



CENTRO SAFA NUESTRA SEÑORA DE LOS REYES.  
DEPARTAMENTO DE INFORMÁTICA.

SGE

2ºDAM

Curso 2022/23

VÍCTOR ADONAY GALIANO POMBO

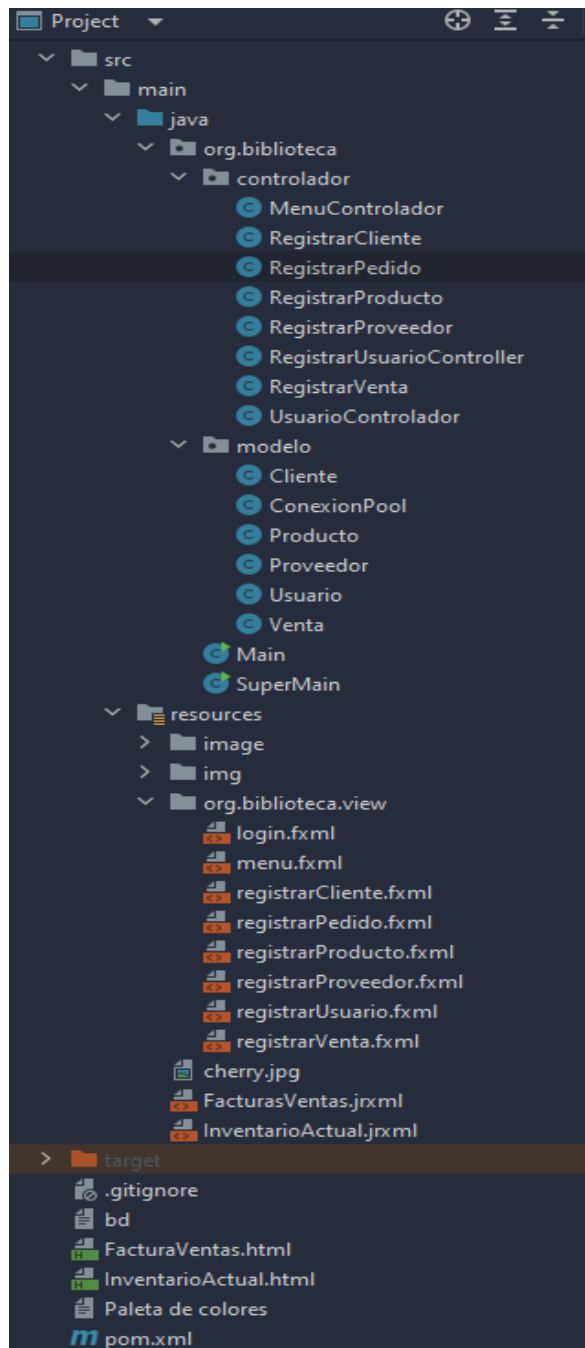




# Índice

1. Estructura del proyecto
2. Métodos de los modelos
3. Métodos de los controladores
4. Estructura de los archivos JavaFX

## Estructura del proyecto



## Métodos de los modelos

Vamos a comentar el modelo Venta, ya que es el más completo de todos los métodos.

Esta es la clase que representa una venta de la empresa

```
public class Venta {
```

Propiedades de la venta, utilizando JavaFX properties para facilitar el enlace de datos

```
private final IntegerProperty ventaId = new SimpleIntegerProperty();  
4 usages  
private final IntegerProperty cantidad = new SimpleIntegerProperty();  
4 usages  
private final ObjectProperty<Date> fecha = new SimpleObjectProperty<>();  
4 usages  
private final IntegerProperty clienteId = new SimpleIntegerProperty();  
4 usages  
private final IntegerProperty productoId = new SimpleIntegerProperty();  
4 usages  
private final IntegerProperty proveedorId = new SimpleIntegerProperty();
```

Constructor de la clase que inicializa las propiedades con los valores proporcionados

```
public Venta(int ventaId, int cantidad, Date fecha, int clienteId, int productoId, int proveedorId) {...}
```

## SGE

Métodos de acceso y modificación para la propiedades. Aunque la mayoría no se use, está bien ponerlos por escalabilidad del programa

```
no usages  Víctor Galiano
public int getVentaId() { return ventaId.get(); }
no usages  Víctor Galiano
public IntegerProperty ventaIdProperty() { return ventaId; }
no usages  Víctor Galiano
public void setVentaId(int ventaId) { this.ventaId.set(ventaId); }
no usages  Víctor Galiano
public int getCantidad() { return cantidad.get(); }
no usages  Víctor Galiano
public IntegerProperty cantidadProperty() { return cantidad; }
no usages  Víctor Galiano
public void setCantidad(int cantidad) { this.cantidad.set(cantidad); }
no usages  Víctor Galiano
public Date getFecha() { return fecha.get(); }
no usages  Víctor Galiano
public ObjectProperty<Date> fechaProperty() { return fecha; }
no usages  Víctor Galiano
public void setFecha(Date fecha) { this.fecha.set(fecha); }
no usages  Víctor Galiano
public int getClienteId() { return clienteId.get(); }
no usages  Víctor Galiano
public IntegerProperty clienteIdProperty() { return clienteId; }
no usages  Víctor Galiano
public void setClienteId(int clienteId) { this.clienteId.set(clienteId); }
no usages  Víctor Galiano
public int getProductoId() { return productoId.get(); }
no usages  Víctor Galiano
public IntegerProperty productoIdProperty() { return productoId; }
no usages  Víctor Galiano
public void setProductoId(int productoId) { this.productoId.set(productoId); }
no usages  lhuji
public int getProveedorId() { return proveedorId.get(); }
no usages  lhuji
public IntegerProperty proveedorIdProperty() { return proveedorId; }
no usages  lhuji
public void setProveedorId(int proveedorId) { this.proveedorId.set(proveedorId); }
no usages  Víctor Galiano
public void setPrecioProducto ( double precioProducto ) {}
no usages  Víctor Galiano
public void setPrecioSinIVA ( double precioSinIva ) {}
```

# Métodos de los controladores

Ahora comentaremos el controlador para la interfaz del menú principal.

```
public class MenuControlador {
```

Elementos de la interfaz para búsqueda, generación de informes HTML, mostrar fecha, hora, clientes, productos, proveedor, ventas, botones y registros.

```
2 usages
public TextField buscaNombre;
2 usages
public TextField buscaContacto;
2 usages
public TextField buscaDireccion;
no usages
public Button generarHTMLGeneral;
1 usage
public Button generarHTMLFactura;
@FXML
private Label fechaActual;

@FXML
private Label horaActual;

// Tabla y columnas para Clientes
@FXML
private TableView<Cliente> tablaCliente;

@FXML
private TableColumn<Cliente, Integer> colClientesID;

@FXML
private TableColumn<Cliente, String> colClientesNombre;

@FXML
private TableColumn<Cliente, String> colClientesDireccion;

@FXML
private TableColumn<Cliente, String> colClientesContacto;

@FXML
private TableColumn<Proveedor, String> colProveedoresNombre;

@FXML
private TableColumn<Proveedor, String> colProveedoresDireccion;

@FXML
private TableColumn<Proveedor, String> colProveedoresContacto;

// Botón para crear Proveedor
@FXML
private Button buttonCrearProveedor;

// Tabla y columnas para Ventas
@FXML
private TableView<Venta> tablaVentas;

@FXML
private TableColumn<Venta, Integer> colVentaID;

@FXML
private TableColumn<Venta, Integer> colVentaCantidad;

@FXML
private TableColumn<Venta, Date> colVentaFecha;

@FXML
private TableColumn<Venta, Integer> colVentaClienteID;

2 usages
@FXML
public TableColumn<Venta, Integer> colVentaProductoID;
2 usages
@FXML
public TableColumn<Venta, Integer> colVentaProveedorID;

// Botón para crear Venta
@FXML
private Button buttonCrearVenta;
```

Inicialización del controlador

```
@FXML
private void initialize() {...}
```

Configuración del reloj para mostrar fecha y hora actuales

```
private void configurarReloj() {...}
```

Configuración de las columnas de las tablas

```
private void configurarColumnas() {...}
```

Métodos para crear un nuevo cliente, producto, venta y pedido

```
@FXML
private void crearCliente(ActionEvent event) {...}
1 usage  👤 lhuji
@FXML
public void crearProducto() {...}
👤 lhuji +1
@FXML
private void crearProveedor(ActionEvent event) {...}
👤 lhuji +1
@FXML
private void crearVenta(ActionEvent event) {...}
1 usage  👤 lhuji
@FXML
public void crearPedido(ActionEvent event) {...}
```



Métodos para cargar los datos de clientes, productos, proveedores y ventas desde la base de datos

```
2 usages  👤 Víctor Galiano +1  
private void cargarDatosClientes() {...}  
2 usages  👤 Víctor Galiano  
private void cargarDatosProducto() {...}  
2 usages  👤 Víctor Galiano  
private void cargarDatosProveedores() {...}  
2 usages  👤 Víctor Galiano +1  
private void cargarDatosVentas() {...}
```

Métodos para generar un informe HTML del inventario actual y un HTML con todas las facturas

```
👤 lhuji +1  
@FXML  
private void generarHTMLInventario() throws SQLException {...}  
👤 lhuji +1  
@FXML  
private void generarHTMLFactura() throws SQLException {...}
```

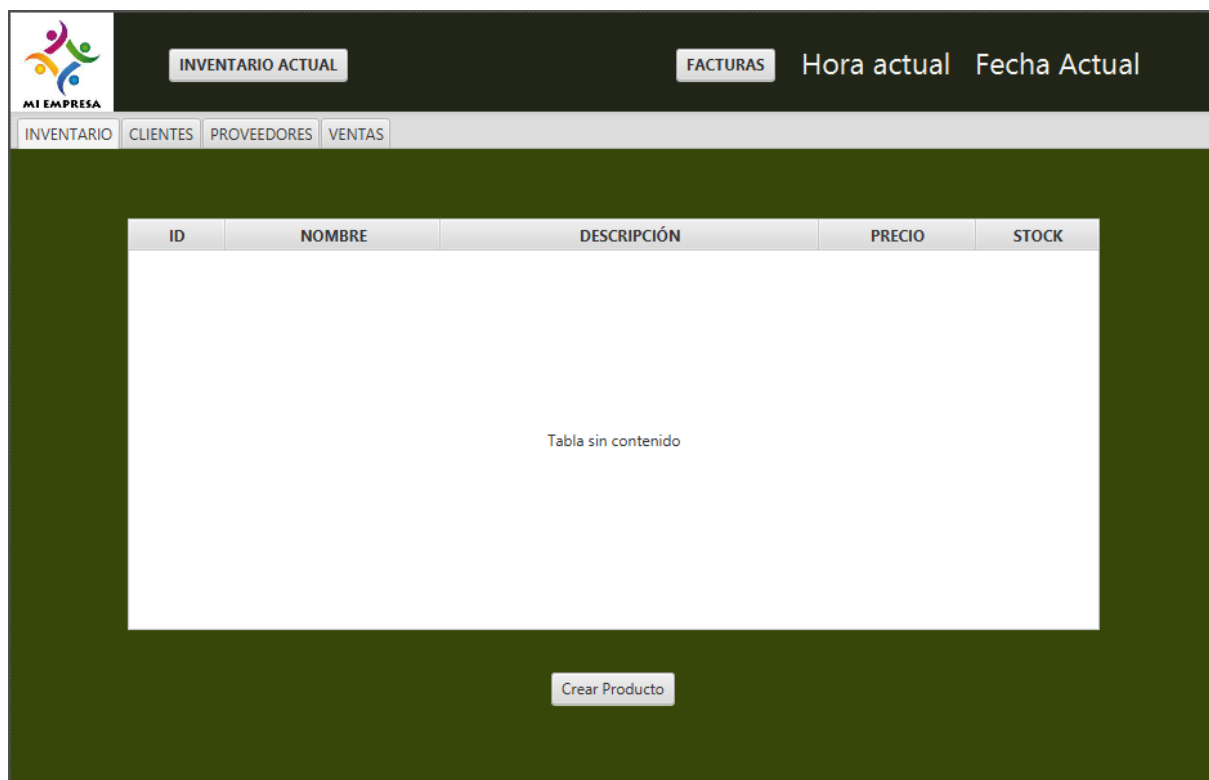
# Estructura de los archivos

## JavaFX

Ahora vamos a comentar el menú principal de la aplicación con todos sus componentes.

AnchorPane como contenedor principal:

Se utiliza un AnchorPane como contenedor principal con dimensiones preferidas especificadas.



Barra superior:

Incluye un ImageView con una imagen de logo.

Dos Label para mostrar la fecha y la hora actual.

Button para generar un informe de inventario (generarHTMLInventario).

Button para generar un informe de facturas (generarHTMLFactura).



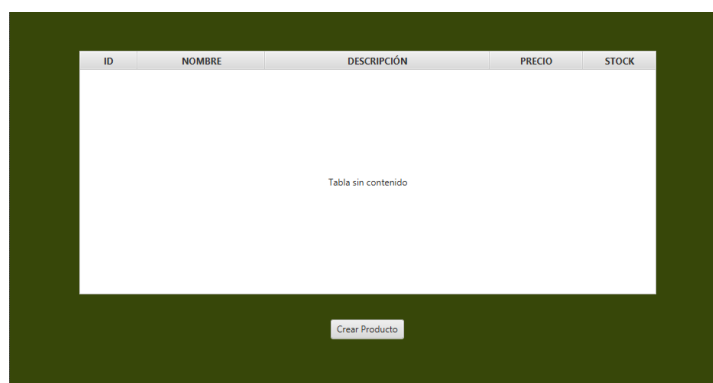
Pestañas (TabPane):

Contiene varias pestañas para diferentes secciones: "INVENTARIO", "CLIENTES", "PROVEEDORES" y "VENTAS".



Contenido de las Pestañas:

Cada pestaña tiene un AnchorPane que contiene un TableView para mostrar datos tabulares y botones asociados a cada sección.



Columnas de las Tablas (TableView):

Cada TableView tiene columnas (TableColumn) que representan diferentes atributos de los elementos (por ejemplo, productos, clientes, proveedores, ventas).

ID	NOMBRE	DESCRIPCIÓN	PRECIO	STOCK
Tabla sin contenido				

Botones en las Pestañas:

Botones como "Crear Producto", "Crear Cliente", "Crear Proveedor", "Crear Pedido" y "Crear Venta" asociados a las operaciones específicas de cada sección.

Crear Producto

Campos de Búsqueda (TextField):

En la pestaña "CLIENTES", hay tres campos de búsqueda para buscar por nombre, contacto y dirección.


Buscar por nombre

Busca por dirección

Buscar por contacto

## Estructura de los jasper

Ahora comentaremos un informe JasperReports utilizado para generar facturas de ventas. Aquí hay un resumen de sus principales características:

Factura				
		"Nº: " + \$F {fecha} + \$F {hora}		
		new java.util.Date()		
"Número de la venta: " + \$F {ventaId}				
"Cantidad: "+\$F {cantidad}	\$F {fecha}	"Cliente: " + "\n" + \$F {nombre_cliente}	"Producto: " + "\n" + \$F {nombre_producto}	"Proveedor: " + "\n" + \$F {nombre_proveedo}
"El precio sin IVA de la venta ha sido de: " + new java.text. Group1 Group Footer 1				
"El precio total de la venta ha sido de: " + new java.				
new java.util.Date()		"Page "+\$V" "+ \$V		
"El precio sin IVA total de las ventas ha sido de: " + new java. Summary				
"El precio con IVA total de las ventas ha sido de: " + new java.				

## SGE

Configuración del informe:

Se establecen propiedades como el tamaño de la página, los márgenes y el nombre del informe.

Se especifica la versión de JasperReports utilizada.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created with JasperSoft Studio version 6.20.6.final using JasperReports Library version 6.20.6-5c96b0aa8a39ac1dc6b0bea4b81 -->
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports http://jasperreports.sourceforge.net/jasperreports.xsd">
  <property name="com.jaspersoft.studio.data.sql.tables" value="" />
  <property name="com.jaspersoft.studio.data.defaultdataadapter" value="Mi Empresa" />
  <style name="Title" forecolor="#FFFFFF" backcolor="#BFBFBF" fontName="Times New Roman" fontSize="50" isBold="false" />
  <style name="SubTitle" forecolor="#CCCCCC" backcolor="#BFBFBF" fontName="Times New Roman" fontSize="18" isBold="false" />
  <style name="Column header" forecolor="#666666" backcolor="#BFBFBF" fontName="Times New Roman" fontSize="14" isBold="true" />
  <style name="Detail" mode="Transparent" backcolor="#BFBFBF" fontName="Times New Roman" />
  <style name="Row" mode="Transparent" backcolor="#BFBFBF" fontName="Times New Roman" pdfFontName="Times-Roman" />
</jasperReport>
```

Conexión a datos:

Se utiliza una consulta SQL para recuperar datos de ventas, clientes, productos y proveedores.

Se definen campos para cada columna en el resultado de la consulta.

```
<queryString language="SQL">
  <![CDATA[SELECT
v.ventaId,
v.cantidad,
v.fecha,
c.nombre AS nombre_cliente,
p.nombre AS nombre_producto,
pr.nombre AS nombre_proveedor,
p.precio AS precio_producto,
ROUND(p.precio * v.cantidad * 0.79, 2) AS precio_sin_iva
FROM venta v
JOIN cliente c ON v.clienteId = c.clienteId
JOIN producto p ON v.productoId = p.productoId
JOIN proveedor pr ON v.proveedorId = pr.proveedorId]]>
</queryString>
```

## SGE

Estilos:

Se definen varios estilos para diferentes elementos del informe, como títulos, subtítulos, encabezados de columnas y detalles.

```
<background...>
<pageHeader...>
<detail...>
<pageFooter...>
<summary...>
```

Variables:

Se define una variable llamada totalPrecioSinIva para calcular la suma del campo precio\_sin\_iva en el informe.

```
<variable name="totalPrecioSinIva" class="java.lang.Double" calculation="Sum">
  <variableExpression><![CDATA[${F{precio_sin_iva}}]></variableExpression>
</variable>
```

Grupos:

Se crea un grupo llamado Group1 basado en la columna ventald. El encabezado del grupo muestra información sobre el número de venta y la fecha.

```
<groupHeader>
  <band height="171">
    <frame>
      <reportElement mode="Opaque" x="0" y="137" width="555" height="24" forecolor="#889F7D" backcolor="#000000" uuid="0a5fc4ac-2e39-4b7a-
      <textField textAdjust="StretchHeight"...>
    </frame>
    <frame>
      <reportElement mode="Opaque" x="0" y="100" width="451" height="32" forecolor="#000000" backcolor="#CC0000" uuid="e3c14156-5b51-4201-
      <textField pattern="EEEE dd MMMM yyyy"...>
    </frame>
    <frame>
      <reportElement x="455" y="0" width="99" height="132" uuid="97705585-a0b5-4ec7-86f3-ae10c220e6af"/>
      <imageExpression><![CDATA["cherry.jpg"]]></imageExpression>
    </frame>
    <frame>
      <reportElement mode="Opaque" x="0" y="0" width="451" height="95" backcolor="#000000" uuid="6c6ca4e1-17cd-4913-93d6-5e670ea9ae8a"/>
      <staticText...>
      <textField...>
    </frame>
  </band>
</groupHeader>
```

El pie del grupo muestra información sobre el precio sin IVA de la venta y el precio total de la venta.

```
<groupFooter>
  <band height="198">
    <textField textAdjust="StretchHeight">
      <reportElement style="Detail" positionType="Float" x="210" y="60" width="344" height="28" uuid="a611aeb8-2df3-4529-b8a5-22f8f16c656b">
        <property name="com.jaspersoft.studio.spreadsheet.connectionID" value="054eaff1-ec1e-4de6-8247-0a35381d46f1"/>
      </reportElement>
      <textField textAlignment="Right" verticalAlignment="Middle">
        <font size="14"/>
      </textField>
      <textFieldExpression><![CDATA[El precio sin IVA de la venta ha sido de: " + new java.text.DecimalFormat("#,##0.00 €").format(new BigDec
</textField>
    <textField>
      <reportElement x="240" y="130" width="314" height="30" uuid="1142e7b1-829d-4671-b30a-e6becbb13db3"/>
      <textField textAlignment="Right" verticalAlignment="Middle">
        <font fontName="Times New Roman" size="14"/>
      </textField>
      <textFieldExpression><![CDATA[El precio total de la venta ha sido de: " + new java.text.DecimalFormat("#,##0.00 €").format($F{precio_pr
</textField>
    <break...>
  </band>
</groupFooter>
```

Detalles del informe:

Se especifica el diseño del detalle del informe, que incluye información sobre la cantidad, fecha, cliente, producto y proveedor de cada venta.

```
<detail>
  <band height="60" splitType="Stretch">
    <frame>
      <reportElement style="Row" mode="Opaque" x="0" y="0" width="555" height="60" uuid="b50221b5-8254-440b-8c6f-1f0b138c9259"/>
      <textField textAdjust="StretchHeight">
        <reportElement style="Detail" positionType="Float" x="0" y="0" width="111" height="60" backgroundColor="#FF0905" uuid="fd63767f-b9f7-4191-a71c-b95480e3
        <textField textAlignment="Center" verticalAlignment="Middle">
          <textFieldExpression><![CDATA[Cantidad: " + $F{cantidad}]]></textFieldExpression>
        </textField>
      <textField textAdjust="StretchHeight">
        <reportElement style="Detail" positionType="Float" x="111" y="0" width="111" height="60" uuid="dc365b89-a57a-489f-b200-6d6cda3342ad">
          <textField textAlignment="Center" verticalAlignment="Middle">
            <textFieldExpression><![CDATA[$F{fecha}]]></textFieldExpression>
          </textField>
      <textField textAdjust="StretchHeight">
        <reportElement style="Detail" positionType="Float" x="222" y="0" width="111" height="60" uuid="fcdaaab1-73df-439b-990e-4327d0e870e1">
          <textField textAlignment="Center" verticalAlignment="Middle">
            <textFieldExpression><![CDATA[Cliente: " + "\n" + $F{nombre_cliente}]]></textFieldExpression>
          </textField>
      <textField textAdjust="StretchHeight">
        <reportElement style="Detail" positionType="Float" x="333" y="0" width="111" height="60" uuid="6dfa9fb0-f2a4-4feb-a639-6496bf0a3e1b">
          <textField textAlignment="Center" verticalAlignment="Middle">
            <textFieldExpression><![CDATA[Producto: " + "\n" + $F{nombre_producto}]]></textFieldExpression>
          </textField>
      <textField textAdjust="StretchHeight">
        <reportElement style="Detail" positionType="Float" x="444" y="0" width="111" height="60" uuid="5c979369-8907-423b-a217-283ab51c903f">
          <textField textAlignment="Center" verticalAlignment="Middle">
            <textFieldExpression><![CDATA[Proveedor: " + "\n" + $F{nombre_proveedor}]]></textFieldExpression>
          </textField>
    </frame>
  </band>
</detail>
```



## Imágenes:

Se incluye una imagen en el informe (cherry.jpg), un logotipo o elemento gráfico.

```
<image>
  <reportElement x="455" y="0" width="99" height="132" uuid="97705585-a0b5-4ec7-86f3-ae10c220e6af"/>
  <imageExpression><![CDATA[cherry.jpg]]></imageExpression>
</image>
```

## Pie de página:

Se establece un pie de página que muestra el número de página y la fecha actual.

```
<pageFooter>
  <band height="25" splitType="Stretch">
    <frame>
      <reportElement mode="Opaque" x="0" y="1" width="555" height="24" forecolor="#D0B48E" backcolor="#000000" uuid="729c3539-f946-4d0e-a0a7-bda2815ea1b0"/>
      <textField evaluationTime="Report">
        <reportElement style="Column header" x="513" y="0" width="40" height="20" forecolor="#FFFFFF" uuid="4834e99c-7f6c-485a-b098-50e5187f2ab4"/>
        <textElement verticalAlignment="Middle">
          <font size="10" isBold="false"/>
        </textElement>
        <textFieldExpression><![CDATA[" " + ${PAGE_NUMBER}]]></textFieldExpression>
      </textField>
      <textField>
        <reportElement style="Column header" x="433" y="0" width="80" height="20" forecolor="#FFFFFF" uuid="9825b46e-76f1-469a-abb8-a1cc27aad685"/>
        <textElement textAlignment="Right" verticalAlignment="Middle">
          <font size="10" isBold="false"/>
        </textElement>
        <textFieldExpression><![CDATA["Page " + ${PAGE_NUMBER} + " of "]]></textFieldExpression>
      </textField>
      <textField pattern="EEEE dd MMMM yyyy">
        <reportElement style="Column header" x="2" y="1" width="197" height="20" forecolor="#FFFFFF" uuid="137d38c7-1a83-4fc5-b156-9d5f1b1a0ddc"/>
        <textElement verticalAlignment="Middle">
          <font size="10" isBold="false"/>
        </textElement>
        <textFieldExpression><![CDATA[new java.util.Date()]]></textFieldExpression>
      </textField>
    </frame>
  </band>
</pageFooter>
```

### Summary:

Se incluye una sección de resumen que muestra el precio sin IVA total de las ventas y el precio total con IVA.

Se utilizan expresiones en los campos de texto para mostrar información específica.

Se aplican formatos a los valores numéricos utilizando la clase `java.text.DecimalFormat`.

```
<summary>
  <band height="107" splitType="Stretch">
    <property name="com.jaspersoft.studio.unit.height" value="px"/>
    <textField>
      <reportElement x="0" y="10" width="555" height="40" uuid="4394b8fc-5192-4208-94c7-83af53c43572"/>
      <textElement>
        <font size="20"/>
      </textElement>
      <textFieldExpression><![CDATA["El precio sin IVA total de las ventas ha sido de: " + new java.text.DecimalFormat("#,##0.00 €").format(new BigDecimal($V{totalPrec:
    </textField>
    <textField>
      <reportElement x="0" y="60" width="555" height="40" uuid="9d1588cf-da31-4cd2-9327-b0f0f5d33343"/>
      <textElement>
        <font size="20"/>
      </textElement>
      <textFieldExpression><![CDATA["El precio con IVA total de las ventas ha sido de: " + new java.text.DecimalFormat("#,##0.00 €").format(new BigDecimal($V{totalPre:
    </textField>
  </band>
</summary>
```

En resumen, este archivo XML define un informe JasperReports detallado que presenta información sobre las ventas, incluyendo detalles de la venta, información del cliente, producto y proveedor, así como totales de precios con y sin IVA. La apariencia del informe se configura a través de estilos y se incluyen expresiones y variables para realizar cálculos y presentar información dinámica.