



SISTEMA DE VENTAS



Alumno:

Hernández Espinoza Gabriel Emanuel.

Materia:

Bases de datos

Profesor:

Roberto Lazcano González

Manual Técnico

Índice

- 01** Sistema
- 02** Estructura del proyecto
- 03** Librerías Utilizadas
- 04** Conexión a Base de Datos
- 05** Funcionalidades Principales
- 06** Flujo de Ventas
- 07** Generación de Reportes
- 08** Requisitos del Sistema
- 09** Instalación del Sistema
- 10** Contacto

01 SISTEMA

El presente documento describe el funcionamiento y aspectos técnicos del Sistema de Ventas de Escritorio, desarrollado en Java con SQLite como sistema de gestión de bases de datos. Esta aplicación está diseñada para cubrir las necesidades de pequeñas y medianas empresas (PyMEs) que requieren llevar un control eficiente de sus operaciones comerciales sin depender de servicios en línea o infraestructura costosa.

El objetivo principal del sistema es proporcionar una herramienta robusta y fácil de usar que permita gestionar clientes, proveedores, productos y ventas, así como generar reportes en formato PDF y archivos en Excel de manera automatizada, mejorando así los procesos administrativos y contables de la empresa.

Este sistema de escritorio ofrece una solución local, segura y práctica para negocios que buscan digitalizar su gestión sin complicaciones, integrando funcionalidades clave como autenticación de usuarios, mantenimiento de datos (altas, bajas y modificaciones), visualización de información y exportación de reportes.

02 Estructura del Proyecto

El proyecto se encuentra organizado en los siguientes paquetes:

- DAO: Contiene clases encargadas del acceso y manipulación de la base de datos para cada entidad del sistema (clientes, productos, proveedores, y demás.).
- ENTIDAD: Contiene las clases que modelan las entidades del dominio como Cliente, Producto, Proveedor, etc. Estas clases funcionan como modelos de datos con atributos y métodos getter/setter.
- INTERFAZ: Agrupa todas las interfaces gráficas (pantallas del sistema), implementadas con JFrame y JPanel.
- UTILIDADES: Incluye clases de soporte como la conexión a la base de datos (Conexion.java) y herramientas para generación de reportes o validaciones.

Esta estructura favorece la separación de responsabilidades, facilitando el mantenimiento y la escalabilidad del sistema.

03 Librerías Utilizadas

AbsoluteLayout.jar

- Permite diseñar interfaces gráficas (pantallas) con disposición absoluta, ubicando componentes (botones, textos, tablas) de forma precisa sobre el área de trabajo. Ideal para maquetar formularios visualmente.

swingx-all-1.6.4.jar

- Añade componentes visuales avanzados a Swing, como calendarios, paneles plegables, barras de tareas, etc. Mejora la experiencia gráfica del usuario.

commons-collections4-4.1.jar

- Proporciona estructuras de datos y algoritmos adicionales (listas, mapas, colas, conjuntos), facilitando la manipulación de colecciones más complejas.

poi-3.16.jar

- Parte del conjunto Apache POI, permite la lectura y escritura de archivos Excel (.xls).

poi-ooxml-3.16.jar

- Complementa poi para soportar archivos .xlsx (Excel moderno).

poi-ooxml-schemas-3.16.jar

- Contiene los esquemas XML necesarios para trabajar con archivos .xlsx.

xmlbeans-2.6.0.jar

- Requerido por Apache POI para manejar archivos Excel mediante estructuras XML.

itextpdf-5.5.1.jar

- Permite crear y modificar archivos PDF desde el sistema, como facturas o reportes automáticos.

sqlite-jdbc-3.49.1.0.jar

- Controlador JDBC que permite la conexión con bases de datos SQLite desde Java.

JDK

- Plataforma de desarrollo Java utilizada para compilar y ejecutar el sistema. Requiere compatibilidad con las librerías mencionadas.

04 Conexión a Base de Datos


El sistema utiliza una base de datos SQLite ubicada en la ruta relativa

`src/database/sistemaventa_COMPLETA.db`. La conexión se gestiona mediante una clase llamada `Conexion`, contenida en el paquete `Modelo`.

Esta clase implementa el método `getConnection()`, que realiza lo siguiente:

1. Define la URL de conexión con el prefijo `jdbc:sqlite`, propio del controlador JDBC para SQLite.
2. Establece la conexión con la base de datos mediante `DriverManager.getConnection(url)`.
3. Activa las claves foráneas con la instrucción `PRAGMA foreign_keys = ON`, lo cual permite que se respeten las relaciones entre tablas definidas en el esquema relacional.
4. Devuelve un objeto `Connection` que puede ser utilizado por las demás clases del sistema para ejecutar sentencias SQL.

El archivo `.db` debe estar ubicado en la ruta indicada para que la conexión se establezca correctamente. Si se cambia de ubicación, también debe actualizarse la ruta en el código.



```
package Modelo;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class Conexion {
    Connection con;
```

```
    public Connection getConnection() {
        try {
```

```
            String url = "jdbc:sqlite:src/database/sistemaventa_COMPLETA.db";
            con = DriverManager.getConnection(url);
```

```
            Statement stmt = con.createStatement();
            stmt.execute("PRAGMA foreign_keys = ON");
            stmt.close();
```

```
        } catch (SQLException e) {
            System.out.println("Error de conexión: " + e.getMessage());
        }
        return con;
```

```
    }
```

```
}
```


05 Funcionalidades Principales

El sistema cuenta con las siguientes funcionalidades clave:

- **Registro de Clientes, Proveedores y Productos:** Se puede agregar, modificar y eliminar información desde la interfaz gráfica.
- **Gestión de Ventas:** Permite seleccionar productos, registrar cantidades y generar facturas automáticas. Calcula el total de forma automática.
- **Generación de Reportes:** Se pueden exportar listados a PDF y Excel desde los módulos de productos y ventas.
- **Login de Usuario:** El acceso al sistema está protegido mediante un inicio de sesión simple.

andez Espinoza Gabriel Emanuel	
andez Almazan Tania Michelle	
Santamaria Rivera Zirar	

tab1 tab2 tab3 tab4 **tab5** tab6

ID	RUC	NOMBRE	TELEFONO	DIRECCION	RAZON SOCIAL
0	18122003	Cleto Reyes	564214200	Mexico	Deportes
1	18122004	VENUM	560116389	Estados Unidos	Deportes
2	181220	Fighters	1234567	Tailandia	Deportes

06 Flujo de Ventas

El módulo de ventas permite al usuario seleccionar un cliente, agregar productos a una lista con cantidades específicas, y posteriormente registrar la venta. El sistema actualiza automáticamente el inventario, genera una factura en PDF y almacena la venta en la base de datos. El flujo general es:

1. Selección de cliente.
2. Búsqueda y selección de productos.
3. Ingreso de cantidades.
4. Confirmación de venta.
5. Generación automática de factura PDF.
6. Descuento automático del stock.

En caso de falta de stock, el sistema muestra un mensaje de advertencia y evita que la venta se registre.

07 Generación de Reportes

Se implementó una funcionalidad para generar un reporte en formato Excel (.xlsx) que contiene la información de los productos registrados en el sistema. Para ello, se utilizó la librería Apache POI.

El reporte incluye:

- Logo institucional insertado en la hoja.
- Título con estilo centrado.
- Encabezados con formato personalizado (color de fondo y texto en blanco).
- Datos extraídos desde la base de datos (código, nombre, precio y stock).
- Estilos aplicados a todas las celdas para mejorar la presentación.
- Ajuste automático de columnas y nivel de zoom predefinido.

El archivo se guarda en la carpeta Descargas del usuario bajo el nombre productos.xlsx, y se abre automáticamente tras la generación.

08 Requisitos del Sistema

Para garantizar el correcto funcionamiento del Sistema de Ventas de Escritorio, se requiere contar con los siguientes componentes de hardware y software:

2.1 **Requisitos de Hardware (mínimo recomendado)**

- Procesador: Intel Core i3 o superior
- Memoria RAM: 4 GB o más
- Espacio en disco: 200 MB libres (sin contar archivos de exportación o base de datos creciente)
- Resolución de pantalla: 1366x768 o superior
- Puerto USB (opcional, si se desea respaldo físico)

2.2 **Requisitos de Software**

Sistema operativo compatible:

- Windows 10 o superior
- Distribuciones Linux modernas (como Ubuntu, Fedora, etc.)

Java Development Kit (JDK):

- JDK versión 17 o superior

Entorno de Desarrollo Recomendado:

- NetBeans IDE versión 12 o superior

Base de Datos:

- SQLite versión 3.39 o superior


Dependencias y librerías necesarias:

- JDBC para SQLite: sqlite-jdbc-3.36.0.3.jar o versión equivalente
- Librerías para generación de reportes en PDF: iText 5.x o iText 7
- Librerías para generación de archivos Excel: Apache POI (para archivos .xlsx) y JExcelAPI (para archivos .xls, si se opta por esta alternativa)

Drivers JDBC

- Incluidos como archivos .jar en el proyecto o añadidos al classpath desde NetBeans

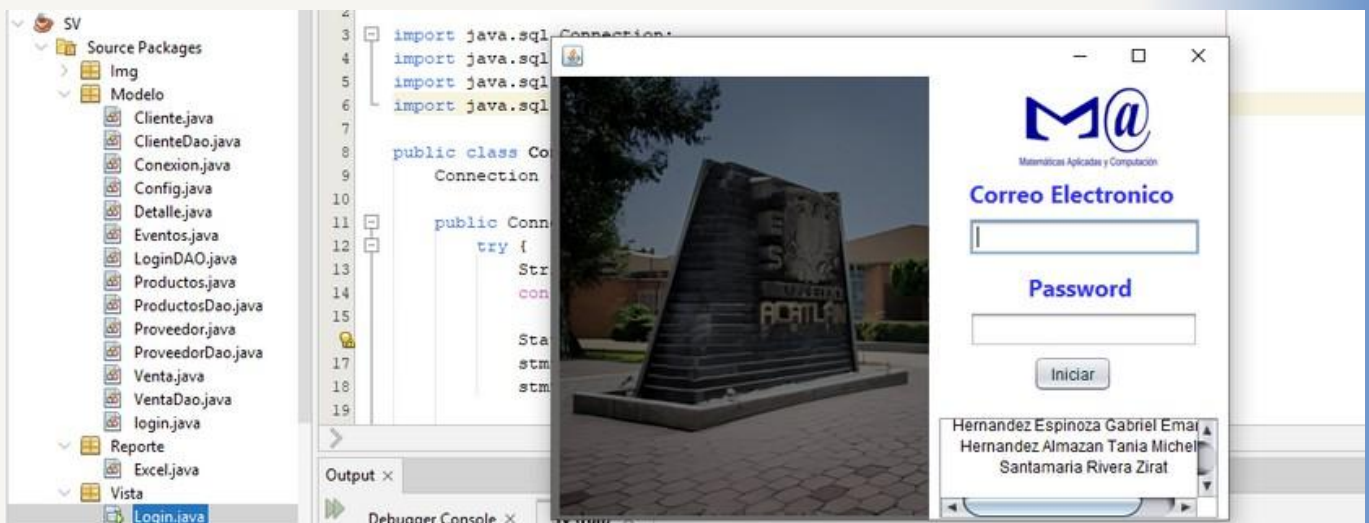


	A	B	C	D	
1					
2	F	 Reporte de Productos			
3	E				
4	S				
5	UNAM				
6	ACATLÁN	Código	Nombre	Precio	Existencia
7		123456	Guantes MMA	380.0	0
8		1234	Uniforme	700.0	47
9		123	Maleta Deportiva	450.0	18
10		12	Espinilleras	300.0	23
11		12345	Telefonos	3200.0	148
12					
13					

09 Instalación del Sistema

1. Descargar o clonar el proyecto a una carpeta local.
2. Abrir el proyecto en NetBeans.
3. Verificar que las librerías externas estén añadidas al classpath: Click derecho sobre el proyecto → Properties → Libraries → Add JAR/Folder e incluir los .jar necesarios como SQLite JDBC, iTextPDF, etc.
4. Confirmar que el archivo de base de datos sistemaventa_COMPLETA.db se encuentra en: src/database/sistemaventa_COMPLETA.db
5. Ejecutar el archivo Login.java para iniciar el sistema.

Nota: Si al descargar el proyecto sale un signo de admiración, es posible que necesites reparar el proyecto para que se adapte a tu versión de NetBeans.



10 Contacto

Hernández Espinoza Gabriel Emanuel

320151772@pcpuma.acatlan.unam.mx