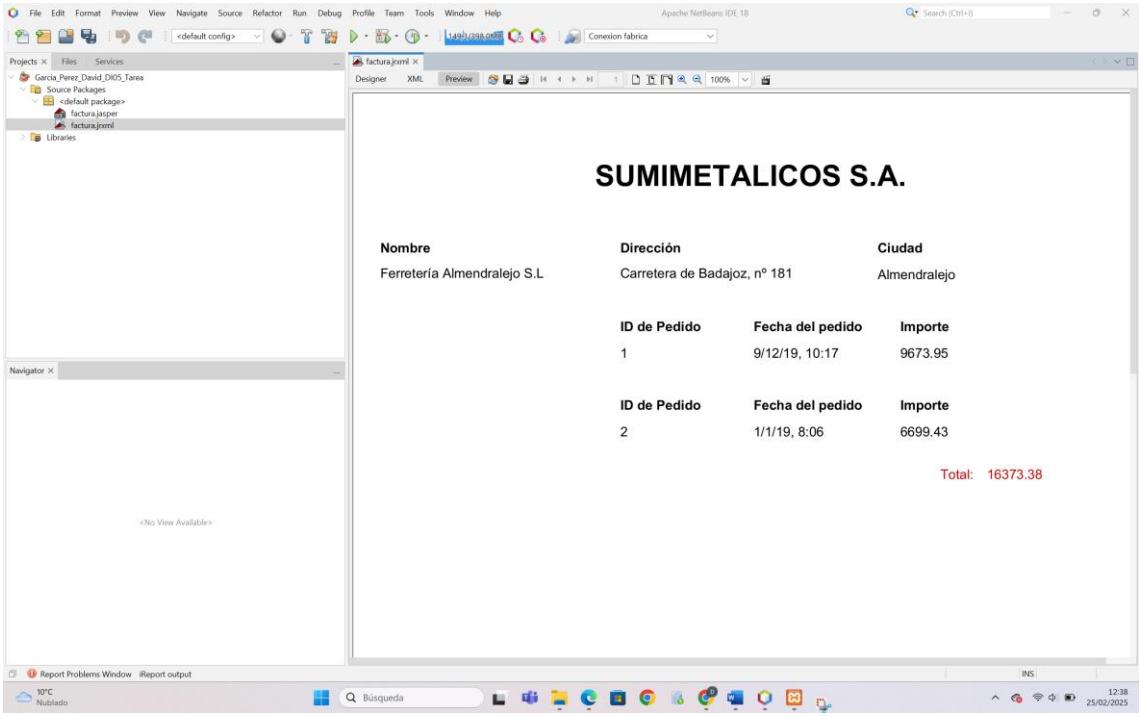


Después, modifico la consulta para asignar ID_Cliente = $P{ID_Cliente}, que será el campo pasado por parámetro.



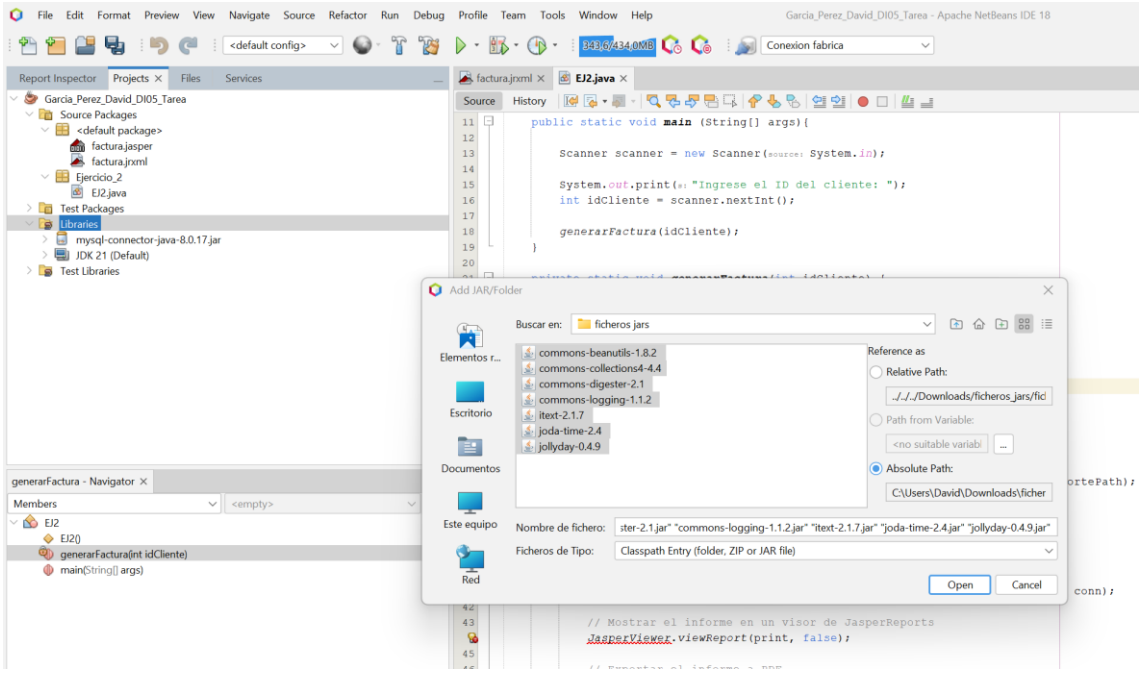
Ahora, al pulsar preview, se nos pedirá el ID_Cliente que queremos pasar como parámetro para mostrar sus datos:





A continuación procedemos a crear una aplicación sencilla en la que al pasar el ID_Cliente por parámetro genere el archivo PDF del informe.

Para ello creamos en un nuevo paquete la clase EJ2. Antes de nada debemos importar las librerías de mysql y jasperReport:

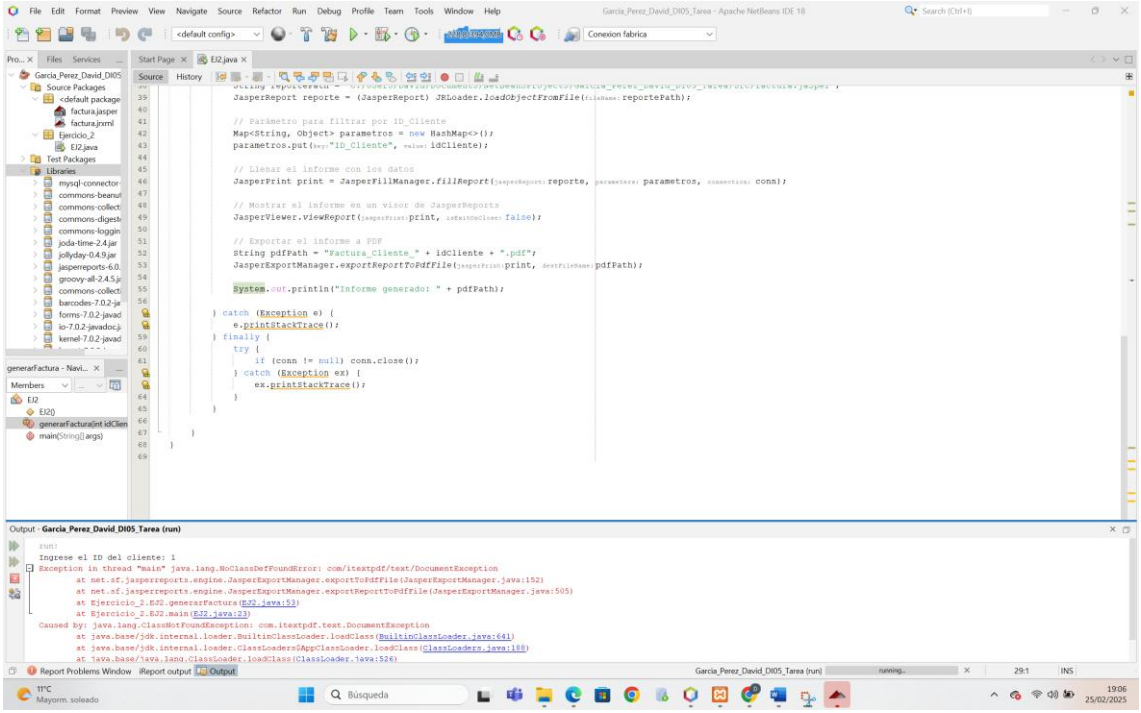


En esta nueva carpeta creamos la clase java:

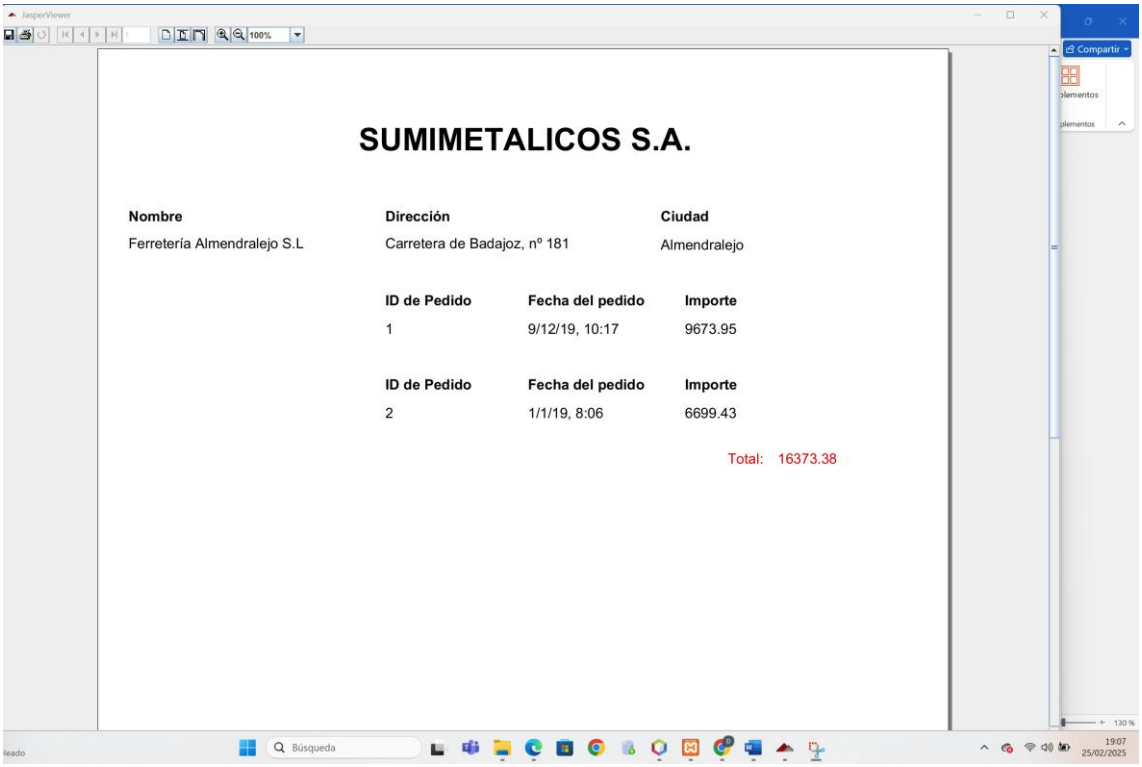
```java
package Ejercicio_2;

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import java.sql.Connection;
import java.sql.DriverManager;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.JasperViewer;


public class EJ2 {

    public static void main (String[] args){


        Scanner scanner = new Scanner(source: System.in);

        System.out.print(s: "Ingrese el ID del cliente: ");
        int idCliente = scanner.nextInt();

        generarFactura(idCliente);
    }

    private static void generarFactura(int idCliente) {

        Connection conn = null;

        try {
            // Conectar a MySQL usando XAMPP
            String url = "jdbc:mysql://localhost:3306/fabrica";
            String usuario = "root";
            String clave = "";
            conn = DriverManager.getConnection(url, user: usuario, password: clave);

            // Cargar el archivo Jasper compilado
            String reportePath = "C:/Users/David/Documents/NetBeansProjects/Garcia_Perez_David_DI05_Tarea/src/factura.jasper";
            JasperReport reporte = (JasperReport) JRLoader.loadObjectFromFile(fileName: reportePath);

            // Parámetro para filtrar por ID_Cliente
            Map<String, Object> parametros = new HashMap<>();
            parametros.put(key: "ID_Cliente", value: idCliente);

            // Llenar el informe con los datos
            JasperPrint print = JasperFillManager.fillReport(jasperReport: reporte, parameters: parametros, connection: conn);
```

```java
            // Exportar el informe a PDF
            String pdfPath = "Factura_Cliente_" + idCliente + ".pdf";
            JasperExportManager.exportReportToPdfFile(jasperPrint: print, destFileName: pdfPath);

            System.out.println("Informe generado: " + pdfPath);

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                if (conn != null) conn.close();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }

    }
}
```

Al ejecutar este código e introducir el id del cliente que queremos mostrar se nos muestra el informe:

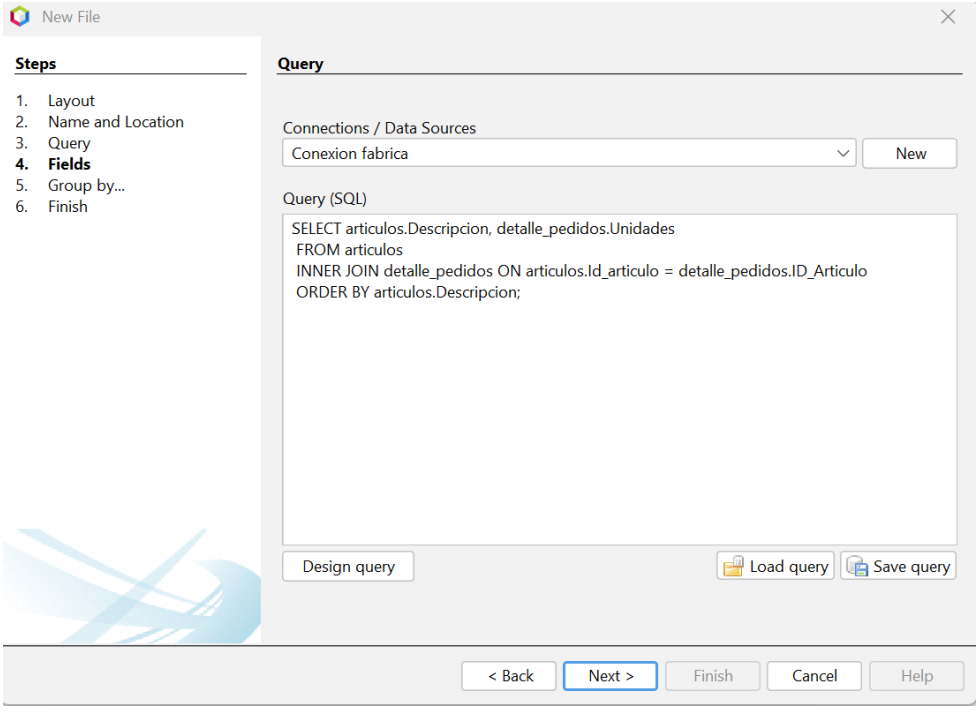


Surgen algunos errores debido a la incompatibilidad de algunas librerías por estar desactualizadas. Pero se nos muestra el informe:

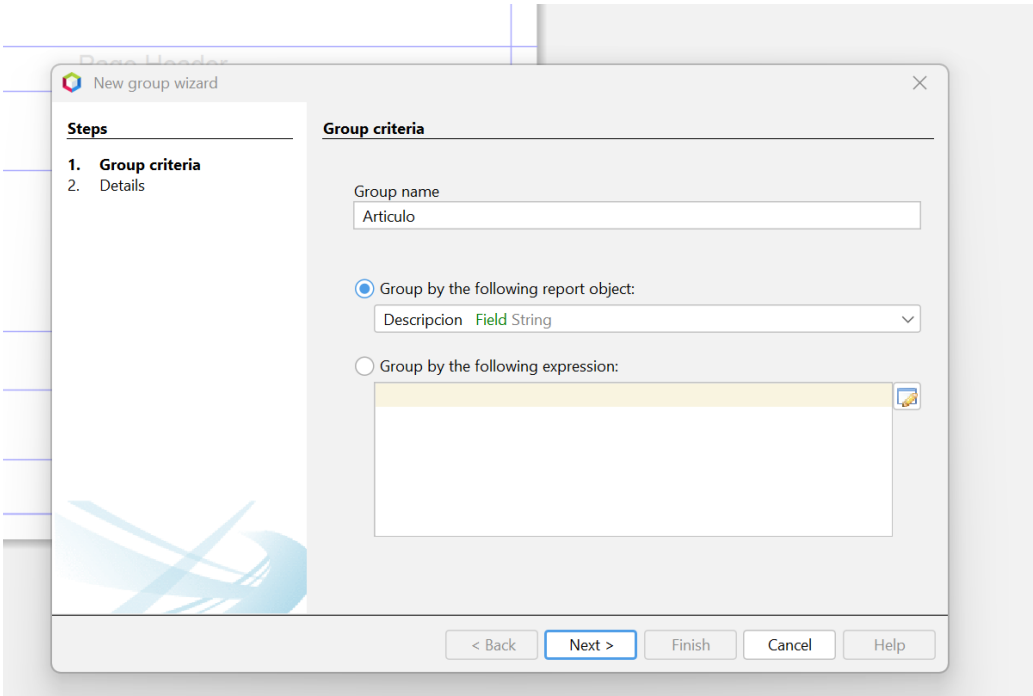


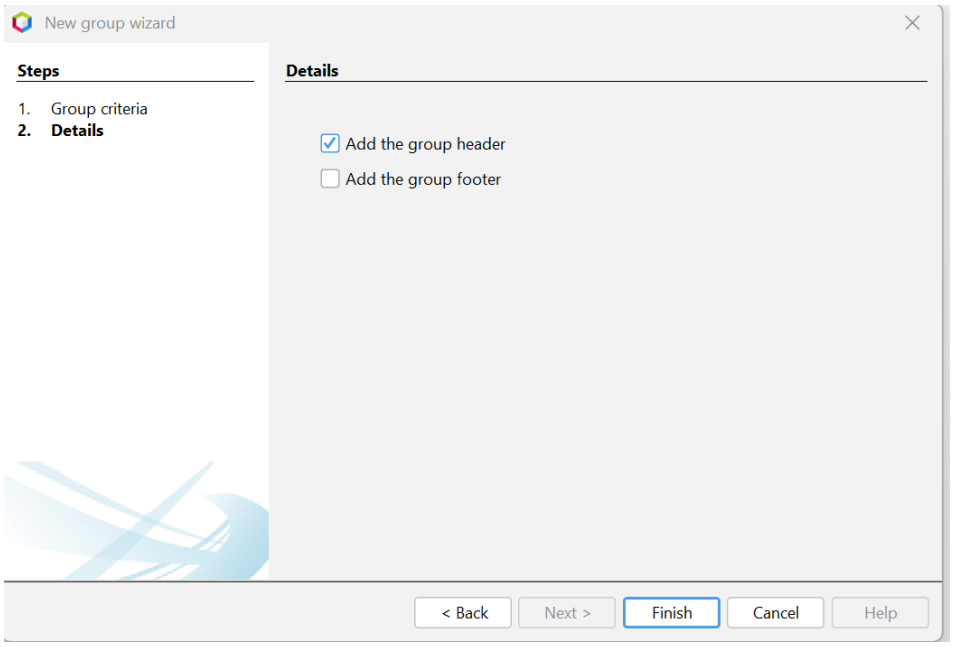
3. **Crear un informe en el que aparezca un listado con las ventas totales (cantidad total) de cada artículo (se visualiza su nombre). Por cada artículo vendido, debe de mostrarse la descripción del mismo junto con el número de unidades totales vendidas. Añadir un gráfico al informe que muestre por cada artículo vendido, el número total de unidades vendidas. Esta información la puedes obtener de la tabla detalle_pedidos.**

Creamos un nuevo reporte con la siguiente consulta:

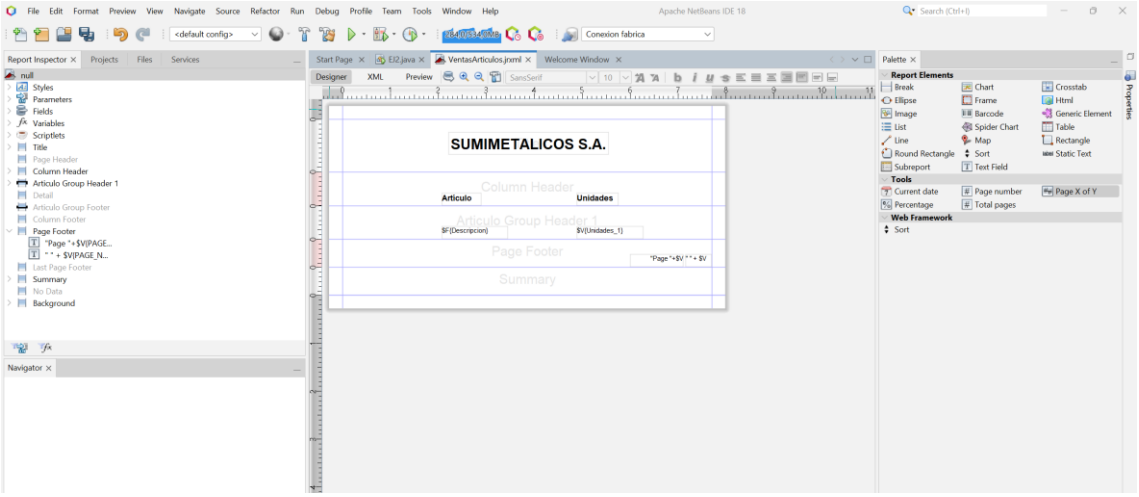


Creamos el grupo de reporte Articulo. Creamos solo el header y se agrupa por descripcion, que incluye el nombre del artículo:

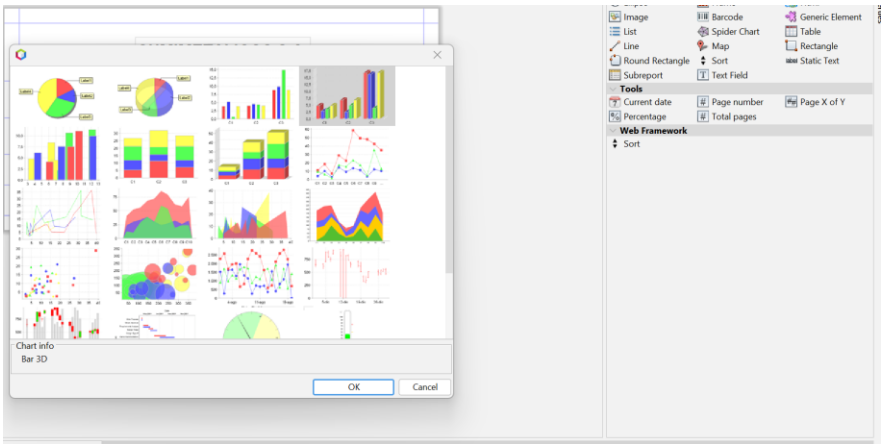


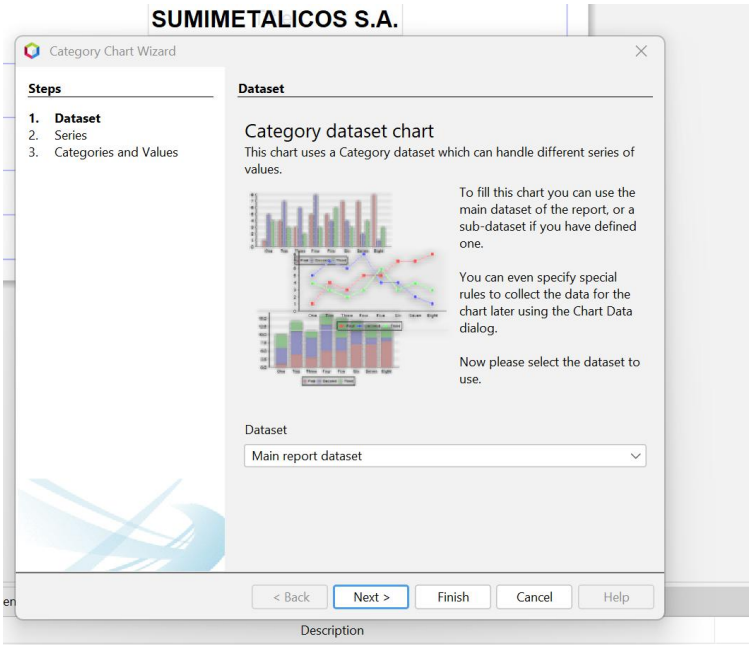


He eliminado el resto de bandas excepto el título, colum header, sumary para meter el gráfico y el footer. Añado al Column Header las static text, en Articulo Group header la descripcion y las unidades vendidas con la funcion SUM.

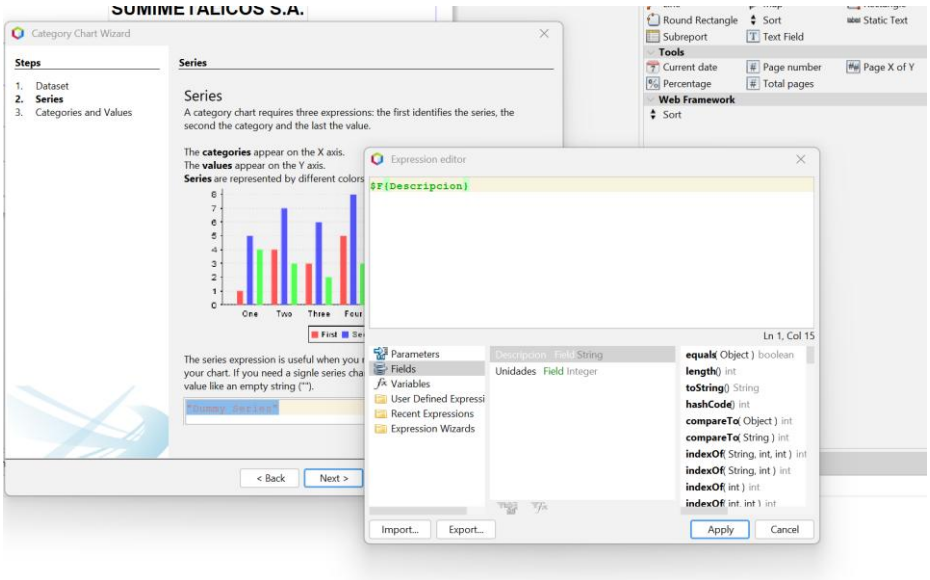


A continuación, introducimos el gráfico en Summary arrastrando el componente chart y selecciono el gráfico de barras en 3D y pulsamos OK:

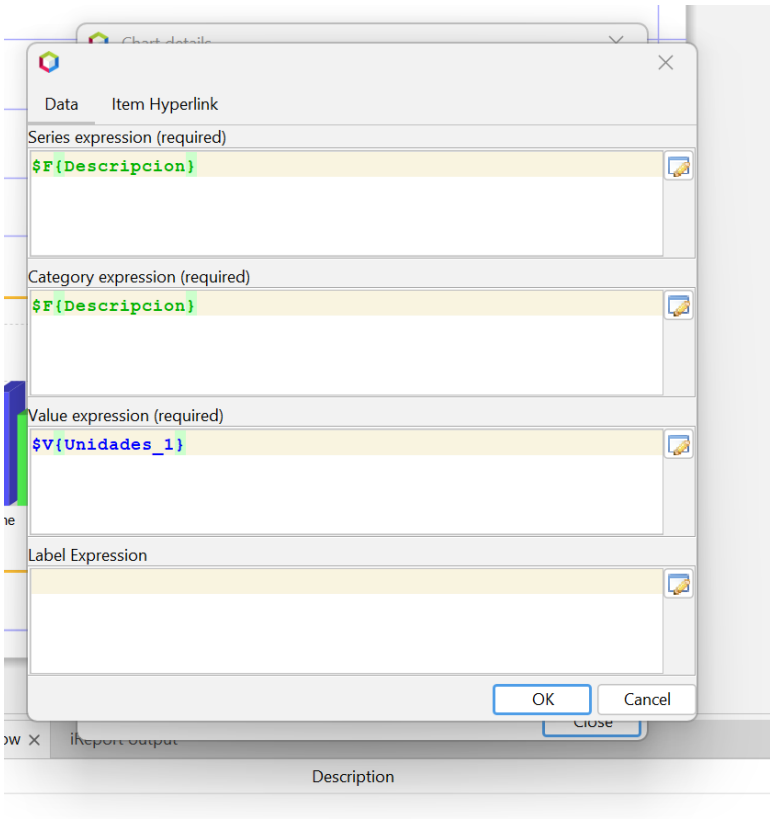




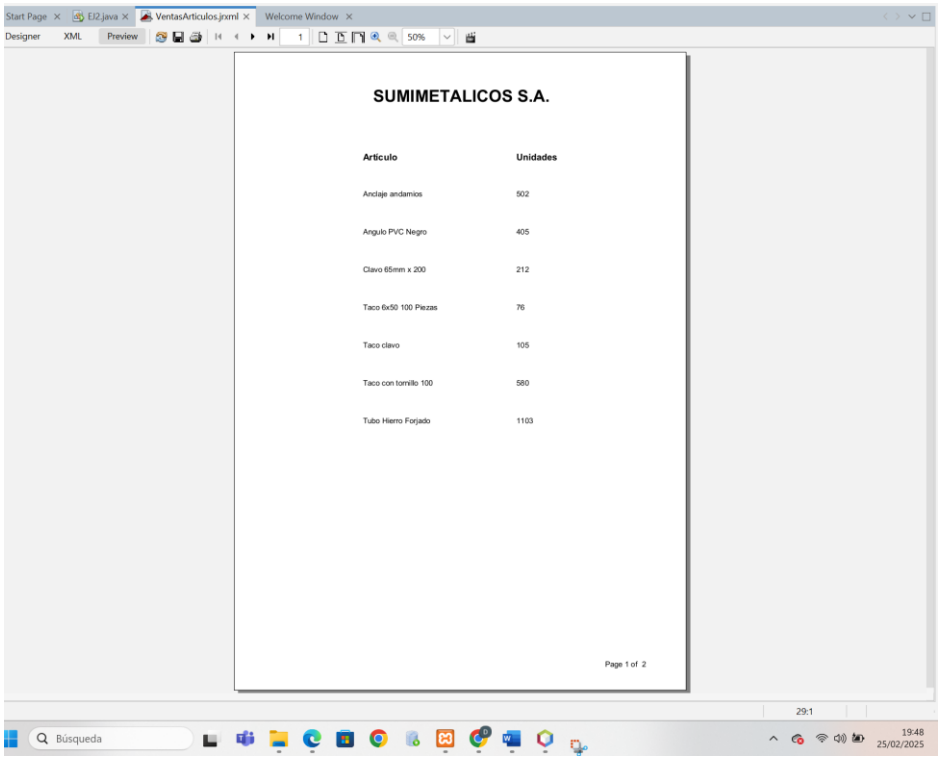
En el expresion editor le meto la descripción:

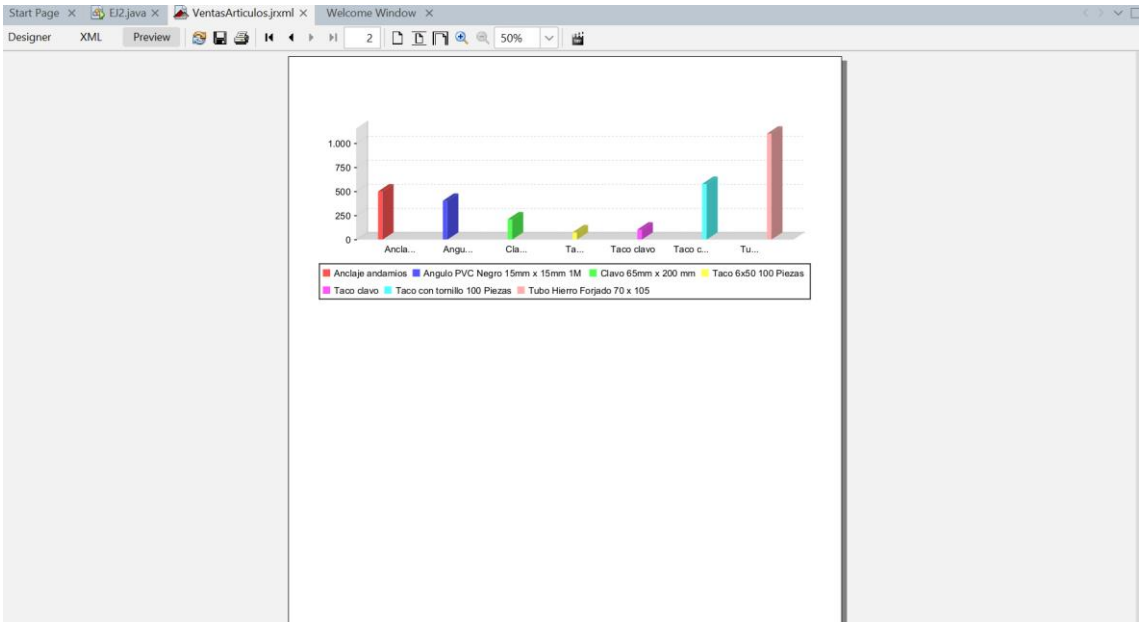


En la siguiente ventana descripción y unidades con la funcion sum (variable creada anteriormente al introducir el total de ventas en el informe):



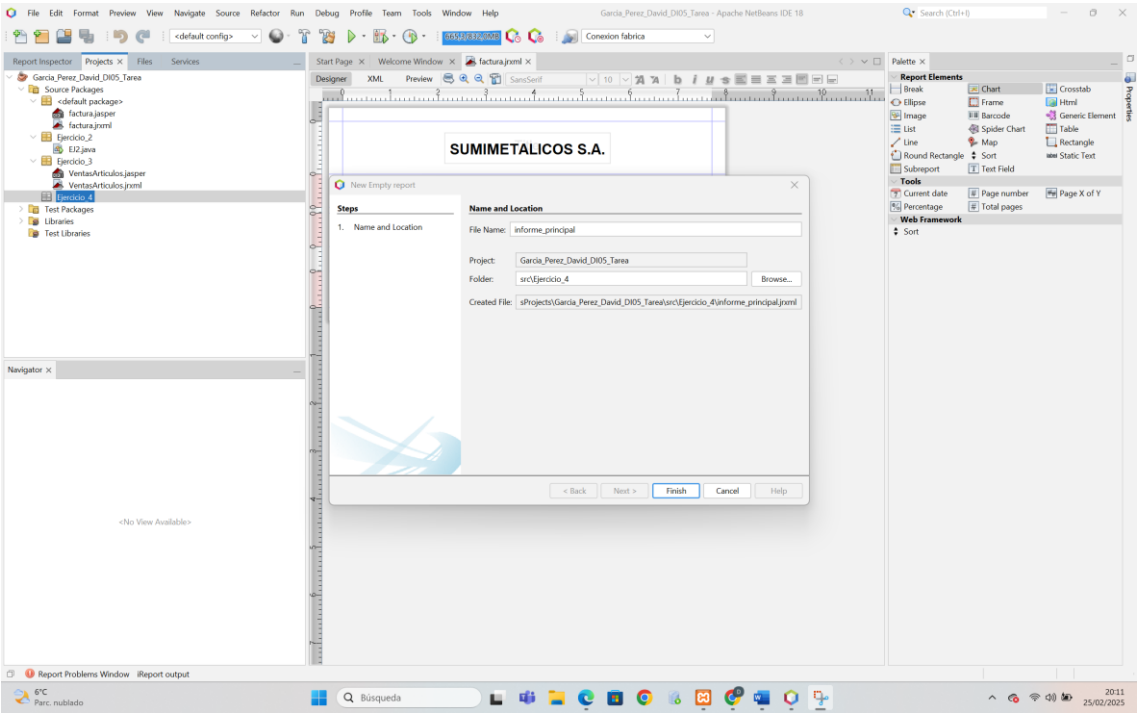
Vista del informe final:



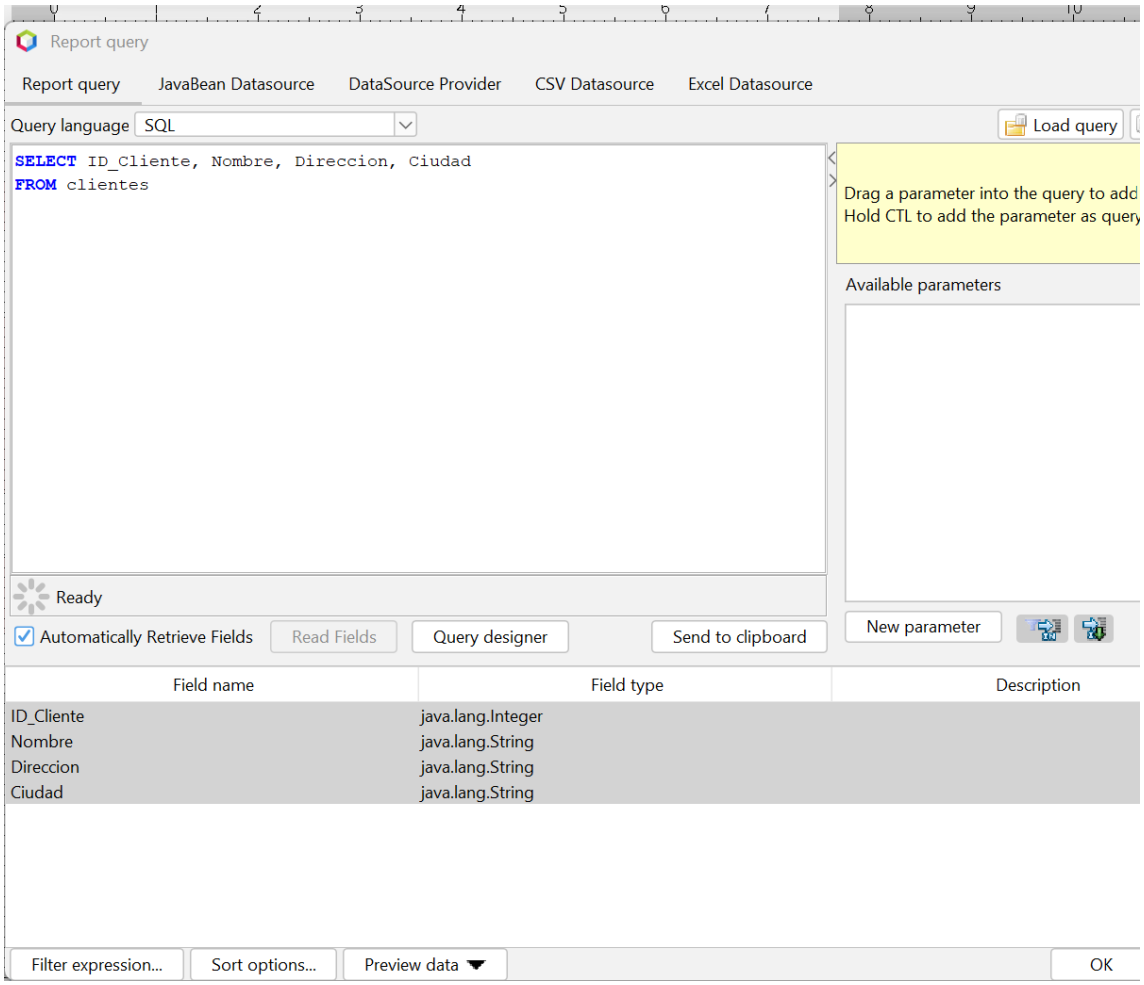


4. Repite el ejercicio 1 (sin subtotales) utilizando subinformes.

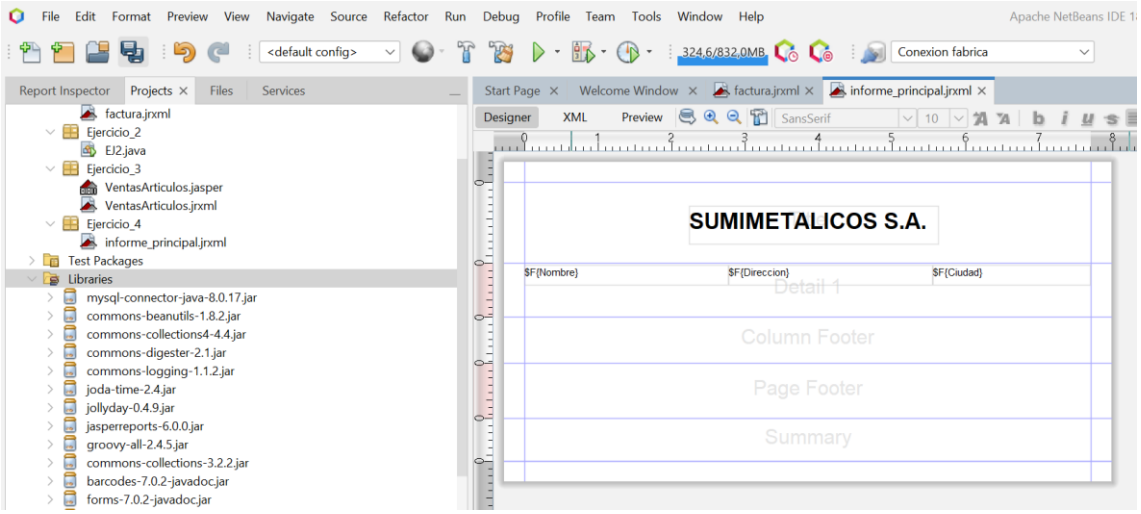
Lo primero ha sido crear un Empty Report (with Groovy):



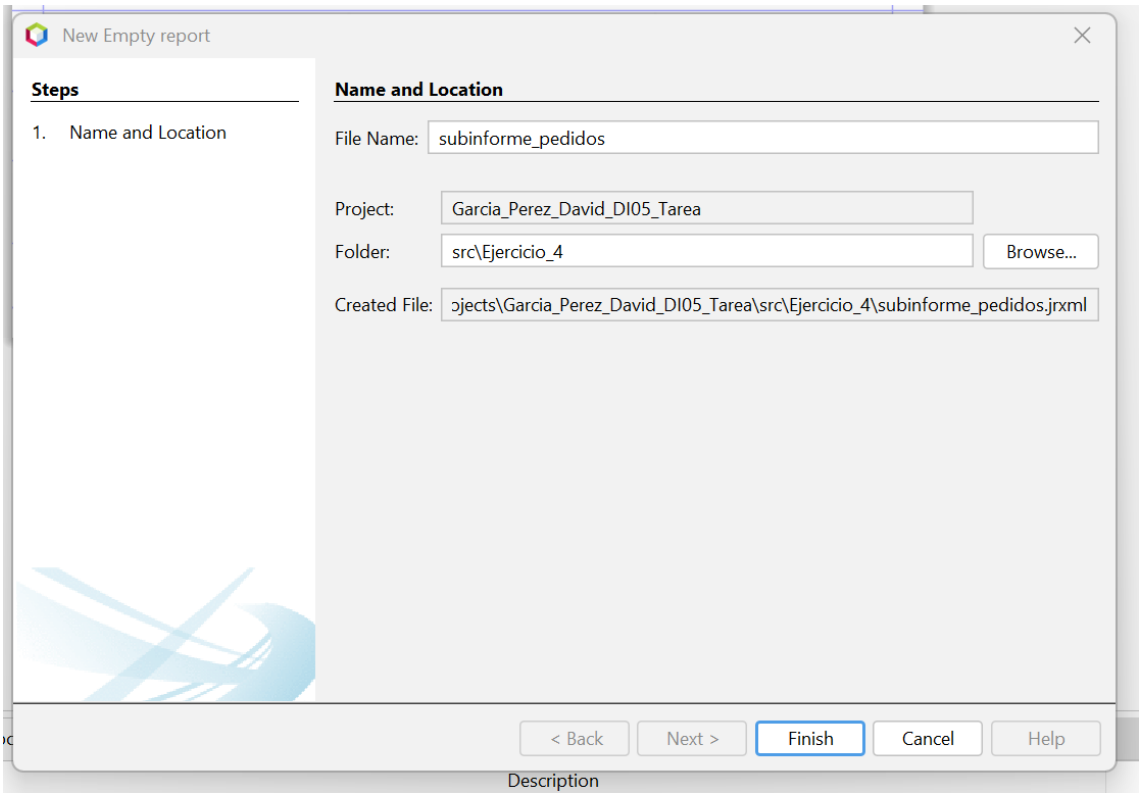
A continuación añadimos la consulta para seleccionar los campos de clientes:



Incluimos los campos de Nombre, Dirección y Ciudad en el apartado Detail 1:

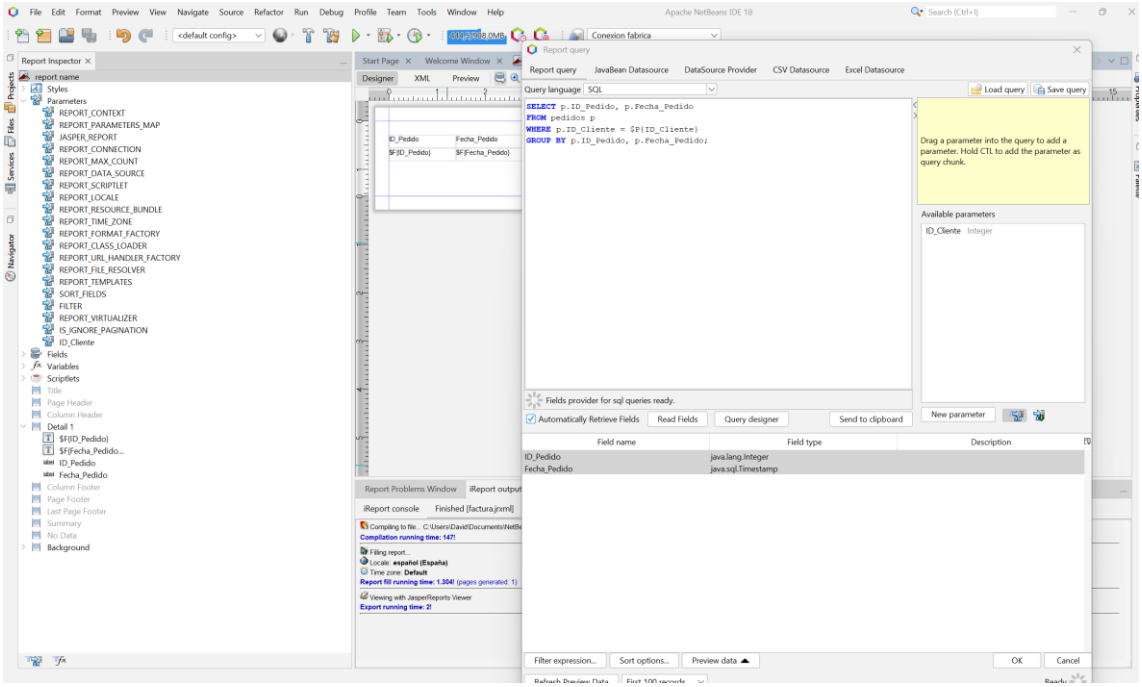


A continuación creo un Empty Report (subinforme_pedidos) para recuperar los datos de los pedidos:

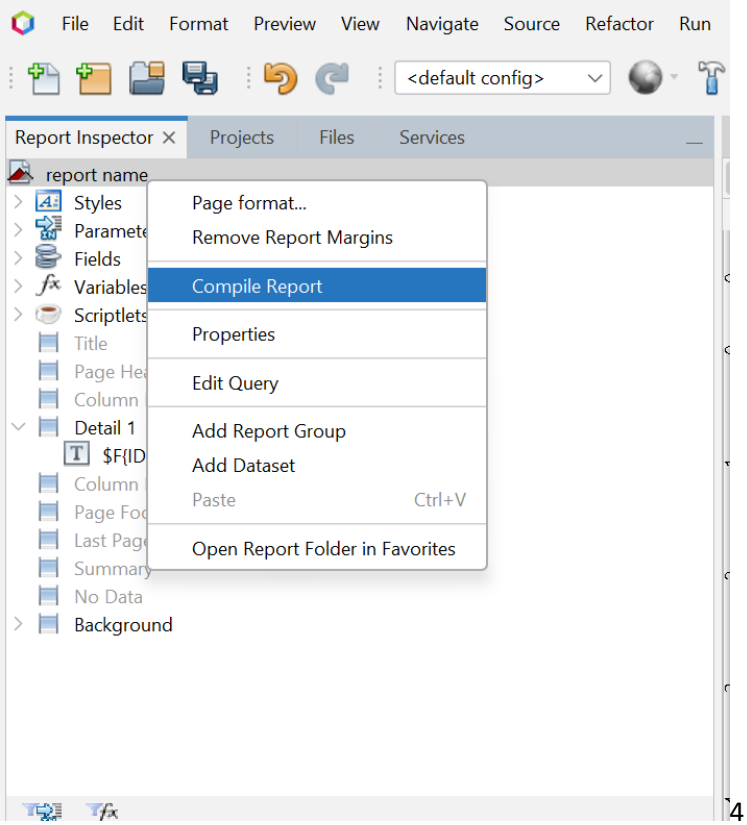


Creamos el informe con la consulta que nos proporciona los campos de ID y fecha del pedido, y creando el parámetro ID_Cliente, que nos servirá para asociar este subinforme al informe principal.

Añadimos los campos ID_pedido y G¡Fecha_Pedido en Detail 1:



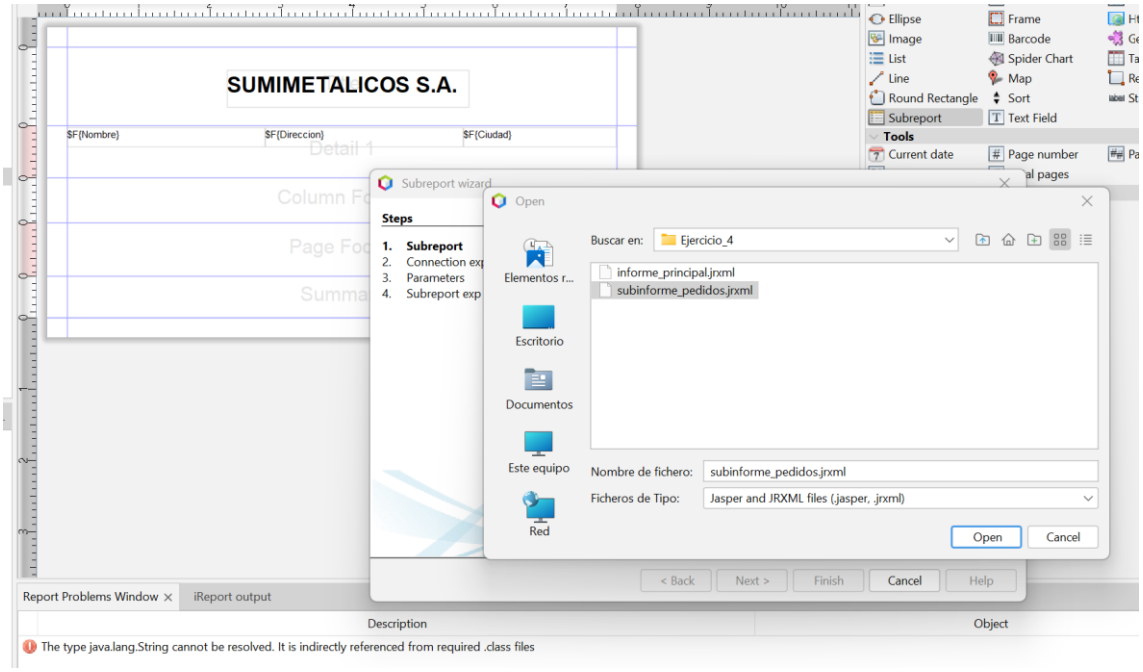
Compilamos el subinforme:

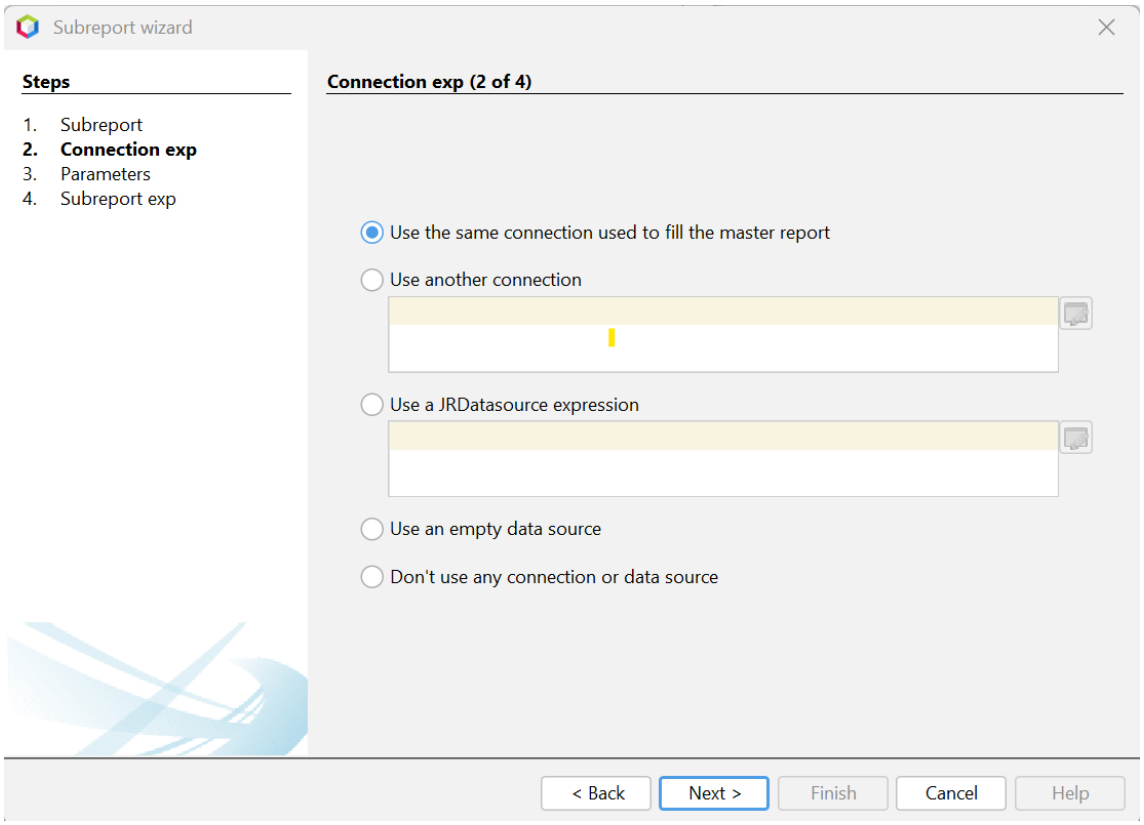


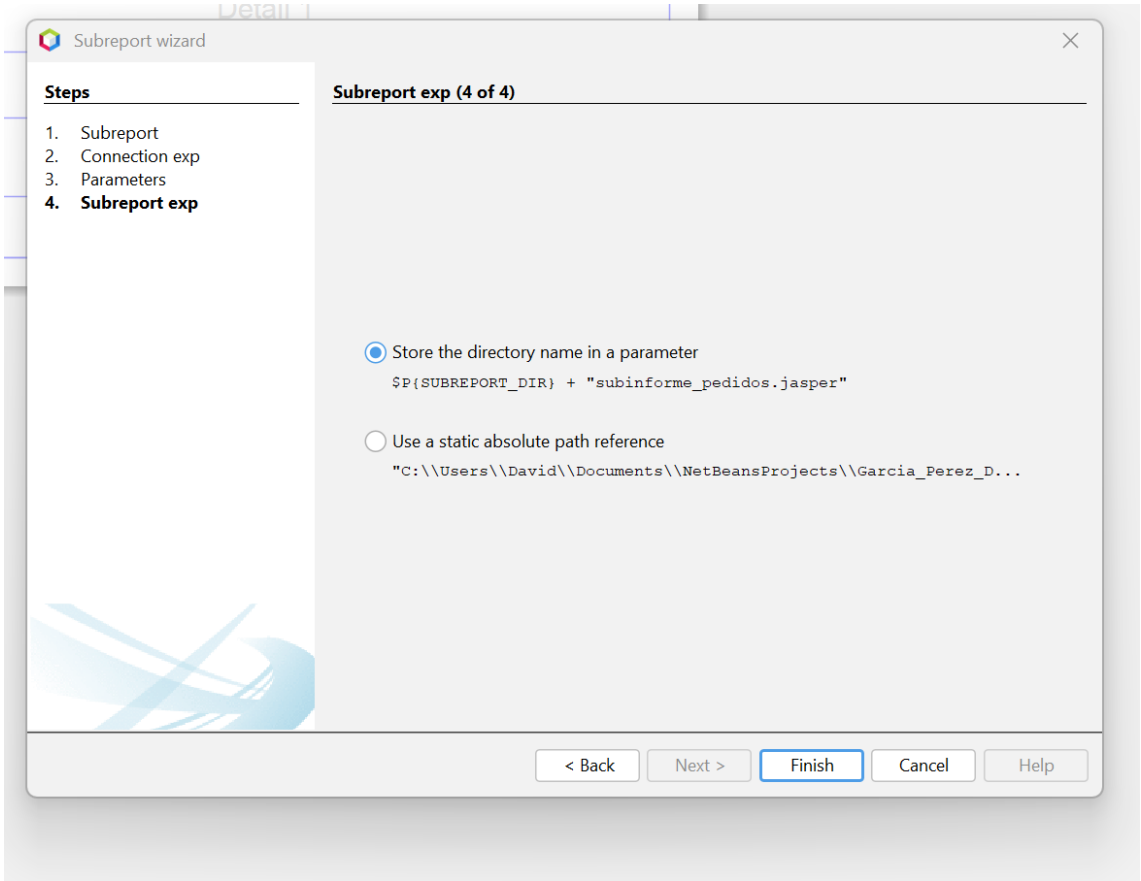
4

Procedemos a añadir en el informe principal el subinforme_pedidos, para ello arrastramos el elemento Subreport al apartado Detail 1 del informe principal.

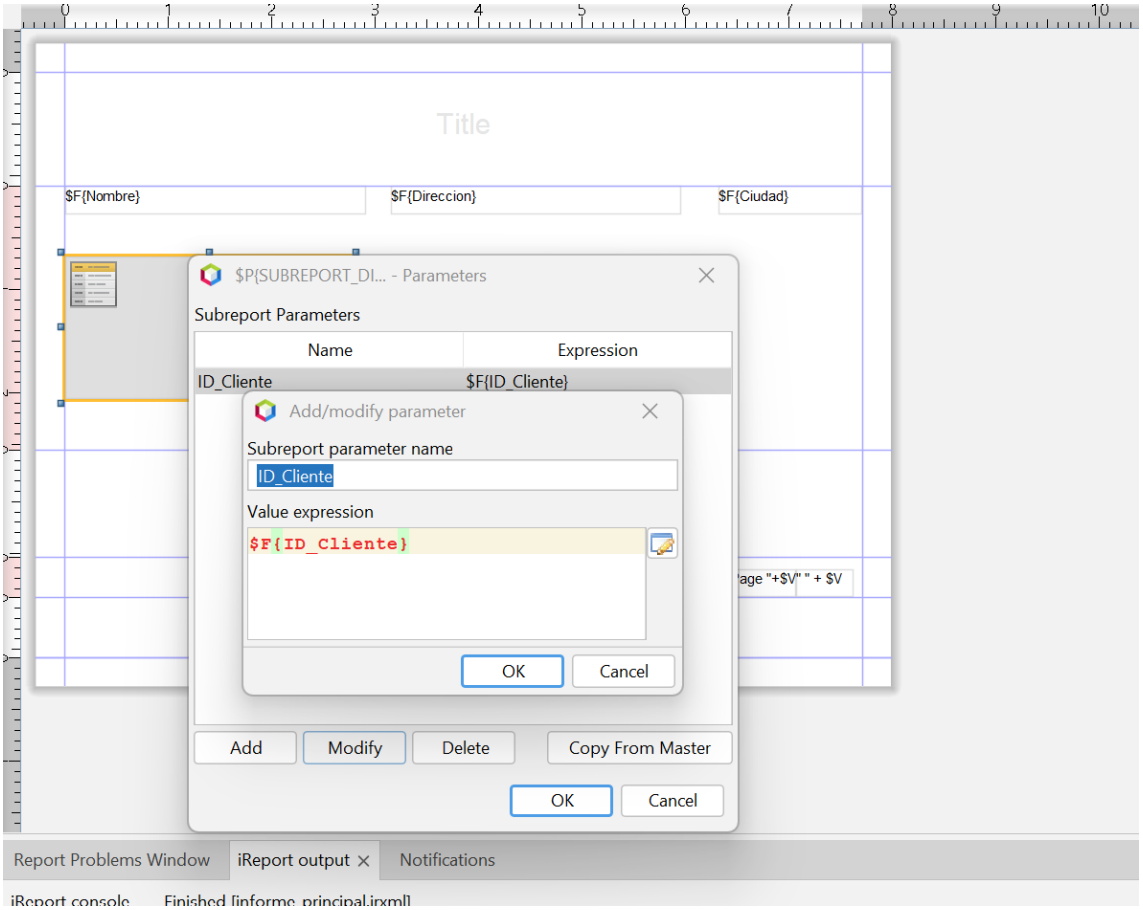
Seleccionamos “Use an existing report” y cargamos el archivo “subinforme_pedidos.jrxml”



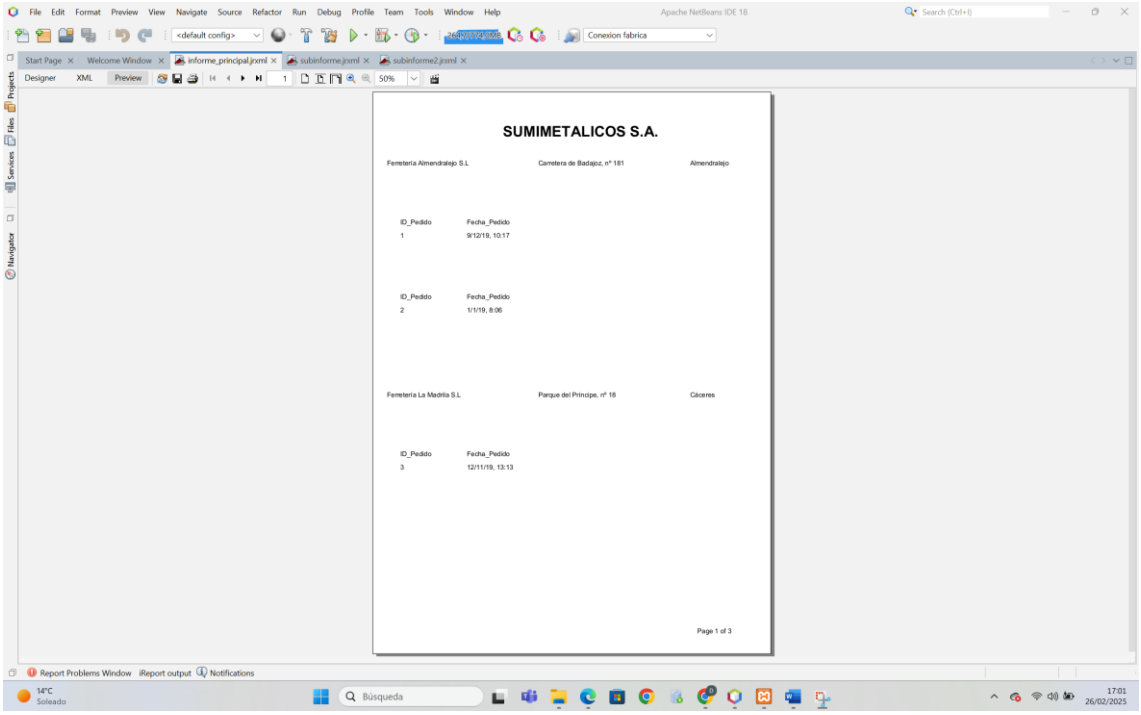

Subreport wizard
Steps
1. Subreport
2. Connection exp
3. Parameters
4. Subreport exp
Connection exp (2 of 4)
Use the same connection used to fill the master report
Use another connection
Use a JRDatasource expression
Use an empty data source
Don't use any connection or data source
< Back
Next >
Finish
Cancel
Help


Subreport wizard
Steps
1. Subreport
2. Connection exp
3. Parameters
4. Subreport exp
Subreport exp (4 of 4)
Store the directory name in a parameter
$P{SUBREPORT_DIR} + "subinforme_pedidos.jasper"
Use a static absolute path reference
"C:\\Users\\David\\Documents\\NetBeansProjects\\Garcia_Perez_D...
< Back
Next >
Finish
Cancel
Help

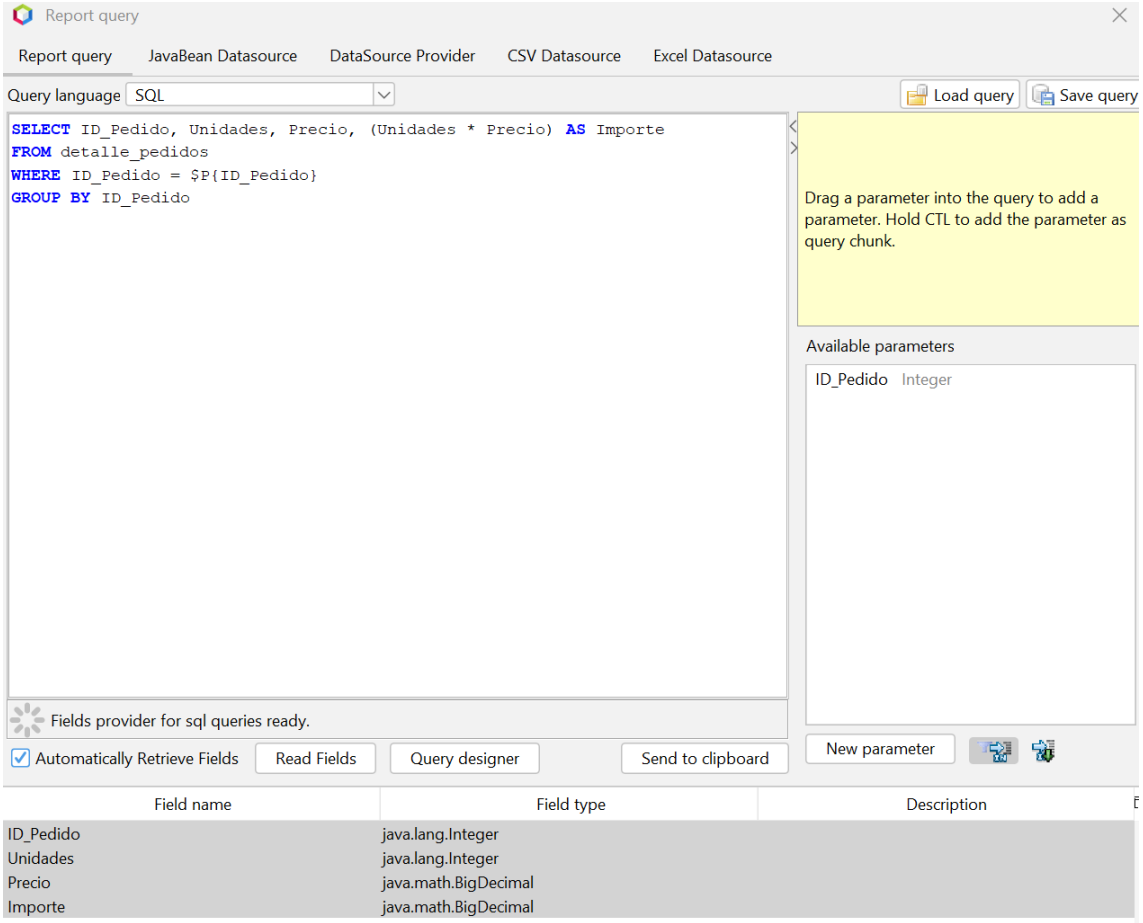
Seleccionamos el subreporte añadido e incluimos en este el parámetro ID_Cliente:



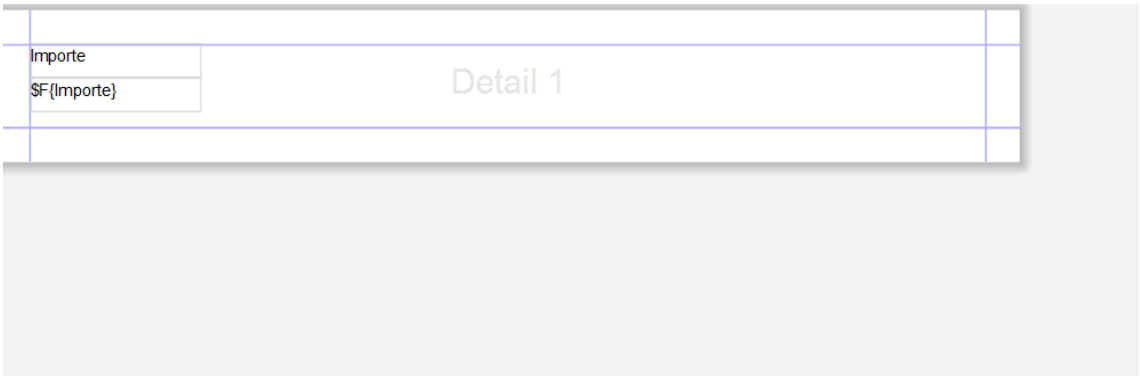
Así quedaría la vista previa de momento:



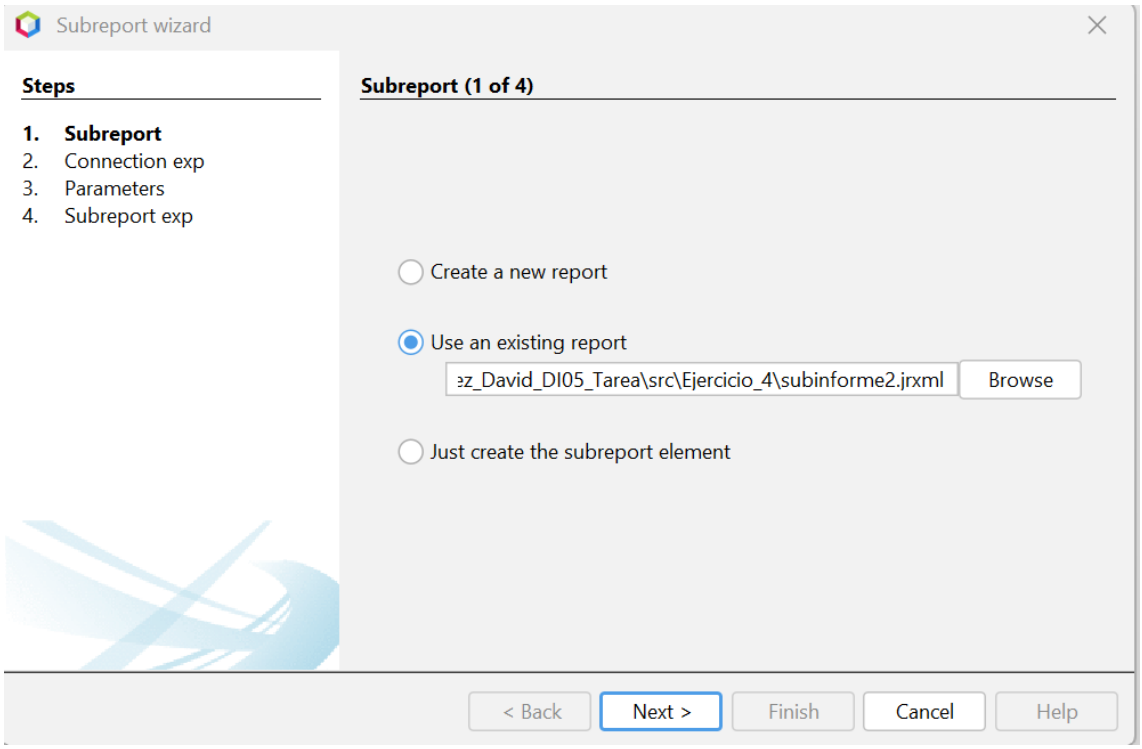
Por último, crearemos el segundo subinforme para incluir los precios totales de cada pedido. Para ello creamos un nuevo Empty Report y añadimos la siguiente consulta: En este caso añadiremos como parámetro ID_Pedido:



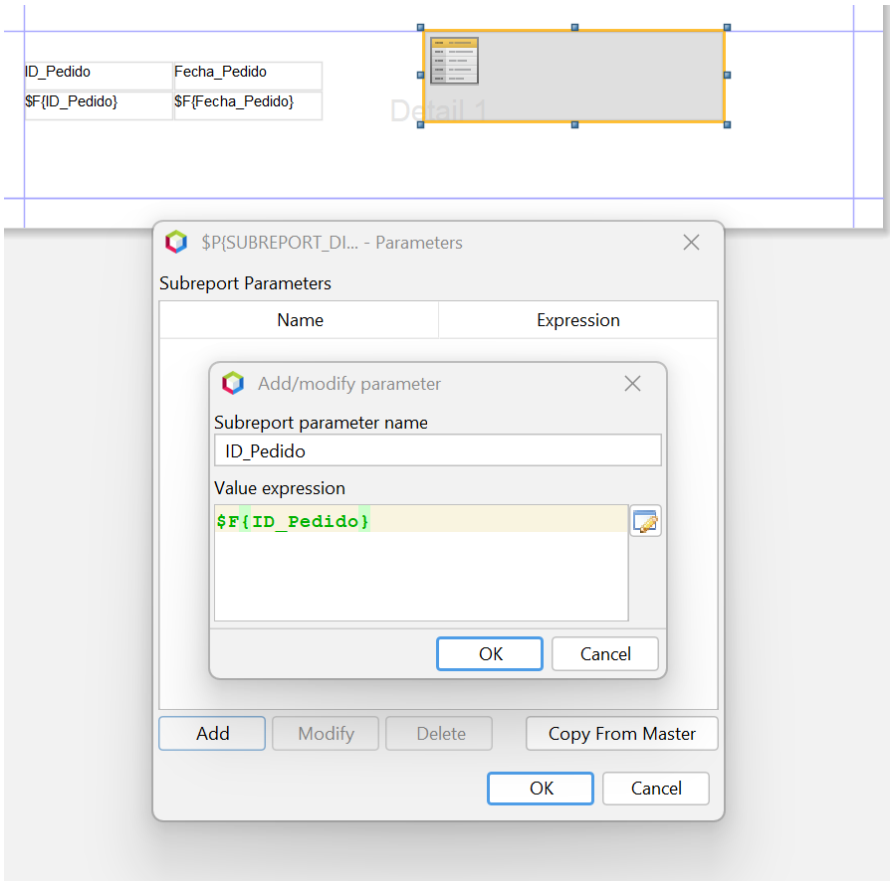
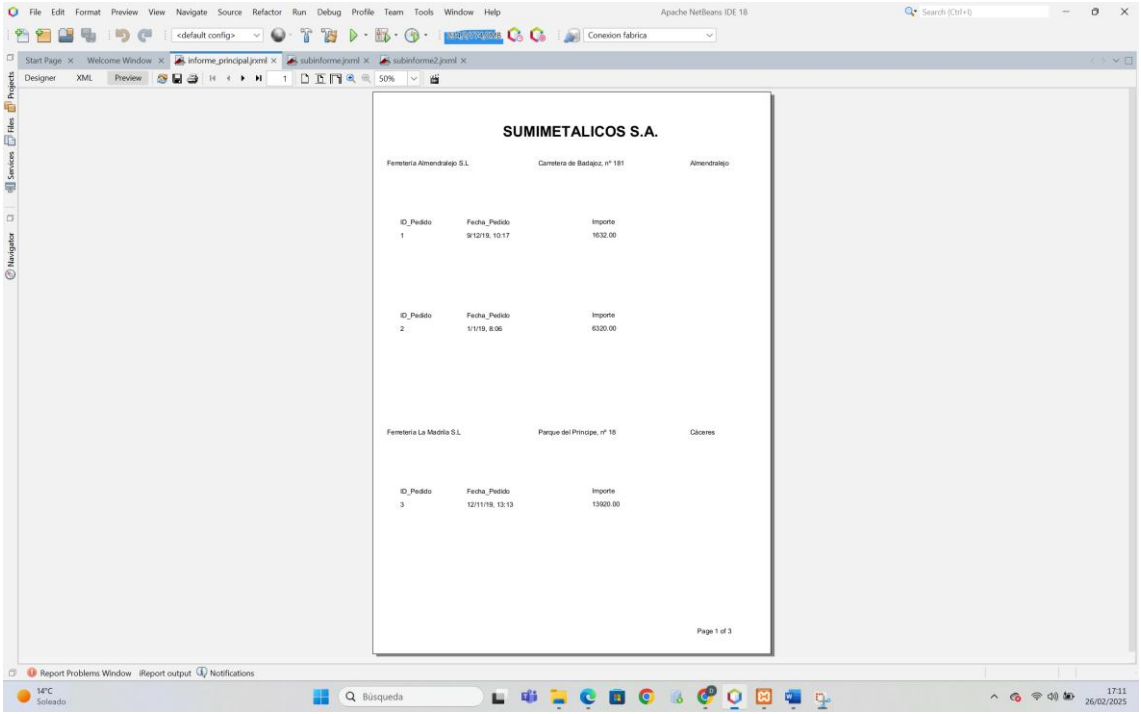
Añadimos el campo importe a Detail 1.



A continuación incluimos este subinforme2 al subinforme_pedidos:



Incluimos el parámetro ID_Pedido:



Compilamos todo y ahora quedaría configurado el informe principal para mostrar todos los datos: