Ejercicios Java 9

Diagrama de clases

Diagrama

El contenido generado por IA puede ser incorrecto.

JAVAPOO.sql

CREATE DATABASE JAVAPOO;

USE JAVAPOO;

create table clientes (

id\_cliente INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100),

email VARCHAR(100),

telefono VARCHAR(15)

);

create table proveedores (

id\_proveedor INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100),

cif VARCHAR(20),

telefono VARCHAR(15)

);

create table articulos (

id\_articulo INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100),

precio\_unitario DECIMAL(10,2),

stock INT

);

create table facturas (

id\_factura INT PRIMARY KEY AUTO\_INCREMENT,

id\_proveedor INT,

fecha DATE,

total DECIMAL(10,2),

FOREIGN KEY (id\_proveedor) references proveedores (id\_proveedor)

);

create table ventas (

id\_venta INT PRIMARY KEY AUTO\_INCREMENT,

id\_cliente INT,

id\_articulo INT,

cantidad INT,

fecha\_venta DATE,

FOREIGN KEY (id\_cliente) references clientes (id\_cliente),

FOREIGN KEY (id\_articulo) references articulos (id\_articulo)

);

INSERT INTO Clientes (nombre, email, telefono) VALUES

('Juan Pérez', 'juanp@gmail.com', '600123456'),

('Lucía Gómez', 'lucia.gomez@gmail.com', '611987654');

INSERT INTO Proveedores (nombre, cif, telefono) VALUES

('TechDistribuciones S.A.', 'A12345678', '913456789'),

('HardwarePro', 'B98765432', '914789123');

INSERT INTO Articulos (nombre, precio\_unitario, stock) VALUES

('Monitor 24"', 149.99, 20),

('Teclado Mecánico', 69.90, 35);

INSERT INTO Facturas (id\_proveedor, fecha, total) VALUES

(1, '2025-05-01', 3000.00),

(2, '2025-05-03', 1500.00);

INSERT INTO Ventas (id\_cliente, id\_articulo, cantidad, fecha\_venta) VALUES

(1, 1, 2, '2025-05-05'),

(2, 2, 1, '2025-05-06');

Conexión

**package** config;

**import** java.sql.\*;

**public** **class** Conexion { //Crea una conexión con la base de datos

Connection conexion;

**public** Connection conectar() {

**try** {

conexion = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/JAVAPOO", "root", "curso");

} **catch** (Exception e) {

System.***out***.println("Error al conectarse: " + e.getMessage());

}

**return** conexion;

}

}

ControladorCliente

package controlador;

import config.\*;

import java.sql.\*;

public class ControladorCliente {

Conexion miConexion;

public ControladorCliente() {

miConexion = new Conexion();

}

public void registrarCliente(String nombre, String email, int telefono) { //Registra los datos de un nuevo cliente

Connection conexion = miConexion.conectar();

try {

String sql = "INSERT INTO clientes (nombre, email, telefono) VALUES (?, ?, ?)";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setString(2, email);

pstmt.setInt(3, telefono);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Cliente registrado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void listarClientes() { //Muestra los datos de todos los clientes registrados

Connection conexion = miConexion.conectar();

try {

Statement stmt = conexion.createStatement();

ResultSet resultado = stmt.executeQuery("SELECT \* FROM clientes");

while (resultado.next()) {

System.out.println("ID: " + resultado.getInt("id\_cliente") + ", Nombre: " + resultado.getString("nombre") +

", Correo: " + resultado.getString("email") + ", Teléfono: " + resultado.getString("telefono"));

}

System.out.println();

stmt.close();

conexion.close();

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void modificarCliente(int id, String nombre, String email, int telefono) { //Modifica los datos del cliente indicado

Connection conexion = miConexion.conectar();

try {

String sql = "UPDATE clientes SET nombre = ?, email = ?, telefono = ? WHERE id\_cliente = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setString(2, email);

pstmt.setInt(3, telefono);

pstmt.setInt(4, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Cliente modificado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void eliminarCliente(int id) { //Elimina los datos del cliente indicado

Connection conexion = miConexion.conectar();

try {

String sql = "DELETE FROM clientes WHERE id\_cliente = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Cliente eliminado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

}

VistaCliente

package vista;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorCliente;

public class VistaCliente {

static String saltoDeLinea;

public void menuCliente() { // Muestra las opciones de CRUD de la tabla clientes

Scanner obj = new Scanner(System.in);

ControladorCliente c = new ControladorCliente();

do {

try {

System.out.println("1. Registrar cliente");

System.out.println("2. Listar clientes");

System.out.println("3. Modificar cliente");

System.out.println("4. Eliminar cliente");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

if (opcion == 1) { //Registra un nuevo cliente

System.out.println("Introduce el nombre del cliente:");

String nombre = obj.nextLine();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce la dirección de correo del cliente:");

String email = obj.nextLine();

System.out.println();

System.out.println("Introduce el número de teléfono del cliente:");

int telefono = obj.nextInt();

System.out.println();

c.registrarCliente(nombre, email, telefono);

} else if (opcion == 2) { //Muestra los datos de los clientes

c.listarClientes();

} else if (opcion == 3) { //Modifica los datos de un cliente

System.out.println("Introduce el id del cliente a modificar:");

int id = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el nuevo nombre del cliente:");

String nombre = obj.nextLine();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce la nueva dirección de correo del cliente:");

String email = obj.nextLine();

System.out.println();

System.out.println("Introduce el nuevo número de teléfono del cliente:");

int telefono = obj.nextInt();

System.out.println();

c.modificarCliente(id, nombre, email, telefono);

} else if (opcion == 4) { //Elimina los datos de un cliente

System.out.println("Introduce el id del cliente a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarCliente(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

ControladorProveedor

package controlador;

import config.\*;

import java.sql.\*;

public class ControladorProveedor {

Conexion miConexion;

public ControladorProveedor() {

miConexion = new Conexion();

}

public void registrarProveedor(String nombre, String cif, int telefono) { //Registra los datos de un nuevo proveedor

Connection conexion = miConexion.conectar();

try {

String sql = "INSERT INTO proveedores (nombre, cif, telefono) VALUES (?, ?, ?)";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setString(2, cif);

pstmt.setInt(3, telefono);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Proveedor registrado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void listarProveedores() { //Muestra los datos de todos los proveedor registrados

Connection conexion = miConexion.conectar();

try {

Statement stmt = conexion.createStatement();

ResultSet resultado = stmt.executeQuery("SELECT \* FROM proveedores");

while (resultado.next()) {

System.out.println("ID: " + resultado.getInt("id\_proveedor") + ", Nombre: " + resultado.getString("nombre") +

", CIF: " + resultado.getString("cif") + ", Teléfono: " + resultado.getString("telefono"));

}

System.out.println();

stmt.close();

conexion.close();

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void modificarProveedor(int id, String nombre, String cif, int telefono) { //Modifica los datos del proveedor indicado

Connection conexion = miConexion.conectar();

try {

String sql = "UPDATE proveedores SET nombre = ?, cif = ?, telefono = ? WHERE id\_proveedor = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setString(2, cif);

pstmt.setInt(3, telefono);

pstmt.setInt(4, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Proveedor modificado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void eliminarProveedor(int id) { //Elimina los datos del proveedor indicado

Connection conexion = miConexion.conectar();

try {

String sql = "DELETE FROM proveedores WHERE id\_cliente = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Proveedor eliminado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

}

VistaProveedor

package vista;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorProveedor;

public class VistaProveedor {

static String saltoDeLinea;

public void menuProveedor() { // Muestra las opciones de CRUD de la tabla proveedores

Scanner obj = new Scanner(System.in);

ControladorProveedor c = new ControladorProveedor();

do {

try {

System.out.println("1. Registrar proveedor");

System.out.println("2. Listar proveedores");

System.out.println("3. Modificar proveedor");

System.out.println("4. Eliminar proveedor");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

if (opcion == 1) { //Registra un nuevo proveedor

System.out.println("Introduce el nombre del proveedor:");

String nombre = obj.nextLine();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el cif del proveedor:");

String cif = obj.nextLine();

System.out.println();

System.out.println("Introduce el número de teléfono del proveedor:");

int telefono = obj.nextInt();

System.out.println();

c.registrarProveedor(nombre, cif, telefono);

} else if (opcion == 2) { //Muestra los datos de los proveedores

c.listarProveedores();

} else if (opcion == 3) { //Modifica los datos de un proveedor

System.out.println("Introduce el id del proveedor a modificar:");

int id = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el nuevo nombre del proveedor:");

String nombre = obj.nextLine();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el nuevo cif del proveedor:");

String email = obj.nextLine();

System.out.println();

System.out.println("Introduce el nuevo número de teléfono del proveedor:");

int telefono = obj.nextInt();

System.out.println();

c.modificarProveedor(id, nombre, email, telefono);

} else if (opcion == 4) { //Elimina los datos de un proveedor

System.out.println("Introduce el id del proveedor a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarProveedor(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

ControladorArticulo

package controlador;

import config.\*;

import java.sql.\*;

public class ControladorArticulo {

Conexion miConexion;

public ControladorArticulo() {

miConexion = new Conexion();

}

public void registrarArticulo(String nombre, double precio, int stock) { //Registra los datos de un nuevo artículo

Connection conexion = miConexion.conectar();

try {

String sql = "INSERT INTO articulos (nombre, precio\_unitario, stock) VALUES (?, ?, ?)";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setDouble(2, precio);

pstmt.setInt(3, stock);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Artículo registrado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void listarArticulos() { //Muestra los datos de todos los artículos registrados

Connection conexion = miConexion.conectar();

try {

Statement stmt = conexion.createStatement();

ResultSet resultado = stmt.executeQuery("SELECT \* FROM articulos");

while (resultado.next()) {

System.out.println("ID: " + resultado.getInt("id\_articulo") + ", Nombre: " + resultado.getString("nombre") +

", Precio por unidad: " + resultado.getDouble("precio\_unitario") + ", Stock: " + resultado.getString("stock"));

}

System.out.println();

stmt.close();

conexion.close();

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void modificarArticulo(int id, String nombre, double precio, int stock) { //Modifica los datos del artículo indicado

Connection conexion = miConexion.conectar();

try {

String sql = "UPDATE articulos SET nombre = ?, precio\_unitario = ?, stock = ? WHERE id\_articulo = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setString(1, nombre);

pstmt.setDouble(2, precio);

pstmt.setInt(3, stock);

pstmt.setInt(4, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Artículo modificado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void eliminarArticulo(int id) { //Elimina los datos del articulo indicado

Connection conexion = miConexion.conectar();

try {

String sql = "DELETE FROM articulos WHERE id\_articulo = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Artículo eliminado exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

}

VistaArticulo

package vista;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorArticulo;

public class VistaArticulo {

static String saltoDeLinea;

public void menuArticulo() { // Muestra las opciones de Crud de la tabla articulos

Scanner obj = new Scanner(System.in);

ControladorArticulo c = new ControladorArticulo();

do {

try {

System.out.println("1. Registrar artículo");

System.out.println("2. Listar artículos");

System.out.println("3. Modificar artículo");

System.out.println("4. Eliminar artículo");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

if (opcion == 1) { //Registra un nuevo artículo

System.out.println("Introduce el nombre del artículo:");

String nombre = obj.nextLine();

System.out.println();

System.out.println("Introduce el precio del artículo:");

Double precio = obj.nextDouble();

System.out.println();

System.out.println("Introduce el stock del artículo:");

int stock = obj.nextInt();

System.out.println();

c.registrarArticulo(nombre, precio, stock);

} else if (opcion == 2) { //Muestra los datos de los artículos

c.listarArticulos();

} else if (opcion == 3) { //Modifica los datos de un artículo

System.out.println("Introduce el id del artículo a modificar:");

int id = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el nuevo nombre del artículo:");

String nombre = obj.nextLine();

System.out.println();

System.out.println("Introduce el nuevo precio del artículo:");

Double precio = obj.nextDouble();

System.out.println();

System.out.println("Introduce el nuevo stock del artículo:");

int stock = obj.nextInt();

System.out.println();

c.modificarArticulo(id, nombre, precio, stock);

} else if (opcion == 4) { //Elimina los datos de un artículo

System.out.println("Introduce el id del artículo a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarArticulo(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

ControladorFacturas

package vista;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorArticulo;

public class VistaArticulo {

static String saltoDeLinea;

public void menuArticulo() { // Muestra las opciones de Crud de la tabla articulos

Scanner obj = new Scanner(System.in);

ControladorArticulo c = new ControladorArticulo();

do {

try {

System.out.println("1. Registrar artículo");

System.out.println("2. Listar artículos");

System.out.println("3. Modificar artículo");

System.out.println("4. Eliminar artículo");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

if (opcion == 1) { //Registra un nuevo artículo

System.out.println("Introduce el nombre del artículo:");

String nombre = obj.nextLine();

System.out.println();

System.out.println("Introduce el precio del artículo:");

Double precio = obj.nextDouble();

System.out.println();

System.out.println("Introduce el stock del artículo:");

int stock = obj.nextInt();

System.out.println();

c.registrarArticulo(nombre, precio, stock);

} else if (opcion == 2) { //Muestra los datos de los artículos

c.listarArticulos();

} else if (opcion == 3) { //Modifica los datos de un artículo

System.out.println("Introduce el id del artículo a modificar:");

int id = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce el nuevo nombre del artículo:");

String nombre = obj.nextLine();

System.out.println();

System.out.println("Introduce el nuevo precio del artículo:");

Double precio = obj.nextDouble();

System.out.println();

System.out.println("Introduce el nuevo stock del artículo:");

int stock = obj.nextInt();

System.out.println();

c.modificarArticulo(id, nombre, precio, stock);

} else if (opcion == 4) { //Elimina los datos de un artículo

System.out.println("Introduce el id del artículo a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarArticulo(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

VistaFactura

package vista;

import java.sql.Date;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorFacturas;

public class VistaFactura {

static String saltoDeLinea;

public void menuFactura() { // Muestra las opciones de CRUD de la tabla facturas

Scanner obj = new Scanner(System.in);

ControladorFacturas c = new ControladorFacturas();

do {

try {

System.out.println("1. Registrar factura");

System.out.println("2. Listar facturas");

System.out.println("3. Modificar facturas");

System.out.println("4. Eliminar factura");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

if (opcion == 1) { //Registra una nueva factura

System.out.println("Introduce el ID del proveedor:");

int id = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce la fecha de la factura:");

String fechaTexto = obj.nextLine();

Date fecha = java.sql.Date.valueOf(fechaTexto);

System.out.println();

System.out.println("Introduce el precio total de la factura:");

Double precio = obj.nextDouble();

System.out.println();

c.registrarFactura(id, fecha, precio);

} else if (opcion == 2) { //Muestra los datos de las facturas

c.listarFacturas();

} else if (opcion == 3) { //Modifica los datos de una factura

System.out.println("Introduce el id de la factura a modificar:");

int id\_factura = obj.nextInt();

System.out.println();

System.out.println("Introduce el nuevo id del proveedor la factura:");

int id\_proveedor = obj.nextInt();

System.out.println();

System.out.println("Introduce la nueva fecha de la factura:");

String fechaTexto = obj.nextLine();

Date fecha = java.sql.Date.valueOf(fechaTexto);

System.out.println();

System.out.println("Introduce el nuevo precio total de la factura:");

Double precio = obj.nextDouble();

System.out.println();

c.modificarFactura(id\_factura, id\_proveedor, fecha, precio);

} else if (opcion == 4) { //Elimina los datos de una factura

System.out.println("Introduce el id de la factura a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarFactura(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

ControladorVentas

package controlador;

import config.\*;

import java.sql.\*;

import java.sql.Date;

public class ControladorVentas {

Conexion miConexion;

public ControladorVentas() {

miConexion = new Conexion();

}

public void registrarVenta(int id\_cliente, int id\_articulo, int cantidad, Date fecha) { //Registra los datos de una nueva venta

Connection conexion = miConexion.conectar();

try {

String sql = "INSERT INTO ventas (id\_cliente, id\_articulo, cantidad, fecha\_venta) VALUES (?, ?, ?, ?)";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id\_cliente);

pstmt.setInt(2, id\_articulo);

pstmt.setInt(3, cantidad);

pstmt.setDate(4, fecha);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Venta registrada exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void listarVentas() { //Muestra los datos de todas las ventas registradas

Connection conexion = miConexion.conectar();

try {

Statement stmt = conexion.createStatement();

ResultSet resultado = stmt.executeQuery("SELECT \* FROM ventas");

while (resultado.next()) {

System.out.println("ID de la venta: " + resultado.getInt("id\_venta") + ", ID del cliente: " + resultado.getString("id\_cliente") +

", ID del artículo: " + resultado.getString("id\_articulo") + ", Cantidad del artículo: " + resultado.getString("cantidad") +

", Fecha de venta: " + resultado.getString("fecha\_venta"));

}

System.out.println();

stmt.close();

conexion.close();

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void modificarVenta(int id\_venta, int id\_cliente, int id\_articulo, int cantidad, java.sql.Date fecha) { //Modifica los datos de la venta indicada

Connection conexion = miConexion.conectar();

try {

String sql = "UPDATE ventas SET id\_cliente = ?, id\_articulo = ?, cantidad = ?, fecha = ? WHERE id\_venta = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id\_cliente);

pstmt.setInt(2, id\_articulo);

pstmt.setInt(3, cantidad);

pstmt.setDate(4, fecha);

pstmt.setInt(5, id\_venta);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Factura modificada exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

public void eliminarVenta(int id) { //Elimina los datos de la venta indicada

Connection conexion = miConexion.conectar();

try {

String sql = "DELETE FROM ventas WHERE id\_venta = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id);

pstmt.executeUpdate();

pstmt.close();

conexion.close();

System.out.println("Venta eliminada exitosamente");

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

}

VistaVenta

package vista;

import java.sql.Date;

import java.util.InputMismatchException;

import java.util.Scanner;

import controlador.ControladorVentas;

public class VistaVenta {

static String saltoDeLinea;

public void menuVenta() { // Muestra las opciones de CRUD de la tabla ventas

Scanner obj = new Scanner(System.in);

ControladorVentas c = new ControladorVentas();

do {

try {

System.out.println("1. Registrar venta");

System.out.println("2. Listar ventas");

System.out.println("3. Modificar venta");

System.out.println("4. Eliminar venta");

System.out.println("5. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

if (opcion == 1) { //Registra una nueva venta

System.out.println("Introduce el ID del cliente:");

int id\_cliente = obj.nextInt();

System.out.println();

System.out.println("Introduce el ID del artículo:");

int id\_articulo = obj.nextInt();

System.out.println();

System.out.println("Introduce la cantidad del artículo:");

int cantidad = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce la fecha de la venta:");

String fechaTexto = obj.nextLine();

Date fecha = java.sql.Date.valueOf(fechaTexto);

System.out.println();

c.registrarVenta(id\_cliente, id\_articulo, cantidad, fecha);

} else if (opcion == 2) { //Muestra los datos de las facturas

c.listarVentas();

} else if (opcion == 3) { //Modifica los datos de una factura

System.out.println("Introduce el id de la venta a modificar:");

int id\_venta = obj.nextInt();

System.out.println();

System.out.println("Introduce el nuevo ID del cliente:");

int id\_cliente = obj.nextInt();

System.out.println();

System.out.println("Introduce el nuevo ID del artículo:");

int id\_articulo = obj.nextInt();

System.out.println();

System.out.println("Introduce la nueva cantidad del artículo:");

int cantidad = obj.nextInt();

System.out.println();

saltoDeLinea = obj.nextLine();

System.out.println("Introduce la nueva fecha de la venta:");

String fechaTexto = obj.nextLine();

Date fecha = java.sql.Date.valueOf(fechaTexto);

System.out.println();

c.modificarVenta(id\_venta, id\_cliente, id\_articulo, cantidad, fecha);

} else if (opcion == 4) { //Elimina los datos de una venta

System.out.println("Introduce el id de la venta a eliminar:");

int id = obj.nextInt();

System.out.println();

c.eliminarVenta(id);

} else if (opcion == 5) { //Vuelve al menú principal

System.out.println("Volviendo al menú principal");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

} while(true);

obj.close();

}

}

ControladorInforme

package controlador;

import config.\*;

import java.sql.\*;

public class ControladorInforme {

Conexion miConexion;

public ControladorInforme() {

miConexion = new Conexion();

}

public void listarVentas(int id) { //Muestra los datos las ventas registradas del cliente indicado

Connection conexion = miConexion.conectar();

try {

String sql = "SELECT \* FROM ventas WHERE id\_cliente = ?";

PreparedStatement pstmt = conexion.prepareStatement(sql);

pstmt.setInt(1, id);

ResultSet resultado = pstmt.executeQuery("SELECT \* FROM ventas WHERE id\_cliente = ?");

while (resultado.next()) {

System.out.println("ID de la venta: " + resultado.getInt("id\_venta") + ", ID del cliente: " + resultado.getString("id\_cliente") +

", ID del artículo: " + resultado.getString("id\_articulo") + ", Cantidad del artículo: " + resultado.getString("cantidad") +

", Fecha de venta: " + resultado.getString("fecha\_venta"));

}

System.out.println();

pstmt.close();

conexion.close();

} catch (Exception e) {

System.out.println("Error al conectarse: " + e.getMessage());

}

}

}

VistaInforme

package vista;

import java.util.Scanner;

import controlador.ControladorInforme;

public class VistaInforme {

public void buscarInforme() {

Scanner obj = new Scanner(System.in);

ControladorInforme c = new ControladorInforme();

System.out.println("Introduce el ID del cliente:");

int id\_cliente = obj.nextInt();

System.out.println();

c.listarVentas(id\_cliente);

obj.close();

}

}

Principal

package principal;

import vista.\*;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Principal {

public static void main(String[] args) {

Scanner obj = new Scanner(System.in);

do {

try {

System.out.println("1. Clientes");

System.out.println("2. Proveedores");

System.out.println("3. Artículos");

System.out.println("4. Facturas");

System.out.println("5. Ventas");

System.out.println("6. Informes de venta por cliente");

System.out.println("7. Salir");

System.out.println();

System.out.println("Escoje una opción:");

int opcion = obj.nextInt();

System.out.println();

if (opcion == 1) { //Va al menu de clientes

VistaCliente vc = new VistaCliente();

vc.menuCliente();

} else if (opcion == 2) { //Va al menu de proveedores

VistaProveedor vp = new VistaProveedor();

vp.menuProveedor();

} else if (opcion == 3) { //Va al menu de artículos

VistaArticulo va = new VistaArticulo();

va.menuArticulo();

} else if (opcion == 4) { //Va al menu de facturas

VistaFactura vf = new VistaFactura();

vf.menuFactura();

} else if (opcion == 5) { //Va al menu de ventas

VistaVenta vv = new VistaVenta();

vv.menuVenta();

} else if (opcion == 6) { //Va al menu de informe de ventas

VistaInforme vi = new VistaInforme();

vi.buscarInforme();

} else if (opcion == 7) { //Sale del programa

System.out.println("Saliendo del programa");

break;

} else { //Mensaje que aparece al escoger una opción inválida

System.out.println("Opción inválida");

}

} catch(InputMismatchException e) {

System.out.println("Tipo de dato incorrecto");

}

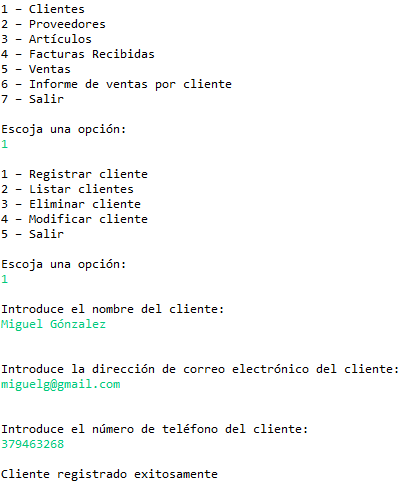
} while(true);

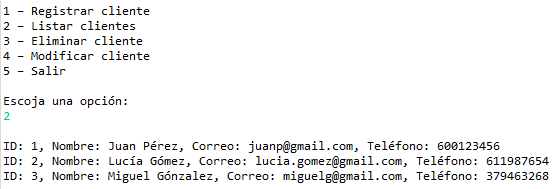
obj.close();

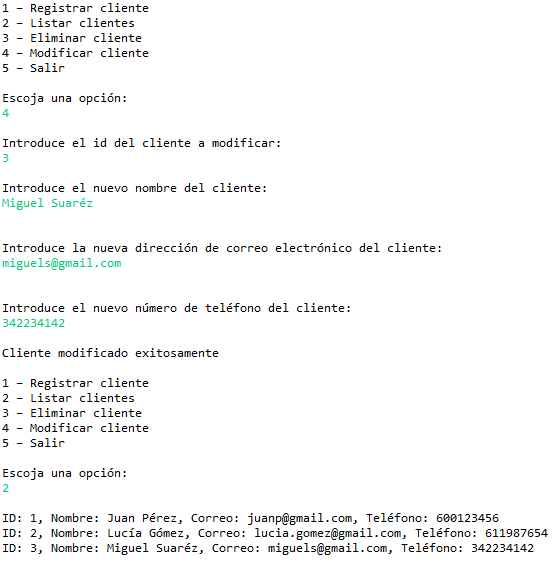
}

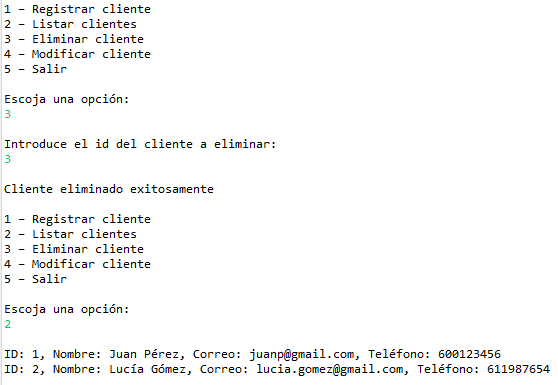
}

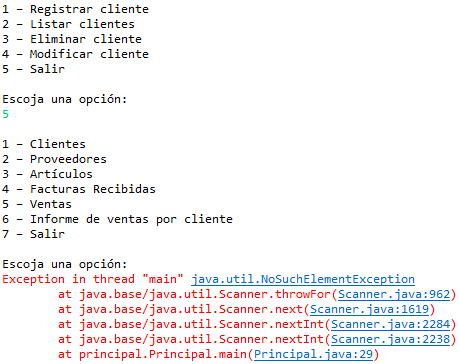
Comprobación











Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>