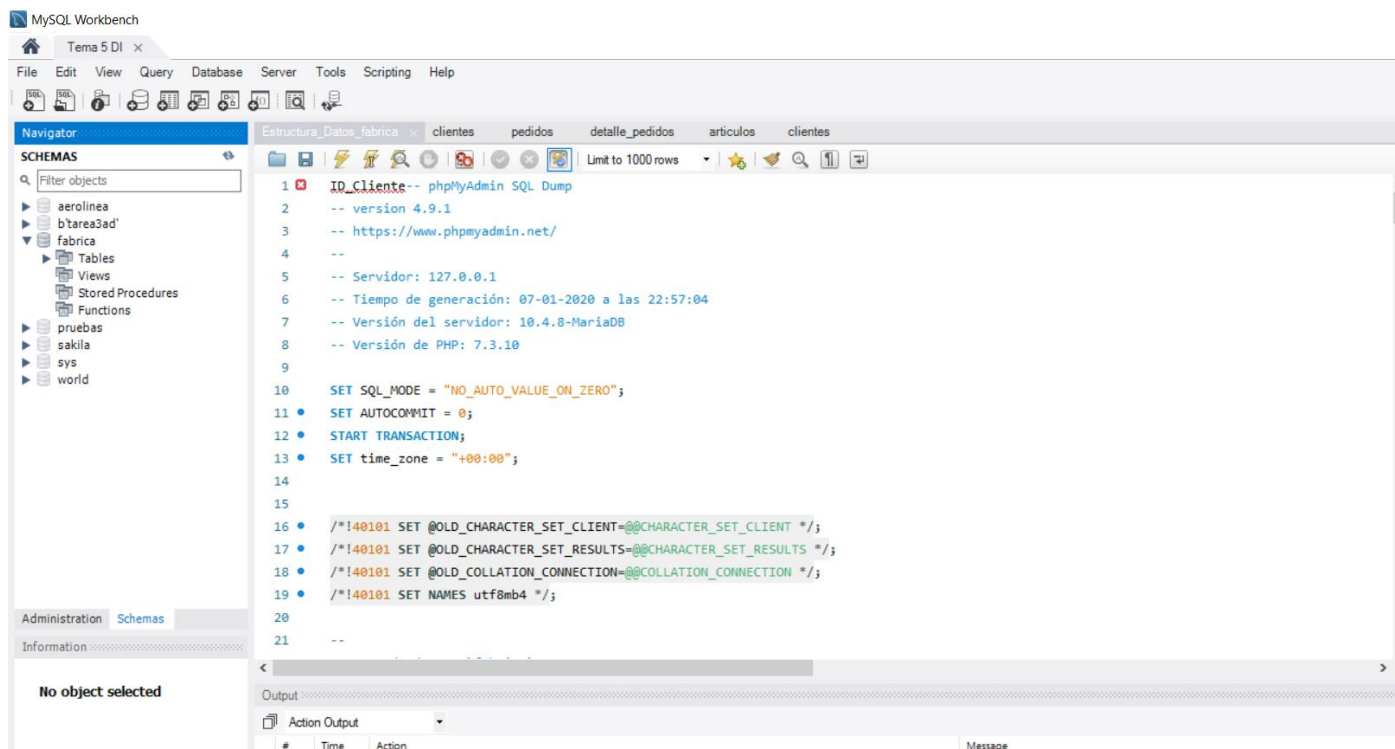
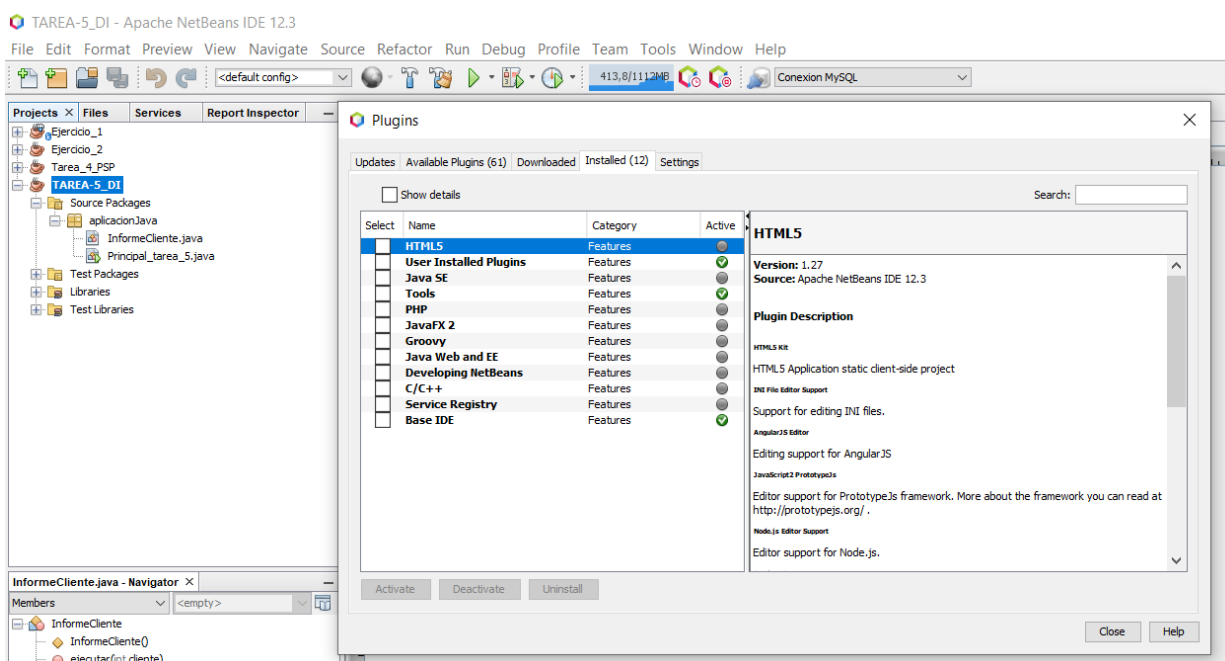


Para realizar esta tarea lo primero que voy a hacer será importar a MySQL la base de datos perteneciente a la compañía SumiMetalicos S.A que se nos facilita en la tarea:

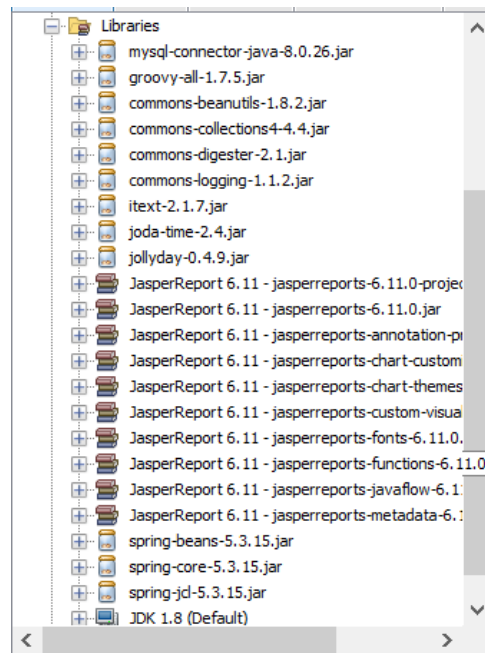


Ejecutamos el script y ya vemos en la parte izquierda como se nos ha creado el esquema "fabrica"

Lo segundo que voy a hacer es preparar mi entorno de NetBeans para poder generar informes en JasperReport instalando los plugins y librerías que se facilitan en el temario dentro del proyecto creado para la tarea:

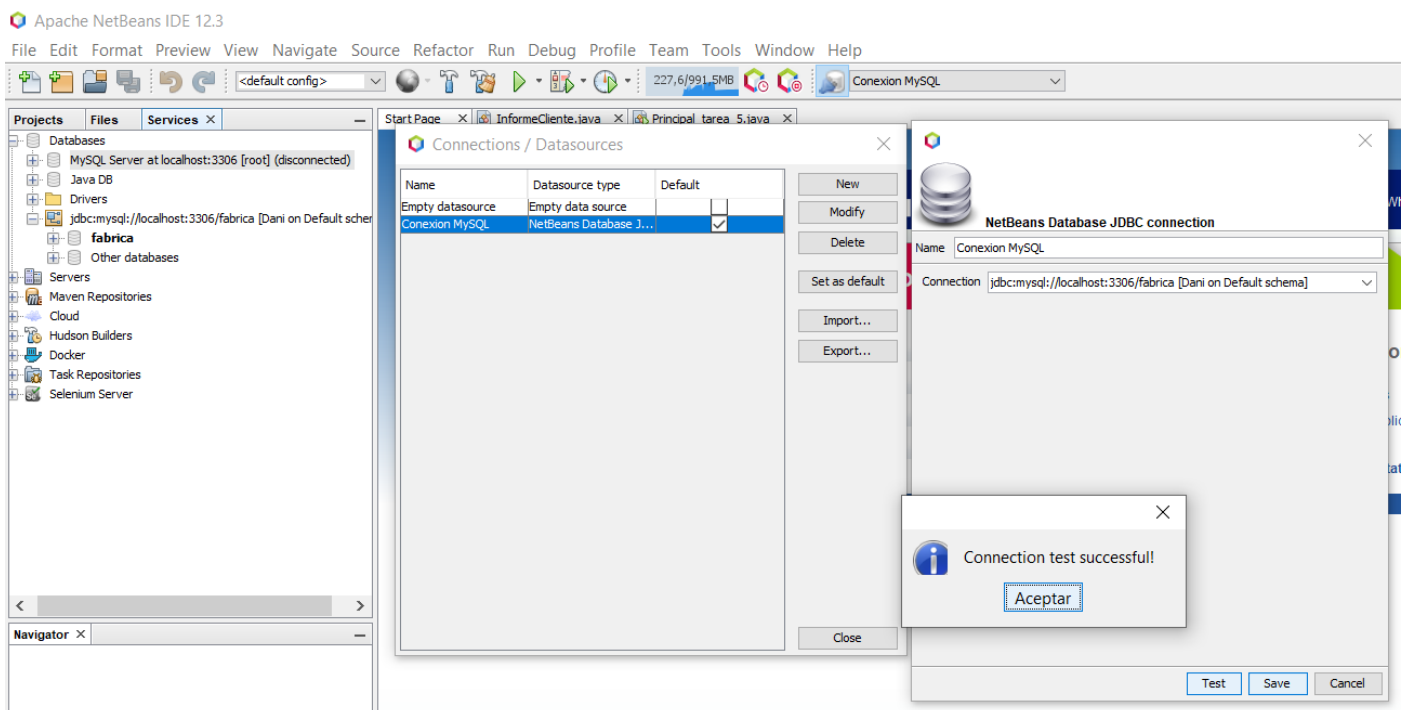


Instalo los plugins descargados del temario



Añado todas las librerías del temario incluido también el driver jdbc para MySQL

Ahora tenemos que conectar el proyecto NetBeans con la base de datos, pulsamos en conexiones y creamos una nueva con los datos de la base de datos creada:



Testeamos la conexión para ver que está bien

Una vez tengo ya todo preparado ya puedo crear los informes requeridos para la tarea.

1. GENERAR INFORME “FACTURAS”

Para generar este informe vamos a crear un nuevo “report wizard” dentro del proyecto creado para esta tarea. Se nos abrirá un asistente y vamos a seguir los siguientes pasos:

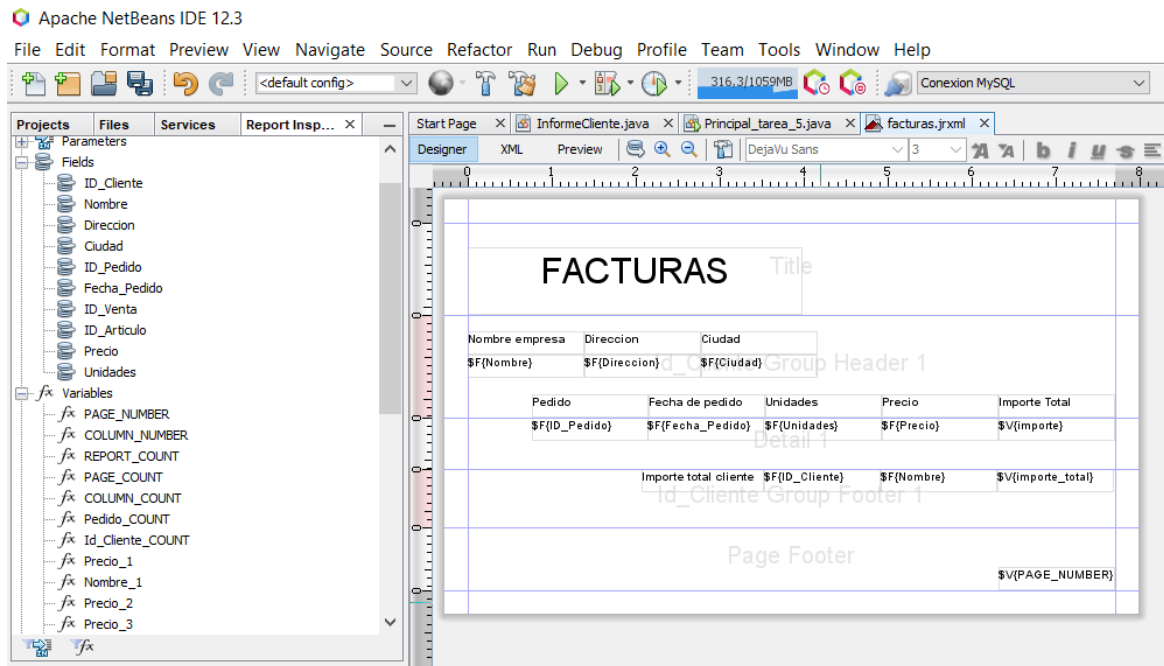
Damos nombre al informe

Asignamos la conexión y la sentencia SQL necesarias

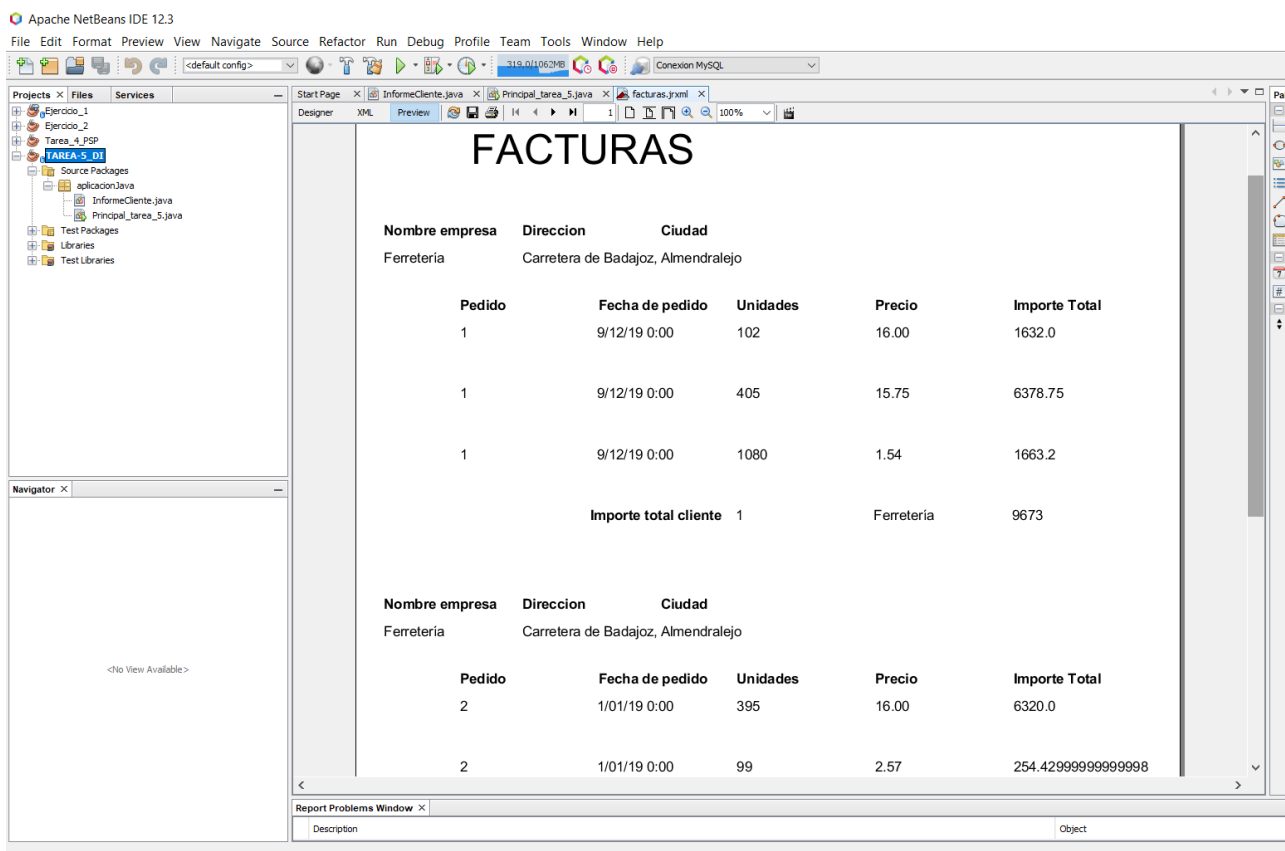
Seleccionamos todos los campos que vamos a necesitar

Elegimos las agrupaciones que queremos para nuestro informe

Ya se nos ha creado un fichero llamado “facturas.jrxml”, pinchamos sobre él y se nos abrirá el diseñador de informes. Añadimos los campos necesarios y las variables creadas “importe” e “importe total” a la plantilla que nos quedará de la siguiente forma:



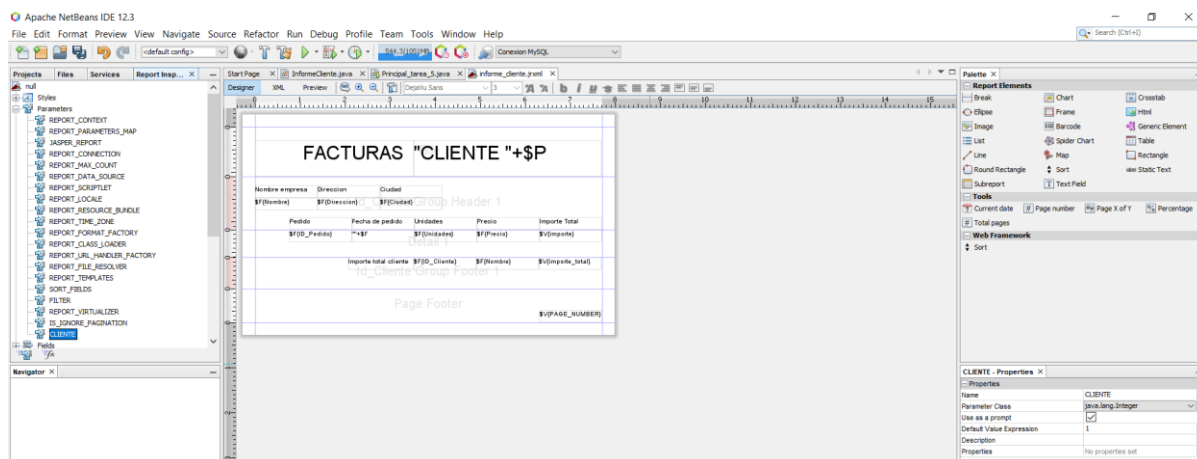
Diseño de la plantilla “facturas.jrxml”



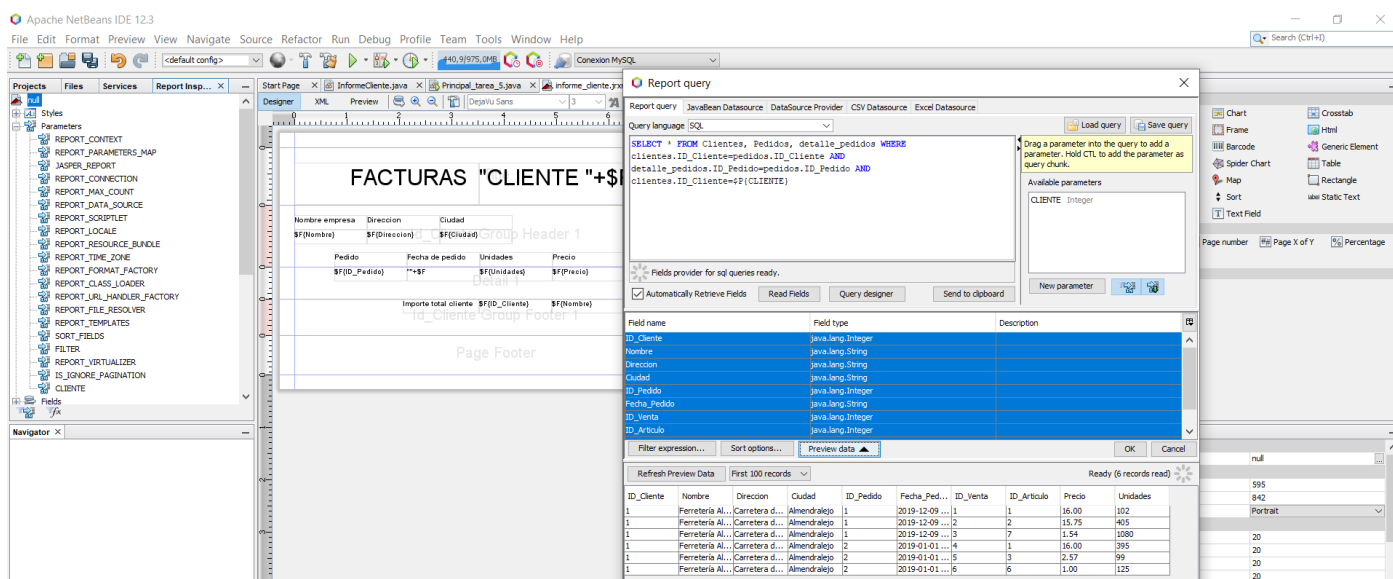
Preview de la plantilla

2. APLICACIÓN JAVA PARA MOSTRAR INFORME EN PDF

Se nos pide crear una aplicación java que dado un id de cliente nos muestre el informe de facturas como el creado en el apartado 1. Por lo que vamos a copiar el informe facturas y renombrarlo como “informe_cliente”:

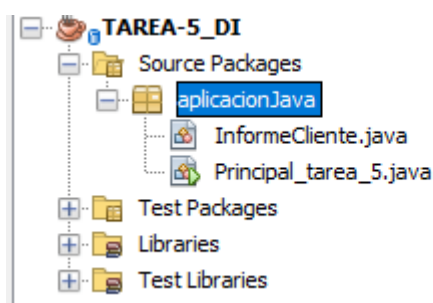


Dentro crearemos un nuevo parámetro llamado “cliente” de tipo entero



Modificamos también el Query del reporte para que tenga en cuenta el parámetro “cliente”

Ahora creamos 2 nuevas clases dentro del proyecto:



Dentro de la clase InformeCliente vamos a crear una conexión a la base de datos y crearemos una instancia del tipo Jasperreport con el informe creado "informe_cliente.jrxml", para posteriormente convertirlo en un archivo "informe.pdf":

```

22 public class InformeCliente {
23
24     public static Connection conexion = null;
25     String baseDatos = "jdbc:mysql://localhost/fabrica";
26     String usuario = "Dani";
27     String clave = "5678";
28
29     public InformeCliente() {
30         try {
31             Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
32             conexion = DriverManager.getConnection(baseDatos, usuario, clave);
33         } catch (ClassNotFoundException cnfe) {
34             System.err.println("Fallo al cargar JDBC");
35             System.exit(1);
36         } catch (SQLException sqle) {
37             System.err.println("No se pudo conectar a BD");
38             System.exit(1);
39         } catch (java.lang.InstantiationException sqlex) {
40             System.err.println("Imposible Conectar");
41             System.exit(1);
42         } catch (Exception ex) {
43             System.err.println("Imposible Conectar");
44             System.exit(1);
45         }
46     }
47
48     //El método ejecutar recibe el parametro del informe
49
50     public void ejecutar(int cliente) {
51         //Ruta del informe respecto del proyecto NetBeans
52         String archivojasper = "informe_cliente.jasper";
53         try {
54             //Cargamos los parametros del informe en una tabla Hash
55             Map parametros = new HashMap();
56             parametros.put("cliente", cliente);
57             //Generamos el informe en memoria
58             JasperPrint print = JasperFillManager.fillReport(archivojasper, parametros, conexion);
59             // Exporta el informe a PDF
60             JasperExportManager.exportReportToPdfFile(print, "informe.pdf");
61             //Abre el archivo PDF generado
62             File path = new File("informe.pdf");
63             Desktop.getDesktop().open(path);
64         } catch (Exception e) {
65             JOptionPane.showMessageDialog(null, e.toString(), "Error", JOptionPane.WARNING_MESSAGE);
66         }
67     }
68 }

```

Por otro lado en la clase principal creamos una instancia de la clase informeCliente pasándole como parámetro el id del cliente que queramos se genere el informe:

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  package aplicacionJava;
8
9  /**
10   *
11   * @author DANILOR
12   */
13
14  public class Principal_tarea_5 {
15
16      /**
17       * @param args the command line arguments
18       */
19
20      public static void main(String[] args) {
21          InformeCliente informe = new InformeCliente();
22          int cliente = 1; //Valor que se va a pasar para que lo recoja el parámetro
23          informe.ejecutar(cliente);
24      }
25  }

```

Ya solo nos queda comprobar que la aplicación funciona, así que ejecutamos y visualizamos el resultado para el cliente 1 por ejemplo:

The screenshot shows the Apache NetBeans IDE with the project 'TAREA-5_DI' open. The 'Principal_tarea_5.java' file is being edited, showing a Java class with a main method. The output window shows the successful execution of the application. To the right, a PDF report titled 'FACTURASCLIENTE 1' is displayed, showing a list of orders for 'Ferreteria Almendralejo S.L.'.

Nombre empresa	Direccion	Ciudad	Pedido	Fecha de pedido	Unidades	Precio	Importe Total
Ferreteria Almendralejo S.L.	Carretera de Badajoz, Almendralejo nº 181		1	2019-12-09 10:17:00	102	16.00	1632.0
			1	2019-12-09 10:17:00	405	15.75	6378.75
			1	2019-12-09 10:17:00	1080	1.54	1663.2
			Importe total cliente 1				9673.95

A la derecha podemos ver el informe generado en pdf

3. CREAR UN INFORME EN EL QUE APAREZCA UN LISTADO CON LAS VENTAS TOTALES DE CADA ARTÍCULO

Como hicimos en el primer ejercicio vamos a crear un nuevo informe utilizando el asistente de reportes con el nombre de "informe_ventas_totales" y con la sentencia de SQL que relaciona las tablas de artículos y detalles de pedidos:

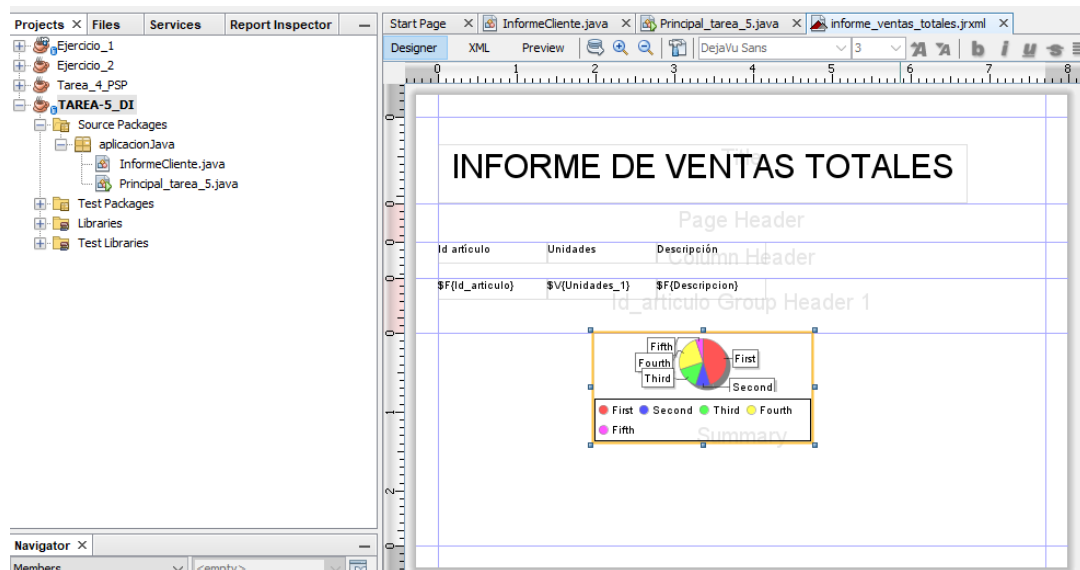
The screenshot shows the 'New File' dialog in NetBeans, with the 'Query' step selected. The 'Connections / Data Sources' dropdown is set to 'Conexion MySQL'. The 'Query (SQL)' text area contains the following SQL statement:

```
SELECT * FROM Articulos, detalle_pedidos WHERE detalle_pedidos.ID_Articulo=articulos.ID_articulo
```

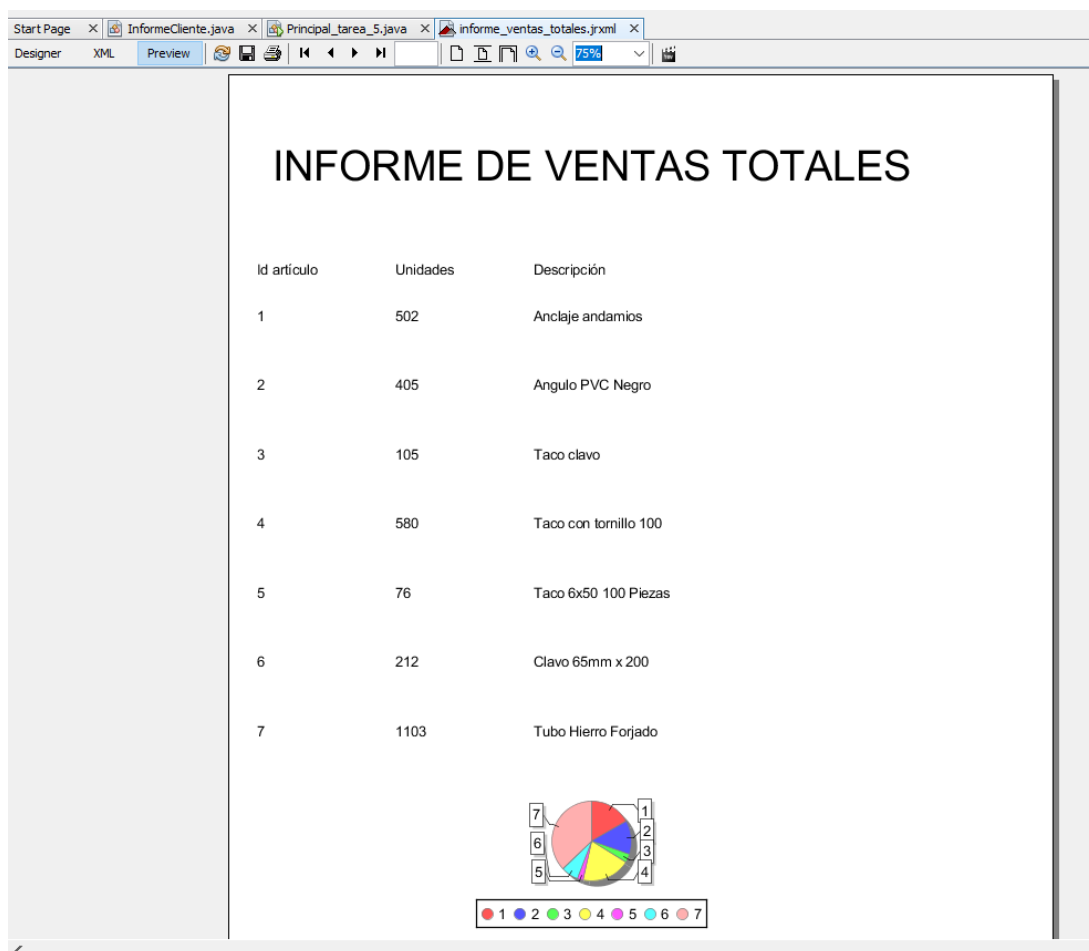
At the bottom of the dialog, there are buttons for 'Design query', 'Load query', and 'Save query'. The 'Next >' button is highlighted.

Seleccionamos todos los campos y agrupamos por id_artículo

Ya en la plantilla de diseño añadimos a la banda cabecera de id_articulo las unidades y la descripción. Y dentro de la banda de summary crearemos un gráfico con 2 campos (id_articulo y unidades), quedando el diseño de la siguiente manera:



Plantilla del reporte de ventas totales

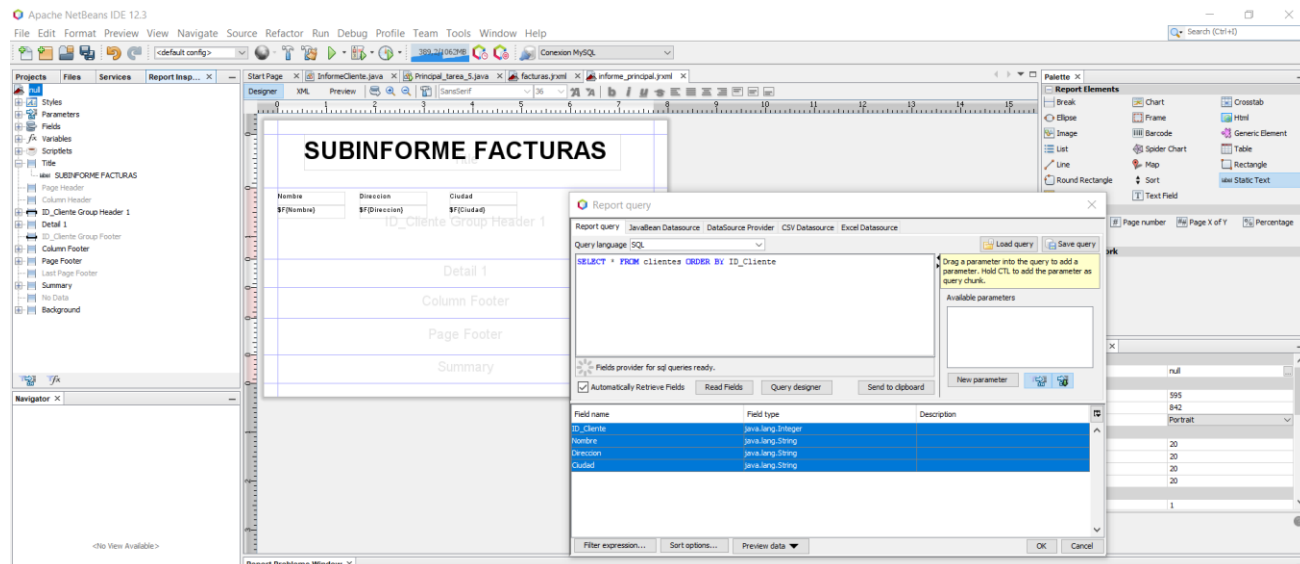


Pulsamos en preview para visualizar el resultado

4. EJERCICIO 1 UTILIZANDO SUBINFORMES

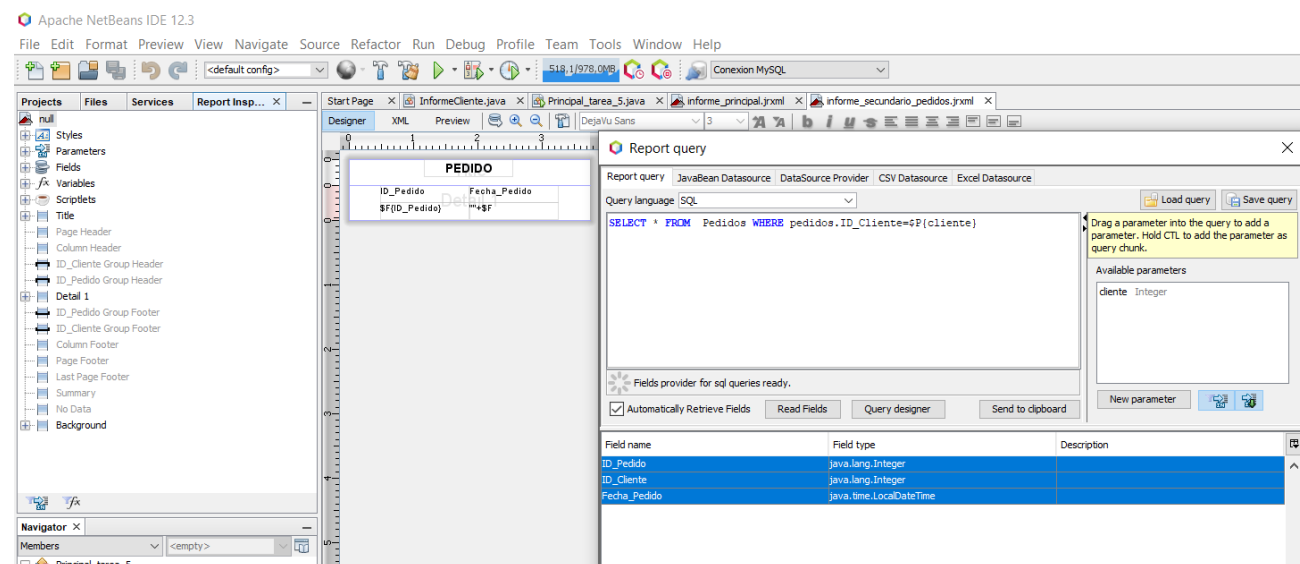
Para esta tarea vamos a crear un informe principal y 2 secundarios que luego vamos a unificar en un solo reporte.

Creamos el informe principal que relaciona las tablas cliente y pedidos mediante la query de SQL:

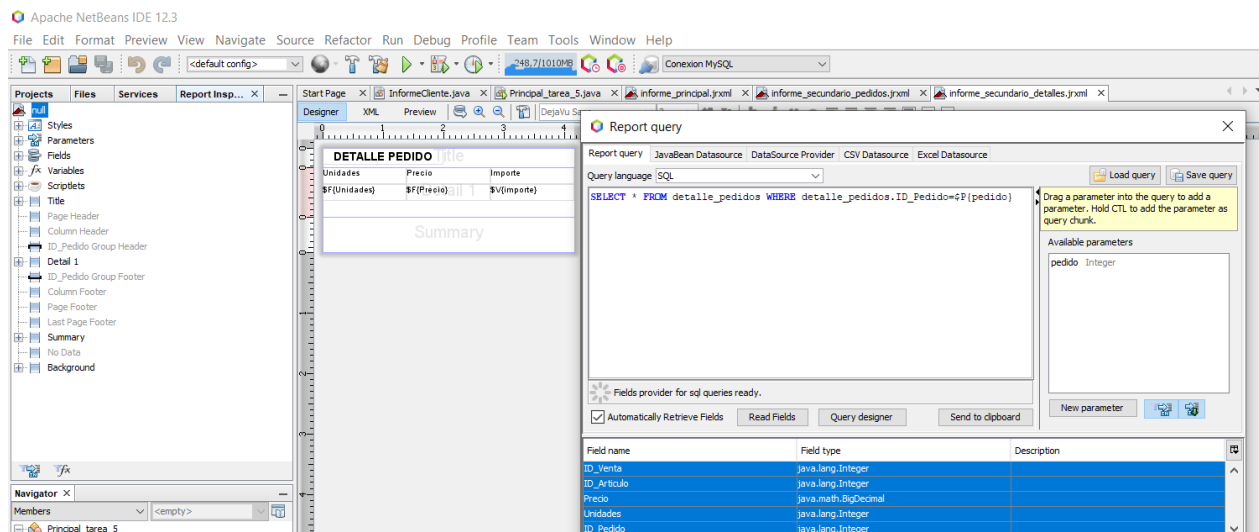


Desde la plantilla añadimos los campos nombre, dirección y ciudad dentro de la cabecera de cliente

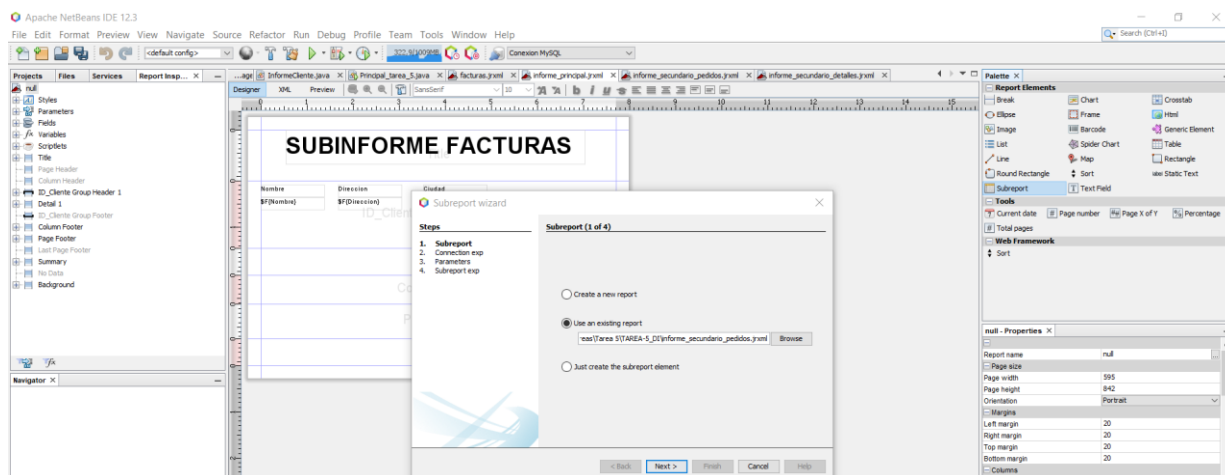
Creamos ahora otro informe al que llamaremos “informe_secundario_pedidos” que estará relacionado con la tabla de pedidos por medio del parámetro de id_cliente y quedará de la siguiente manera:



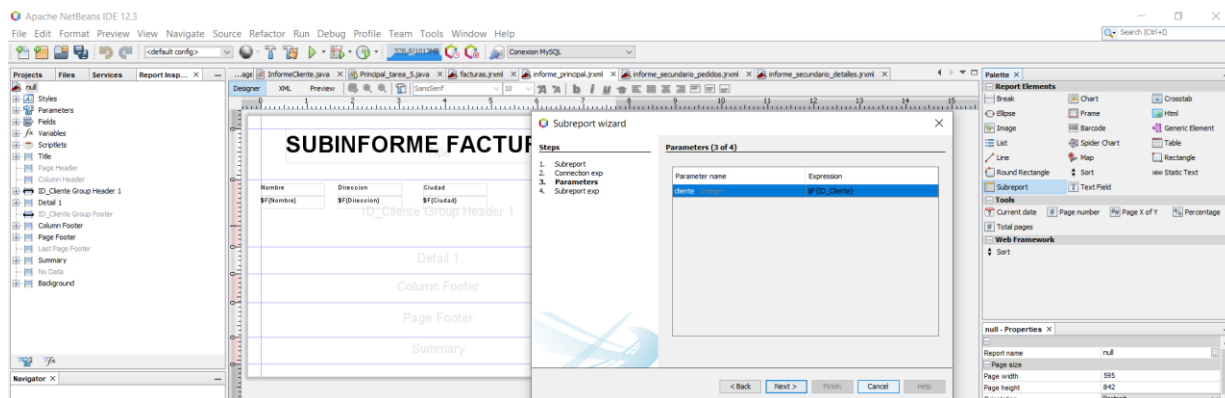
Creamos también otro informe al que llamaremos “informe_secundario_detalle” que estará relacionado con la tabla detalle pedidos por medio de parámetro id_pedido y quedará de la siguiente manera:



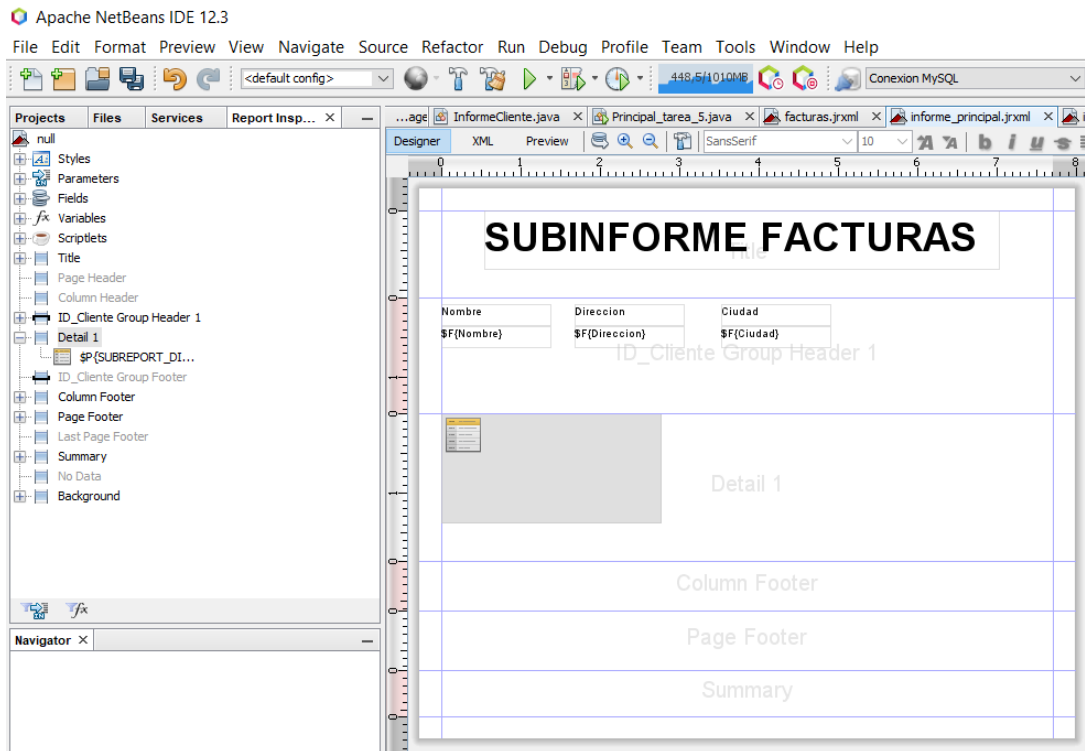
Una vez tenemos ya los 3 informes creados solo nos queda unificarlos, para ello vamos al informe principal y desde la paleta añadimos “subreport” y lo arrastramos a la banda “detail”, nos saldrá un asistente:



Le decimos que queremos seleccionar uno ya creado y seleccionamos informe_secundario_pedidos

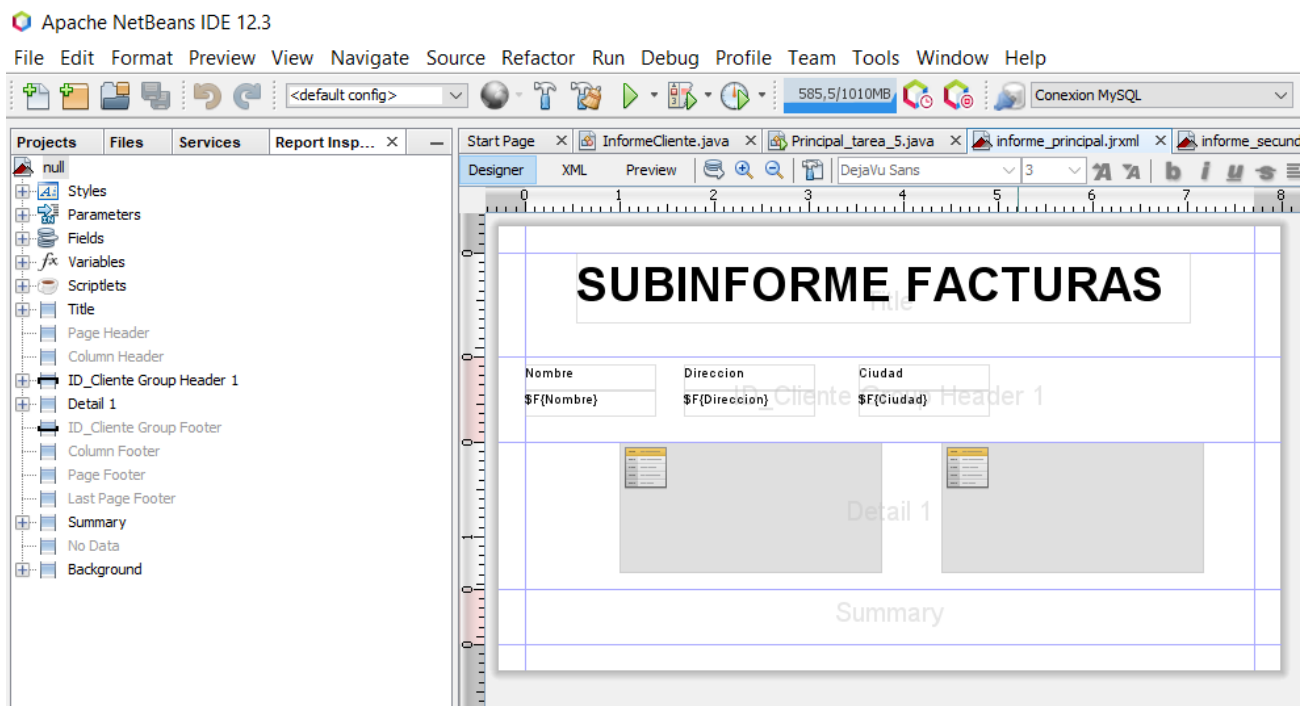


Relacionamos el campo id_cliente con el parámetro de cliente que hemos creado en el subinforme



Este sería el resultado de la plantilla después de añadir el primer subreporte de pedidos

Repetimos los mismos pasos para añadir, esta vez, el subinforme detalles de pedido a la misma banda, quedando la plantilla definitiva de la siguiente manera:



Pulsamos en preview para visualizar el resultado final de nuestro informe:

TAREA-5_DI - Apache NetBeans IDE 12.3

File Edit Format Preview View Navigate Source Refactor Run Debug Profile Team Tools Window Help

387.9/100.1MB Conexion MySQL

Projects Files X Services

Start Page X InformeCliente.java X Principal_tarea_5.java X informe_principal.jrxml X informe_secundario_pedidos.jrxml X informe_secundario_detalle.jrxml X

Designer XML Preview 1 75%

SUBINFORME FACTURAS

Nombre	Direccion	Ciudad
Ferretería	Carretera de Badajoz,	Almendralejo

PEDIDO		DETALLE PEDIDO		
ID_Pedido	Fecha_Pedido	Unidades	Precio	Importe
1	2019-12-09T10:17	102	16.00	1632.0
ID_Pedido	Fecha_Pedido	Unidades	Precio	Importe
2	2019-01-01T08:06	405	15.75	6378.75
		Unidades	Precio	Importe
		1080	1.54	1663.2

PEDIDO		DETALLE PEDIDO		
ID_Pedido	Fecha_Pedido	Unidades	Precio	Importe
1	2019-12-09T10:17	395	16.00	6320.0
ID_Pedido	Fecha_Pedido	Unidades	Precio	Importe
2	2019-01-01T08:06	99	2.57	254.4299999
		Unidades	Precio	Importe
		125	1.00	125.0

Navigator X

Members

Principal_tarea_5

Principal_tarea_5()

main(String[] args)