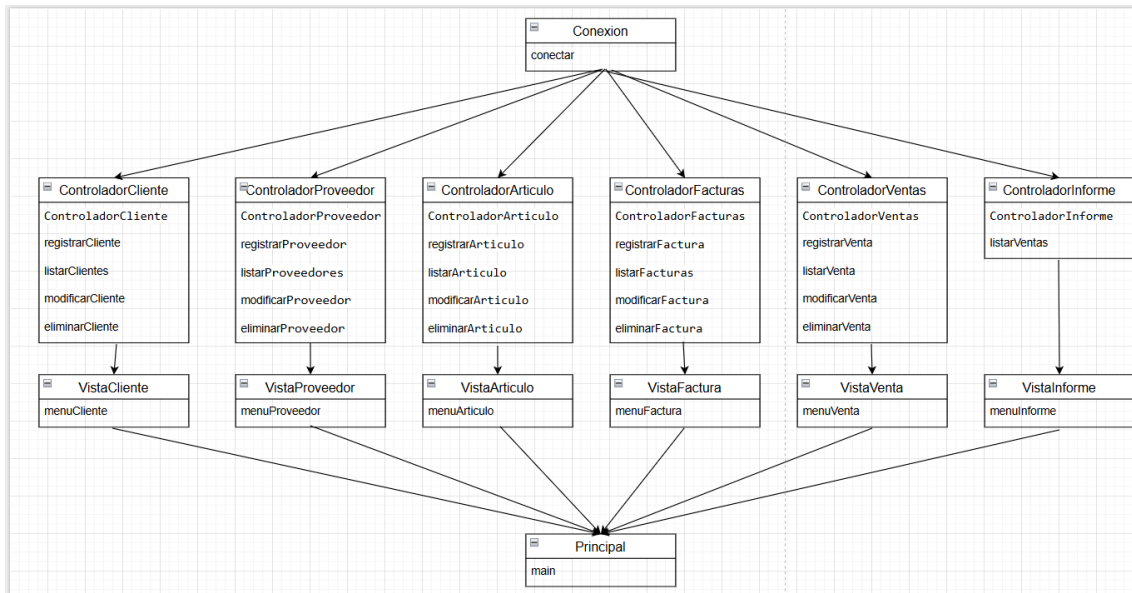


Ejercicios Java 9

Diagrama de clases



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();

            } catch (InputMismatchException e) {
                System.out.println("Error: No se puede ingresar un valor no entero.");
            }

        } while (opcion != 5);
    }
}
```

```

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);
artículos	} else if (opcion == 2) { //Muestra los datos de los
	c.listarArticulos();
artículo	} else if (opcion == 3) { //Modifica los datos de un
artículo a modificar:");	System.out.println("Introduce el id del
	int id = obj.nextInt();
	System.out.println();
	saltoDeLinea = obj.nextLine();
nombre del artículo:");	System.out.println("Introduce el nuevo
	String nombre = obj.nextLine();
	System.out.println();
precio del artículo:");	System.out.println("Introduce el nuevo
	Double precio = obj.nextDouble();
	System.out.println();
stock del artículo:");	System.out.println("Introduce el nuevo
	int stock = obj.nextInt();
	System.out.println();

```

stock);

c.modificarArticulo(id, nombre, precio,

artículo

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

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
```

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

```
String nombre = obj.nextLine();
System.out.println();
```

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

```
Double precio = obj.nextDouble();
System.out.println();
```

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

```
int stock = obj.nextInt();
System.out.println();
```

```
c.registrarArticulo(nombre, precio, stock);
```

artículos	<pre> } else if (opcion == 2) { //Muestra los datos de los </pre>
	<pre> c.listarArticulos(); </pre>
artículo	<pre> } else if (opcion == 3) { //Modifica los datos de un </pre>
artículo a modificar:");	<pre> System.out.println("Introduce el id del </pre>
	<pre> int id = obj.nextInt(); System.out.println(); saltoDeLinea = obj.nextLine(); </pre>
nombre del artículo:");	<pre> System.out.println("Introduce el nuevo </pre>
	<pre> String nombre = obj.nextLine(); System.out.println(); </pre>
precio del artículo:");	<pre> System.out.println("Introduce el nuevo </pre>
	<pre> Double precio = obj.nextDouble(); System.out.println(); </pre>
stock del artículo:");	<pre> System.out.println("Introduce el nuevo </pre>
	<pre> int stock = obj.nextInt(); System.out.println(); </pre>
stock);	<pre> c.modificarArticulo(id, nombre, precio, </pre>

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

                                        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);

```

facturas	<pre> } else if (opcion == 2) { //Muestra los datos de las </pre>
	<pre> c.listarFacturas(); </pre>
factura	<pre> } else if (opcion == 3) { //Modifica los datos de una </pre>
factura a modificar:");	<pre> System.out.println("Introduce el id de la </pre>
	<pre> int id_factura = obj.nextInt(); System.out.println(); </pre>
del proveedor la factura:");	<pre> System.out.println("Introduce el nuevo id </pre>
	<pre> int id_proveedor = obj.nextInt(); System.out.println(); </pre>
fecha de la factura:");	<pre> System.out.println("Introduce la nueva </pre>
	<pre> String fechaTexto = obj.nextLine(); Date fecha = </pre>
java.sql.Date.valueOf(fechaTexto);	<pre> System.out.println(); </pre>
precio total de la factura:");	<pre> System.out.println("Introduce el nuevo </pre>
	<pre> Double precio = obj.nextDouble(); System.out.println(); </pre>
id_proveedor, fecha, precio);	<pre> c.modificarFactura(id_factura, </pre>

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

                                        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:");
```

```

del artículo:");

int id_articulo = obj.nextInt();
System.out.println();

System.out.println("Introduce la cantidad

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();

```

```

del cliente:");

System.out.println("Introduce el nuevo ID

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

- 1 - Clientes
- 2 - Proveedores
- 3 - Artículos
- 4 - Facturas Recibidas
- 5 - Ventas
- 6 - Informe de ventas por cliente
- 7 - Salir

Escoja una opción:

1

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

1

Introduce el nombre del cliente:

Miguel González

Introduce la dirección de correo electrónico del cliente:

miguelg@gmail.com

Introduce el número de teléfono del cliente:

379463268

Cliente registrado exitosamente

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

2

ID: 1, Nombre: Juan Pérez, Correo: juanp@gmail.com, Teléfono: 600123456

ID: 2, Nombre: Lucía Gómez, Correo: lucia.gomez@gmail.com, Teléfono: 611987654

ID: 3, Nombre: Miguel González, Correo: miguelg@gmail.com, Teléfono: 379463268

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

4

Introduce el id del cliente a modificar:

3

Introduce el nuevo nombre del cliente:

Miguel Suárez

Introduce la nueva dirección de correo electrónico del cliente:

miguels@gmail.com

Introduce el nuevo número de teléfono del cliente:

342234142

Cliente modificado exitosamente

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

2

ID: 1, Nombre: Juan Pérez, Correo: juanp@gmail.com, Teléfono: 600123456

ID: 2, Nombre: Lucía Gómez, Correo: lucia.gomez@gmail.com, Teléfono: 611987654

ID: 3, Nombre: Miguel Suárez, Correo: miguels@gmail.com, Teléfono: 342234142

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

3

Introduce el id del cliente a eliminar:

3

Cliente eliminado exitosamente

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

2

ID: 1, Nombre: Juan Pérez, Correo: juanp@gmail.com, Teléfono: 600123456

ID: 2, Nombre: Lucía Gómez, Correo: lucia.gomez@gmail.com, Teléfono: 611987654

- 1 - Registrar cliente
- 2 - Listar clientes
- 3 - Eliminar cliente
- 4 - Modificar cliente
- 5 - Salir

Escoja una opción:

5

- 1 - Clientes
- 2 - Proveedores
- 3 - Artículos
- 4 - Facturas Recibidas
- 5 - Ventas
- 6 - Informe de ventas por cliente
- 7 - Salir

Escoja una opción:

```
Exception in thread "main" java.util.NoSuchElementException
    at java.base/java.util.Scanner.throwFor(Scanner.java:962)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2284)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2238)
    at principal.Principal.main(Principal.java:29)
```

```
1. Clientes
2. Proveedores
3. Articulos
4. Facturas
5. Ventas
6. Informes de venta por cliente
7. Salir

Escoje una opción:
6

Introduce el ID del cliente:
1

Error al conectarse: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
Error al conectarse: Cannot invoke "java.sql.Connection.prepareStatement(String)" because "conexion" is null
1. Clientes
2. Proveedores
3. Articulos
4. Facturas
5. Ventas
6. Informes de venta por cliente
7. Salir

Escoje una opción:
Exception in thread "main" java.util.NoSuchElementException
    at java.base/java.util.Scanner.throwFor(Scanner.java:962)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2284)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2238)
    at JAVAP00/principal.Principal.main(Principal.java:27)
```

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