



CARATULA



INTEGRANTE:

- JOSUÉ JESÚS TORRES SANCHEZ

BLOQUE: 304

ID: 001559028

PROF: GIANCARLOS BARBOZA ANIETO

CARRERA: ING DE SOFTWARE CON IA



BASE DE DATOS – HEIDISQL

Creación de base de datos y creación de tablas

```
CREATE DATABASE GESTION_VENTAS
USE GESTION_VENTAS

CREATE TABLE PRODUCTO(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    CODIGO CHAR(8) NOT NULL,
    NOMBRE VARCHAR(40) NOT NULL,
    MARCA VARCHAR(40) NOT NULL,
    MODELO VARCHAR(40) NOT NULL,
    PRECIO DOUBLE NOT NULL,
    CANTIDAD INT NOT NULL
)
```

```
TRUNCATE TABLE PRODUCTO
SELECT * FROM PRODUCTO
```

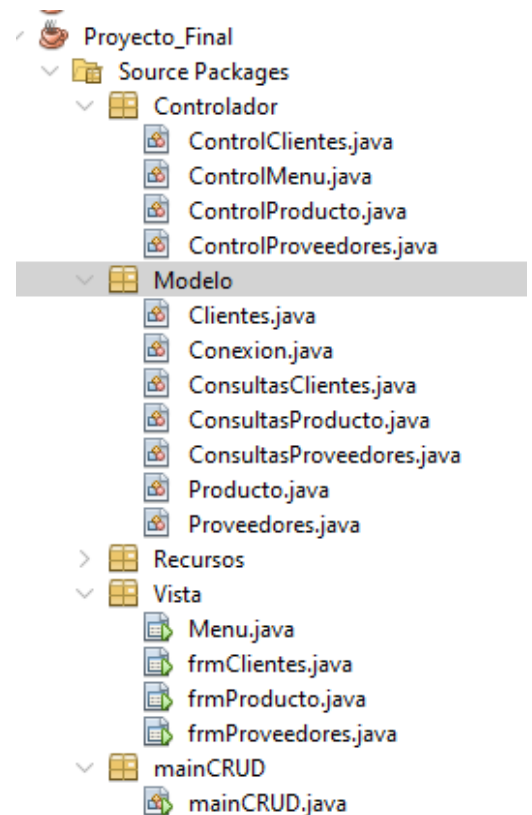
```
CREATE TABLE CLIENTES(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    DESCRIPCION VARCHAR(50) NOT NULL,
    DIRECCION VARCHAR(100) NOT NULL,
    TELEFONO CHAR(9) NOT NULL,
    RUC CHAR(11) NOT NULL
)
```

```
TRUNCATE TABLE clientes
SELECT * FROM CLIENTES
```

```
CREATE TABLE PROVEEDORES(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    DESCRIPCION VARCHAR(50) NOT NULL,
    DIRECCION VARCHAR(100) NOT NULL,
    TELEFONO CHAR(9) NOT NULL
)

TRUNCATE TABLE PROVEEDORES;
SELECT * FROM PROVEEDORES
```

Carpeta - Proyecto Final



JAVA – MAIN

```
package mainCRUD;
```

```
import Controlador.ControlMenu;
```

```
import Vista.Menu;
```

```
public class mainCRUD {
```

```
    public static void main(String[] args) {
```

```
        Menu vista = new Menu();
```

```
        ControlMenu ctrl = new ControlMenu(vista);
```

```
        ctrl.iniciar();
```

```
        vista.setVisible(true);
```

```
    }
```

```
}
```

JAVA – CLIENTES (ALL)

1. Clientes.java

```
package Modelo;
public class Clientes {
    private int id;
    private String descripcion, direccion, telefono, ruc;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }

    public String getRuc() {
        return ruc;
    }

    public void setRuc(String ruc) {
        this.ruc = ruc;
    }
}
```

2. ConsultasClientes.java

```
package Modelo;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.ResultSet;

public class ConsultasClientes extends Conexion{
    public boolean Guardar(Clientes cli){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "INSERT INTO CLIENTES (descripcion, direccion, telefono, ruc) VALUES (?, ?, ?, ?)";

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, cli.getDescripcion());
            ps.setString(2, cli.getDireccion());
            ps.setString(3, cli.getTelefono());
            ps.setString(4, cli.getRuc());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }

    public boolean Editar(Clientes cli){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "UPDATE clientes SET descripcion = ?, direccion = ?, telefono = ?, ruc = ? WHERE id = ?";

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, cli.getDescripcion());
            ps.setString(2, cli.getDireccion());
            ps.setString(3, cli.getTelefono());
            ps.setString(4, cli.getRuc());
            ps.setInt(5, cli.getId());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }
}
```

```

public boolean Eliminar(Clientes cli){
    PreparedStatement ps = null;
    Connection con = getConexion();
    String sql = "DELETE FROM clientes WHERE id = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setInt(1, cli.getId());
        ps.execute();
        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}

public boolean Buscar(Clientes cli){
    PreparedStatement ps = null;
    Connection con = getConexion();
    ResultSet rs = null;
    String sql = "SELECT * FROM clientes WHERE id = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setInt(1, cli.getId());
        rs = ps.executeQuery();

        if (rs.next()){
            cli.setId(Integer.parseInt(rs.getString("id")));
            cli.setDescripcion(rs.getString("descripcion"));
            cli.setDireccion(rs.getString("direccion"));
            cli.setTelefono(rs.getString("telefono"));
            cli.setRuc(rs.getString("ruc"));
        } else {
        }

        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
}

```

3. ControlCliente.java

```
package Controlador;

import Modelo.Clientes;
import Modelo.ConsultasClientes;
import Vista.frmClientes;
import Vista.Menu;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ControlClientes implements ActionListener {
    // Variables para cada paquete
    private final Clientes modelo;
    private final ConsultasClientes consultas;
    private final frmClientes vista;
    private final Menu vistaMenu;

    public ControlClientes(Clientes modelo, ConsultasClientes consultas, frmClientes vista, Menu vistaMenu){
        this.modelo = modelo;
        this.consultas = consultas;
        this.vista = vista;
        this.vistaMenu = vistaMenu;

        // Conectar botones del formulario
        this.vista.btnGuardar.addActionListener(this);
        this.vista.btnNuevo.addActionListener(this);
        this.vista.btnEliminar.addActionListener(this);
        this.vista.btnBuscar.addActionListener(this);
        this.vista.btnEditar.addActionListener(this);
        this.vista.btnCancelar.addActionListener(this);
        this.vista.btnMenu.addActionListener(this);
    }

    public void iniciar(){
        vista.setTitle("Clientes");
        vista.setLocationRelativeTo(null);
        vista.setVisible(true);
        cargarTabla(); // Cargar la tabla al iniciar la vista
    }
}
```

```

// Método para cargar los datos en el JTable1
public void cargarTabla() {
    DefaultTableModel modeloTabla = (DefaultTableModel) vista.tClientes.getModel();
    modeloTabla.setRowCount(0); // Limpiar la tabla antes de cargar nuevos datos

    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    String sql = "SELECT * FROM clientes"; // Consulta SQL para obtener todos los datos

    try {
        con = consultas.getConexion(); // Obtener la conexión de la clase ConsultasClientes
        ps = con.prepareStatement(sql);
        rs = ps.executeQuery();

        // Recorrer los resultados y agregarlos al JTable
        while (rs.next()) {
            Object[] columna = new Object[5]; // Número de columnas en la tabla clientes
            columna[0] = rs.getInt("id");
            columna[1] = rs.getString("descripcion");
            columna[2] = rs.getString("direccion");
            columna[3] = rs.getString("telefono");
            columna[4] = rs.getString("ruc");
            modeloTabla.addRow(columna); // Agregar fila al modelo de la tabla
        }

    } catch (SQLException e) {
        System.err.println(e);
    } finally {
        try {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (con != null) con.close();
        } catch (SQLException e) {
            System.err.println(e);
        }
    }
}

```



```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == vista.btnGuardar) {
        modelo.setDescripcion(vista.txtDescripcion.getText());
        modelo.setDireccion(vista.txtDireccion.getText());
        modelo.setTelefono(vista.txtTelefono.getText());
        modelo.setRuc(vista.txtRuc.getText());

        if(consultas.Guardar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro guardado exitosamente");
            cargarTabla(); // Actualizar el JTable después de guardar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se guardó los datos");
        }
    }

    if (e.getSource() == vista.btnEditar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        modelo.setDescripcion(vista.txtDescripcion.getText());
        modelo.setDireccion(vista.txtDireccion.getText());
        modelo.setTelefono(vista.txtTelefono.getText());
        modelo.setRuc(vista.txtRuc.getText());

        if(consultas.Editar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro actualizado exitosamente");
            cargarTabla(); // Actualizar la tabla después de editar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se pudo actualizar los datos");
        }
    }

    if (e.getSource() == vista.btnEliminar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        if(consultas.Eliminar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro eliminado exitosamente");
            cargarTabla(); // Actualizar la tabla después de eliminar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se eliminó los datos");
        }
    }
}
```

```

if (e.getSource() == vista.btnBuscar) {
    modelo.setId(Integer.parseInt(vista.txtId.getText()));

    if(consultas.Buscar(modelo)){
        vista.txtId.setText(String.valueOf(modelo.getId()));
        vista.txtDescripcion.setText(modelo.getDescripcion());
        vista.txtDireccion.setText(modelo.getDireccion());
        vista.txtTelefono.setText(modelo.getTelefono());
        vista.txtRuc.setText(modelo.getRuc());
    } else {
        JOptionPane.showMessageDialog(null, "No se encontró datos");
    }
}

if(e.getSource() == vista.btnCancelar){
    Limpiar();
}

if(e.getSource() == vista.btnNuevo){
    Limpiar();
}

if(e.getSource() == vista.btnMenu){
    vista.setVisible(false);
    vistaMenu.setVisible(true);
}
}

public void Limpiar(){
    vista.txtId.setText(null);
    vista.txtDescripcion.setText(null);
    vista.txtDireccion.setText(null);
    vista.txtTelefono.setText(null);
    vista.txtRuc.setText(null);
}
}

```

4. frmClientes.java

Vista\frmClientes.java ×

Source Design History


FORMULARIO DE CLIENTES


DATOS DEL CLIENTE: MENU


ID


DESCRIPCIÓN TELÉFONO


DIRECCIÓN RUC



NUEVO


GUARDAR


EDITAR


ELIMINAR


BUSCAR


CANCELAR

ID	Desripición	Dirección	Teléfono	RUC
----	-------------	-----------	----------	-----

JAVA – PROVEEDORES (ALL)

1. Proveedores.java

```
package Modelo;
public class Proveedores {
    private int id;
    private String descripcion, direccion, telefono;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getDireccion() {
        return direccion;
    }

    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}
```

2. ConsultasProveedores.java

```
package Modelo;

import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.ResultSet;

public class ConsultasProducto extends Conexion{

    public boolean Guardar(Producto pro){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "INSERT INTO PRODUCTO (codigo, nombre, marca, modelo, precio, cantidad) VALUES (?, ?, ?, ?, ?, ?)";

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, pro.getCodigo());
            ps.setString(2, pro.getNombre());
            ps.setString(3, pro.getMarca());
            ps.setString(4, pro.getModelo());
            ps.setDouble(5, pro.getPrecio());
            ps.setInt(6, pro.getCantidad());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }
}
```

```
    public boolean Modificar(Producto pro){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "UPDATE producto SET codigo = ?, nombre = ?, marca = ?, modelo = ?, precio = ?, cantidad = ? WHERE id = ?";

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, pro.getCodigo());
            ps.setString(2, pro.getNombre());
            ps.setString(3, pro.getMarca());
            ps.setString(4, pro.getModelo());
            ps.setDouble(5, pro.getPrecio());
            ps.setInt(6, pro.getCantidad());
            ps.setInt(7, pro.getId());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }

    public boolean Eliminar(Producto pro){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "DELETE FROM producto WHERE id = ?";

        try {
            ps = con.prepareStatement(sql);
            ps.setInt(1, pro.getId());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }
}
```

```

public boolean Buscar(Producto pro) {
    PreparedStatement ps = null;
    Connection con = getConexion();
    ResultSet rs = null;
    String sql = "SELECT * FROM producto WHERE codigo = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setString(1, pro.getCodigo());
        rs = ps.executeQuery();

        if (rs.next()) {
            pro.setId(Integer.parseInt(rs.getString("id")));
            pro.setCodigo(rs.getString("codigo"));
            pro.setNombre(rs.getString("nombre"));
            pro.setMarca(rs.getString("marca"));
            pro.setModelo(rs.getString("modelo"));
            pro.setPrecio(Double.parseDouble(rs.getString("precio")));
            pro.setCantidad(Integer.parseInt(rs.getString("cantidad")));
        } else {
        }

        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
}

```

3. ControlProveedores.java

```
package Controlador;

import Modelo.Proveedores;
import Modelo.ConsultasProveedores;
import Vista.Menu;
import Vista.frmProveedores;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ControlProveedores implements ActionListener {
    // Variables para cada paquete
    private final Proveedores modelo;
    private final ConsultasProveedores consultas;
    private final frmProveedores vista;
    private final Menu vistaMenu;

    public ControlProveedores(Proveedores modelo, ConsultasProveedores consultas, frmProveedores vista, Menu vistaMenu){
        this.modelo = modelo;
        this.consultas = consultas;
        this.vista = vista;
        this.vistaMenu = vistaMenu;

        // Conectar botones del formulario
        this.vista.btnGuardar.addActionListener(this);
        this.vista.btnNuevo.addActionListener(this);
        this.vista.btnEliminar.addActionListener(this);
        this.vista.btnBuscar.addActionListener(this);
        this.vista.btnEditar.addActionListener(this);
        this.vista.btnCancelar.addActionListener(this);
        this.vista.btnMenu.addActionListener(this);
    }

    public void iniciar(){
        vista.setTitle("Proveedores");
        vista.setLocationRelativeTo(null);
        vista.setVisible(true);
        cargarTabla(); // Cargar la tabla al iniciar la vista
    }
}
```

```

// Método para cargar los datos en el JTable1
public void cargarTabla() {
    DefaultTableModel modeloTabla = (DefaultTableModel) vista.tProveedores.getModel();
    modeloTabla.setRowCount(0); // Limpiar la tabla antes de cargar nuevos datos

    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    String sql = "SELECT * FROM PROVEEDORES"; // Consulta SQL para obtener todos los datos

    try {
        con = consultas.getConexion(); // Obtener la conexión de la clase ConsultasClientes
        ps = con.prepareStatement(sql);
        rs = ps.executeQuery();

        // Recorrer los resultados y agregarlos al JTable
        while (rs.next()) {
            Object[] columna = new Object[4]; // Número de columnas en la tabla clientes
            columna[0] = rs.getInt("id");
            columna[1] = rs.getString("descripcion");
            columna[2] = rs.getString("direccion");
            columna[3] = rs.getString("telefono");
            modeloTabla.addRow(columna); // Agregar fila al modelo de la tabla
        }

    } catch (SQLException e) {
        System.err.println(e);
    } finally {
        try {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (con != null) con.close();
        } catch (SQLException e) {
            System.err.println(e);
        }
    }
}

```



```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == vista.btnGuardar) {
        modelo.setDescripcion(vista.txtDescripcion.getText());
        modelo.setDireccion(vista.txtDireccion.getText());
        modelo.setTelefono(vista.txtTelefono.getText());

        if(consultas.Guardar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro guardado exitosamente");
            cargarTabla(); // Actualizar el JTable después de guardar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se guardó los datos");
        }
    }

    if (e.getSource() == vista.btnEditar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        modelo.setDescripcion(vista.txtDescripcion.getText());
        modelo.setDireccion(vista.txtDireccion.getText());
        modelo.setTelefono(vista.txtTelefono.getText());

        if(consultas.Editar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro actualizado exitosamente");
            cargarTabla(); // Actualizar la tabla después de editar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se pudo actualizar los datos");
        }
    }

    if (e.getSource() == vista.btnEliminar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        if(consultas.Eliminar(modelo)){
            JOptionPane.showMessageDialog(null, "Registro eliminado exitosamente");
            cargarTabla(); // Actualizar la tabla después de eliminar
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se eliminó los datos");
        }
    }
}

```

```

if (e.getSource() == vista.btnBuscar) {
    modelo.setId(Integer.parseInt(vista.txtId.getText()));

    if(consultas.Buscar(modelo)) {
        vista.txtId.setText(String.valueOf(modelo.getId()));
        vista.txtDescripcion.setText(modelo.getDescripcion());
        vista.txtDireccion.setText(modelo.getDireccion());
        vista.txtTelefono.setText(modelo.getTelefono());
    } else {
        JOptionPane.showMessageDialog(null, "No se encontró datos");
    }
}

if(e.getSource() == vista.btnCancelar){
    Limpiar();
}

if(e.getSource() == vista.btnNuevo){
    Limpiar();
}

if(e.getSource() == vista.btnMenu){
    vista.setVisible(false);
    vistaMenu.setVisible(true);
}
}

public void Limpiar(){
    vista.txtId.setText(null);
    vista.txtDescripcion.setText(null);
    vista.txtDireccion.setText(null);
    vista.txtTelefono.setText(null);
}
}

```

4. frmProveedores.java

Modelo\Proveedores.java x Modelo\ConsultasProducto.java x Controlador\ControlProveedores.java x Vista\frmProveedores.java x

Source Design History


FORMULARIO DE PROVEEDORES


DATOS DEL PROVEEDOR: MENU


ID


DESCRIPCIÓN


DIRECCIÓN TELÉFONO



NUEVO


GUARDAR


EDITAR


ELIMINAR


BUSCAR


CANCELAR

ID	Desripción	Dirección	Teléfono
----	------------	-----------	----------

JAVA – PRODUCTO (ALL)

1. Producto.java

```
package Modelo;
public class Producto {
    /* Aqui creamos los atributos, los mismos campos que tenemos en HeidiSQ.
    Con el mismo tipo de dato.
    */
    private int id;
    private String codigo, nombre, marca, modelo;
    private double precio;
    private int cantidad;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModelo() {
        return modelo;
    }

    public void setModelo(String modelo) {
        this.modelo = modelo;
    }

    public double getPrecio() {
        return precio;
    }

    public void setPrecio(double precio) {
        this.precio = precio;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }
}
```

2. ConsultasProducto.java

```
package Modelo;
import java.sql.PreparedStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.ResultSet;

public class ConsultasProducto extends Conexion{
    public boolean Guardar(Producto pro){
        PreparedStatement ps = null;
        Connection con = getConexion();
        String sql = "INSERT INTO PRODUCTO (codigo, nombre, marca, modelo, precio, cantidad) VALUES (?, ?, ?, ?, ?, ?)";

        try {
            ps = con.prepareStatement(sql);
            ps.setString(1, pro.getCodigo());
            ps.setString(2, pro.getNombre());
            ps.setString(3, pro.getMarca());
            ps.setString(4, pro.getModelo());
            ps.setDouble(5, pro.getPrecio());
            ps.setInt(6, pro.getCantidad());
            ps.execute();
            return true;
        } catch (Exception e) {
            System.err.println(e);
            return false;
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                System.err.println(e);
            }
        }
    }
}
```

```
public boolean Modificar(Producto pro){
    PreparedStatement ps = null;
    Connection con = getConexion();
    String sql = "UPDATE producto SET codigo = ?, nombre = ?, marca = ?, modelo = ?, precio = ?, cantidad = ? WHERE id = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setString(1, pro.getCodigo());
        ps.setString(2, pro.getNombre());
        ps.setString(3, pro.getMarca());
        ps.setString(4, pro.getModelo());
        ps.setDouble(5, pro.getPrecio());
        ps.setInt(6, pro.getCantidad());
        ps.setInt(7, pro.getId());
        ps.execute();
        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
```

```
public boolean Eliminar(Producto pro){
    PreparedStatement ps = null;
    Connection con = getConexion();
    String sql = "DELETE FROM producto WHERE id = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setInt(1, pro.getId());
        ps.execute();
        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
```

```

public boolean Buscar(Producto pro){
    PreparedStatement ps = null;
    Connection con = getConexion();
    ResultSet rs = null;
    String sql = "SELECT * FROM producto WHERE codigo = ?";

    try {
        ps = con.prepareStatement(sql);
        ps.setString(1, pro.getCodigo());
        rs = ps.executeQuery();

        if (rs.next()){
            pro.setId(Integer.parseInt(rs.getString("id")));
            pro.setCodigo(rs.getString("codigo"));
            pro.setNombre(rs.getString("nombre"));
            pro.setMarca(rs.getString("marca"));
            pro.setModelo(rs.getString("modelo"));
            pro.setPrecio(Double.parseDouble(rs.getString("precio")));
            pro.setCantidad(Integer.parseInt(rs.getString("cantidad")));
        } else {
        }

        return true;
    } catch (Exception e) {
        System.err.println(e);
        return false;
    } finally {
        try {
            con.close();
        } catch (Exception e) {
            System.err.println(e);
        }
    }
}
}

```


3. ControlProducto.java

```
package Controlador;
import Modelo.Producto;
import Modelo.ConsultasProducto;
import Vista.frmProducto;
import Vista.Menu;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;
public class ControlProducto implements ActionListener{
    //Vamos a crear una variable para cada PAQUETE
    private final Producto modelo;
    private final ConsultasProducto consultas;
    private final frmProducto vista;
    private final Menu vistaMenu;

    public ControlProducto(Producto modelo, ConsultasProducto consultas, frmProducto vista, Menu vistaMenu){
        this.modelo = modelo;
        this.consultas = consultas;
        this.vista = vista;
        this.vistaMenu = vistaMenu;

        // Realizamos la conexión con el FORMULARIO
        this.vista.btnGuardar.addActionListener(this);
        this.vista.btnModificar.addActionListener(this);
        this.vista.btnEliminar.addActionListener(this);
        this.vista.btnBuscar.addActionListener(this);
        this.vista.btnLimpiar.addActionListener(this);
        this.vista.btnMenu.addActionListener(this);
    }

    public void iniciar(){
        vista.setTitle("Productos");
        vista.setLocationRelativeTo(null);
        vista.setVisible(true);
    }
}
```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == vista.btnGuardar) {
        modelo.setCodigo(vista.txtCodigo.getText());
        modelo.setNombre(vista.txtNombre.getText());
        modelo.setMarca(vista.txtModelo.getText());
        modelo.setModelo(vista.txtModelo.getText());
        modelo.setPrecio(Double.parseDouble(vista.txtPrecio.getText()));
        modelo.setCantidad(Integer.parseInt(vista.txtCantidad.getText()));
        if (consultas.Guardar(modelo)) {
            JOptionPane.showMessageDialog(null, "Registro guardado exitosamente");
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se guardó los datos");
        }
    }

    if (e.getSource() == vista.btnModificar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        modelo.setCodigo(vista.txtCodigo.getText());
        modelo.setNombre(vista.txtNombre.getText());
        modelo.setMarca(vista.txtModelo.getText());
        modelo.setModelo(vista.txtModelo.getText());
        modelo.setPrecio(Double.parseDouble(vista.txtPrecio.getText()));
        modelo.setCantidad(Integer.parseInt(vista.txtCantidad.getText()));
        if (consultas.Modificar(modelo)) {
            JOptionPane.showMessageDialog(null, "Registro actualizado exitosamente");
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se actualizó los datos");
        }
    }

    if (e.getSource() == vista.btnEliminar) {
        modelo.setId(Integer.parseInt(vista.txtId.getText()));
        if (consultas.Eliminar(modelo)) {
            JOptionPane.showMessageDialog(null, "Registro eliminado exitosamente");
        } else {
            JOptionPane.showMessageDialog(null, "¡Error! No se eliminó los datos");
        }
    }
}

```

```

if (e.getSource() == vista.btnBuscar) {
    modelo.setCodigo(vista.txtCodigo.getText());

    if(consultas.Buscar(modelo)){
        vista.txtCodigo.setText(modelo.getCodigo());
        vista.txtNombre.setText(modelo.getNombre());
        vista.txtMarca.setText(modelo.getMarca());
        vista.txtModelo.setText(modelo.getModelo());
        vista.txtPrecio.setText(String.valueOf(modelo.getPrecio()));
        vista.txtCantidad.setText(String.valueOf(modelo.getCantidad()));
    } else {
        JOptionPane.showMessageDialog(null, "No se encontró datos");
    }
}

if(e.getSource() == vista.btnLimpiar){
    Limpiar();
}

if(e.getSource() == vista.btnMenu){
    vista.setVisible(false);
    vistaMenu.setVisible(true);
}
}

public void Limpiar(){
    vista.txtId.setText(null);
    vista.txtCodigo.setText(null);
    vista.txtNombre.setText(null);
    vista.txtMarca.setText(null);
    vista.txtModelo.setText(null);
    vista.txtPrecio.setText(null);
    vista.txtCantidad.setText(null);
}
}

```

4. frmProducto.java

Vista\frmProducto.java ×

Source Design History

MANTENIMIENTO DE PRODUCTOS

ID

CÓDIGO


NOMBRE


MARCA

MODELO


PRECIO


CANTIDAD


 BUSCAR

 LIMPIAR

MENU

 GUARDAR

 MODIFICAR

 ELIMINAR

JAVA – CONEXIÓN (ALL)

```
package Modelo;
import java.sql.Connection;
import java.sql.DriverManager;
public class Conexion {
    Connection con = null;
    // Creamos las variables de conexión
    String bd = "GESTION_VENTAS";
    String url = "jdbc:mysql://localhost:3306/" + bd;
    String user = "root";
    String password = "";

    public Connection getConexion() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, user, password);
        } catch (Exception e) {
            System.err.println(e);
        }
        return con;
    }
}
```

JAVA – MENU (ALL)

1. ControlMenu

```
package Controlador;

import Modelo.ConsultasProveedores;
import Modelo.Proveedores;
import Vista.frmProveedores;
import Modelo.ConsultasClientes;
import Modelo.Clientes;
import Vista.frmClientes;
import Modelo.ConsultasProducto;
import Modelo.Producto;
import Vista.frmProducto;
import Vista.Menu;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ControlMenu implements ActionListener{
    private final Menu vistaMenu;
    private frmClientes vistaClientes;
    private frmProveedores vistaProveedores;
    private frmProducto vistaProducto;

    public ControlMenu(Menu vistaMenu){
        this.vistaMenu = vistaMenu;

        // Conectar botones del formulario
        this.vistaMenu.btnClientes.addActionListener(this);
        this.vistaMenu.btnProveedores.addActionListener(this);
        this.vistaMenu.btnProducto.addActionListener(this);
        this.vistaMenu.btnSalir.addActionListener(this);
    }

    public void iniciar(){
        vistaMenu.setTitle("Gestion de Ventas");
        vistaMenu.setVisible(true);
        vistaMenu.setLocationRelativeTo(null);
    }
}
```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == vistaMenu.btnClientes) {
        abrirClientes();
    }
    if (e.getSource() == vistaMenu.btnProducto) {
        abrirProducto();
    }
    if (e.getSource() == vistaMenu.btnProveedores) {
        abrirProveedores();
    }
    if (e.getSource() == vistaMenu.btnSalir) {
        System.exit(0);
    }
}

public void abrirClientes(){
    if (vistaClientes == null) {
        Clientes modelo = new Clientes();
        ConsultasClientes consulta = new ConsultasClientes();
        frmClientes vistaClientes = new frmClientes();

        ControlClientes ctrl = new ControlClientes(modelo, consulta, vistaClientes, vistaMenu);
        ctrl.iniciar();
    }
    vistaMenu.setVisible(false);
    vistaClientes.setVisible(true);
}

public void abrirProducto(){
    if (vistaProducto == null) {
        Producto modelo = new Producto();
        ConsultasProducto consulta = new ConsultasProducto();
        frmProducto vistaProducto = new frmProducto();

        ControlProducto ctrl = new ControlProducto(modelo, consulta, vistaProducto, vistaMenu);
        ctrl.iniciar();
    }
    vistaMenu.setVisible(false);
    vistaProducto.setVisible(true);
}
}

```

```

public void abrirProveedores(){
    if (vistaProducto == null) {
        Proveedores modelo = new Proveedores();
        ConsultasProveedores consulta = new ConsultasProveedores();
        frmProveedores vistaProveedores = new frmProveedores();

        ControlProveedores ctrl = new ControlProveedores(modelo, consulta, vistaProveedores, vistaMenu);
        ctrl.iniciar();
    }
    vistaMenu.setVisible(false);
    vistaProveedores.setVisible(true);
}
}

```

2. Menu



JAVA – EJECUCIÓN

CLIENTES



The screenshot shows a Java Swing window titled "Clientes". The window has a standard title bar with minimize, maximize, and close buttons. The main content area displays the text "FORMULARIO DE CLIENTES" in a bold, black font. Below the title, there is a section labeled "DATOS DEL CLIENTE:". To the right of this label is a "MENU" button. Below the label, there are four input fields: "ID", "DESCRIPCIÓN", "DIRECCIÓN", and "TELÉFONO". The "ID" field is a single-line text box, while the others are wider. Below the input fields, there are six buttons with icons and labels: "NUEVO" (document with plus), "GUARDAR" (floppy disk with checkmark), "EDITAR" (document with pencil), "ELIMINAR" (trash can), "BUSCAR" (magnifying glass), and "CANCELAR" (red circle with slash). Below these buttons is a table with five columns: "ID", "Desripción", "Dirección", "Teléfono", and "RUC". The table is currently empty. At the bottom of the window, there is a line of red text: "AbstractButton\$Handler.actionPerformed(AbstractButton.java:2314)".

ID	Desripción	Dirección	Teléfono	RUC
----	------------	-----------	----------	-----

FORMULARIO DE CLIENTES

DATOS DEL CLIENTE:

MENU

ID DESCRIPCIÓN TELÉFONO DIRECCIÓN RUC

Message



Registro guardado exitosamente

OK



NUEVO



GUARDAR

EDITAR

ELIMINAR



BUSCAR



CANCELAR

ID	Descripción	Dirección	Teléfono	RUC

FORMULARIO DE CLIENTES

DATOS DEL CLIENTE:

MENU

ID DESCRIPCIÓN TELÉFONO DIRECCIÓN RUC 

NUEVO



GUARDAR



EDITAR



ELIMINAR



BUSCAR



CANCELAR

ID	Descripción	Dirección	Teléfono	RUC
1	*****	Av. Peru, 2024	952652352	1254124

Cientes

FORMULARIO DE CLIENTES

DATOS DEL CLIENTE:

MENU

ID1

DESCRIPCIÓN

TELÉFONO

DIRECCIÓN

RUC

NUEVO

GUARDAR

EDITAR

ELIMINAR

BUSCAR

CANCELAR

Message

Registro eliminado exitosamente

OK

ID	Desripición	Dirección	Teléfono	RUC
1	*****	Av. Peru, 2024	952652352	1254124

AbstractButton\$Handler.actionPerformed(AbstractButton.java:2314)

FORMULARIO DE CLIENTES

DATOS DEL CLIENTE:

MENU

ID 1

DESCRIPCIÓN *****

TELÉFONO 952652352

DIRECCIÓN Av. Peru, 2011

RUC 1254124

Message



Registro actualizado exitosamente

OK



NUEVO



GUARDAR

EDITAR

ELIMINAR



BUSCAR



CANCELAR

ID	Desripción	Dirección	Teléfono	RUC
2	*****	Av. Peru, 2024	952652352	1254124

-0"
abr
act
.Ab
.AbstractButton\$Handler.actionPerformed([AbstractButton.java:2314](#))

staClie)



Productos



MANTENIMIENTO DE PRODUCTOS

ID

CÓDIGO

NOMBRE

MARCA

MODELO

PRECIO

CANTIDAD



BUSCAR



LIMPIAR

MENU



GUARDAR



MODIFICAR



ELIMINAR

MANTENIMIENTO DE PRODUCTOS



Productos



MANTENIMIENTO DE PRODUCTOS

ID

1



BUSCAR

CÓDIGO

1234

Message



LIMPIAR

NOMBRE

Libro

MENU

MARCA

A12

MODELO

A12

PRECIO

20.0

CANTIDAD

2



Registro actualizado exitosamente

OK



GUARDAR



MODIFICAR



ELIMINAR

PROVEEDORES

Proveedores

FORMULARIO DE PROVEEDORES

DATOS DEL PROVEEDOR:

MENU

ID

DESCRIPCIÓN

DIRECCIÓN

Jr. Joaquin capello, 2345

TELÉFONO

952623545

NUEVO

GUARDAR

EDITAR

ELIMINAR

BUSCAR

CANCELAR

Message

Registro guardado exitosamente

OK

ID	Desripción	Dirección	Teléfono
----	------------	-----------	----------

FORMULARIO DE PROVEEDORES

DATOS DEL PROVEEDOR:

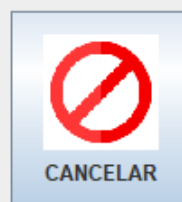
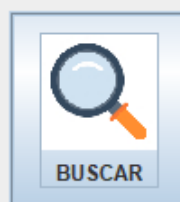
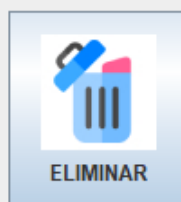
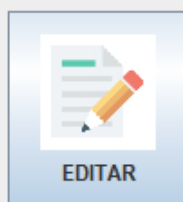
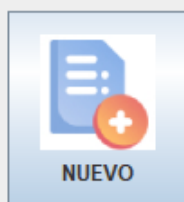
MENU

ID 1

DESCRIPCIÓN *****

DIRECCIÓN Jr. Joaquin capello, 2345

TELÉFONO 952623545



ID	Desripción	Dirección	Teléfono
1	*****	Jr. Joaquin capello, ...	952623545

Proveedores

FORMULARIO DE PROVEEDORES

DATOS DEL PROVEEDOR:

MENU

ID

3

DESCRIPCIÓN

sjr

DIRECCIÓN

jtrjjh,jf,

TELÉFONO

srjjtr

NUEVO

GUARDAR

EDITAR

ELIMINAR

BUSCAR

CANCELAR

Message

Registro actualizado exitosamente

OK

ID	Desripción	Dirección	Teléfono
3	sjr	jtrj	srjjtr

ctionPerformed(ControlMenu.java:46)

FORMULARIO DE PROVEEDORES

DATOS DEL PROVEEDOR:

MENU

ID 3

DESCRIPCIÓN

DIRECCIÓN

TELÉFONO

Message



Registro eliminado exitosamente

OK



NUEVO



GUARDAR

EDITAR

ELIMINAR



BUSCAR



CANCELAR

ID	Desripción	Dirección	Teléfono
3	sjr	jtrj	srjitr

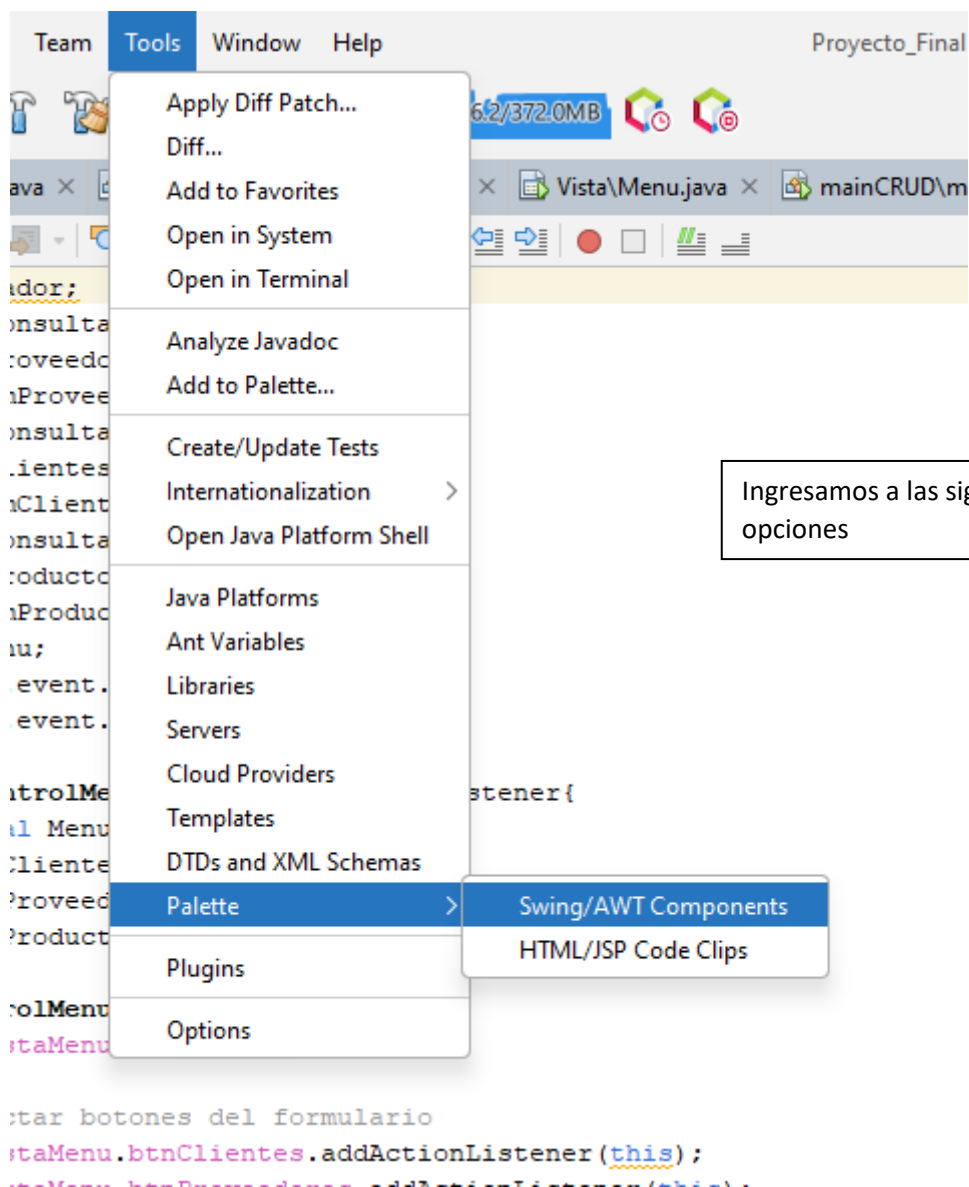
JAVA – JCalender

Download and Latest Version

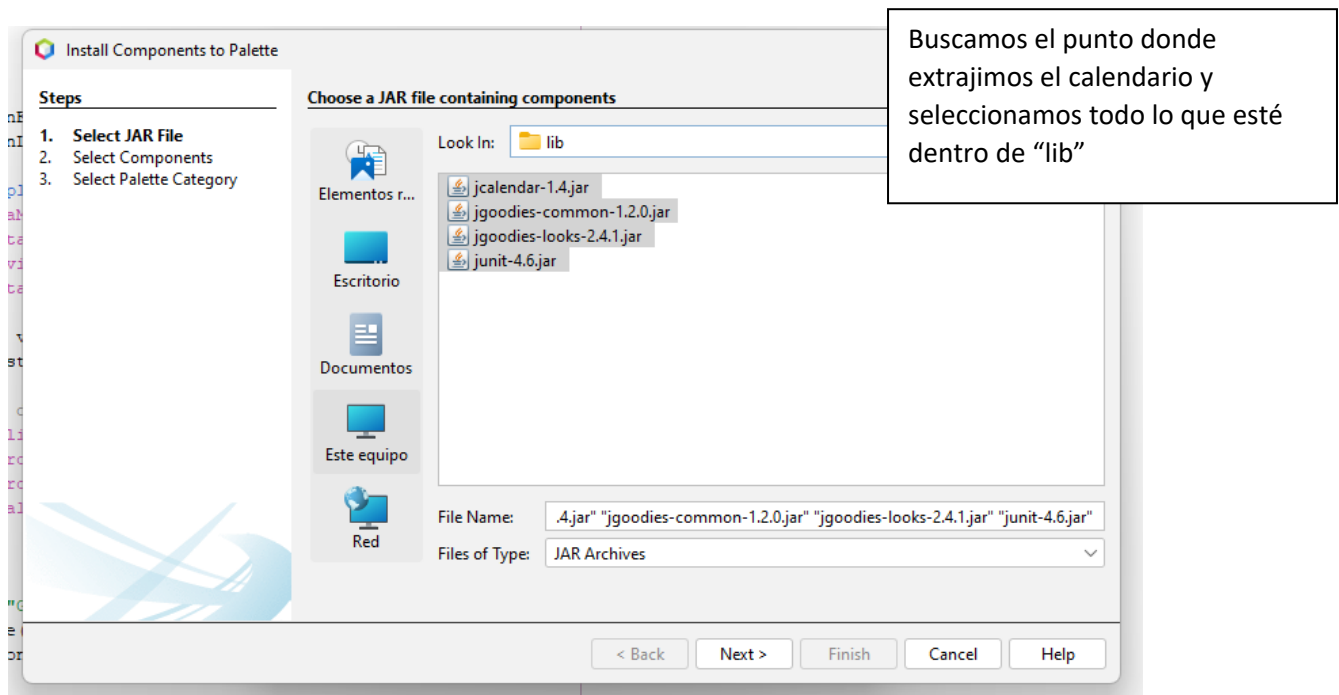
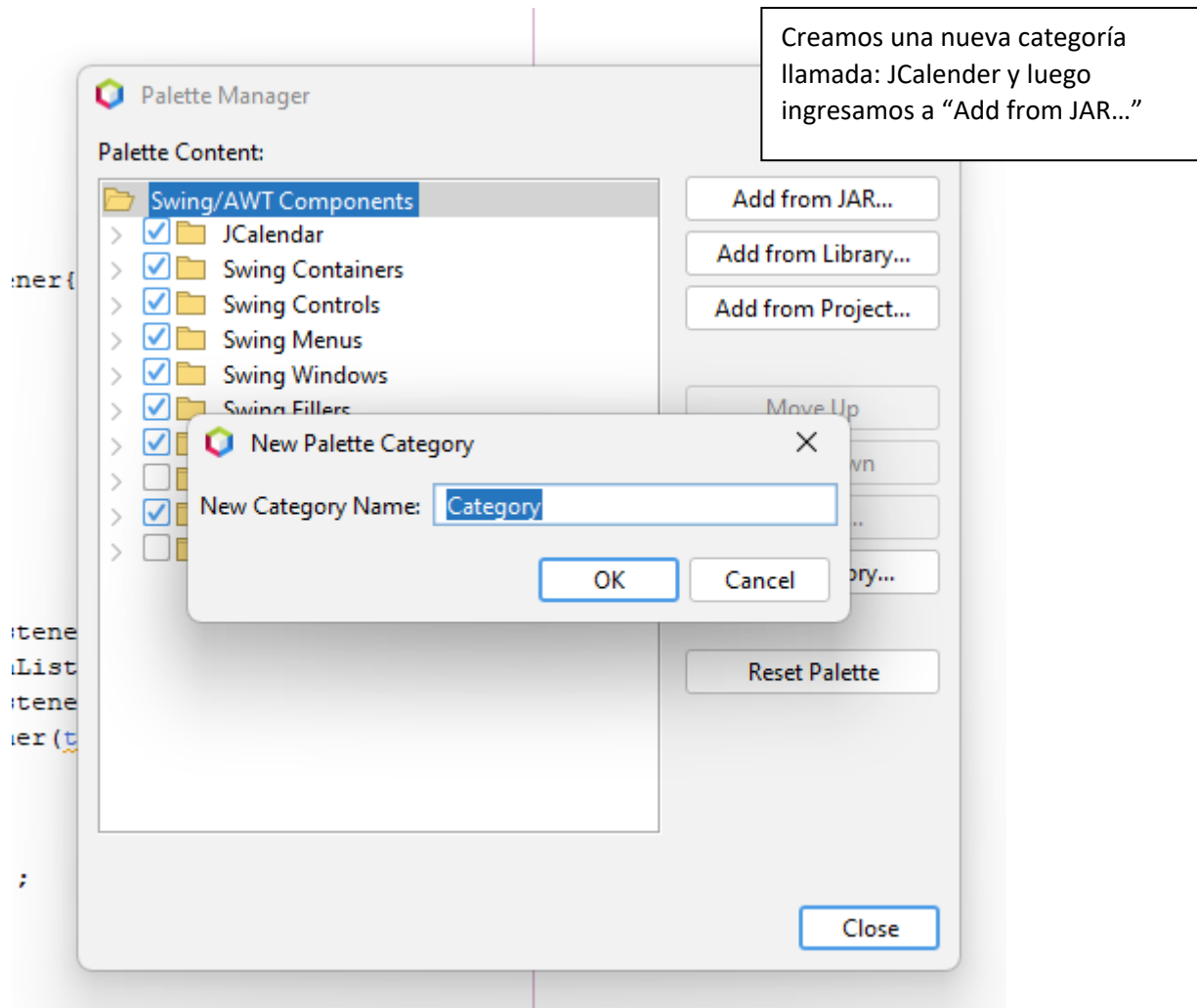
You can download a zip file, containing all the sources, libs and documentation.

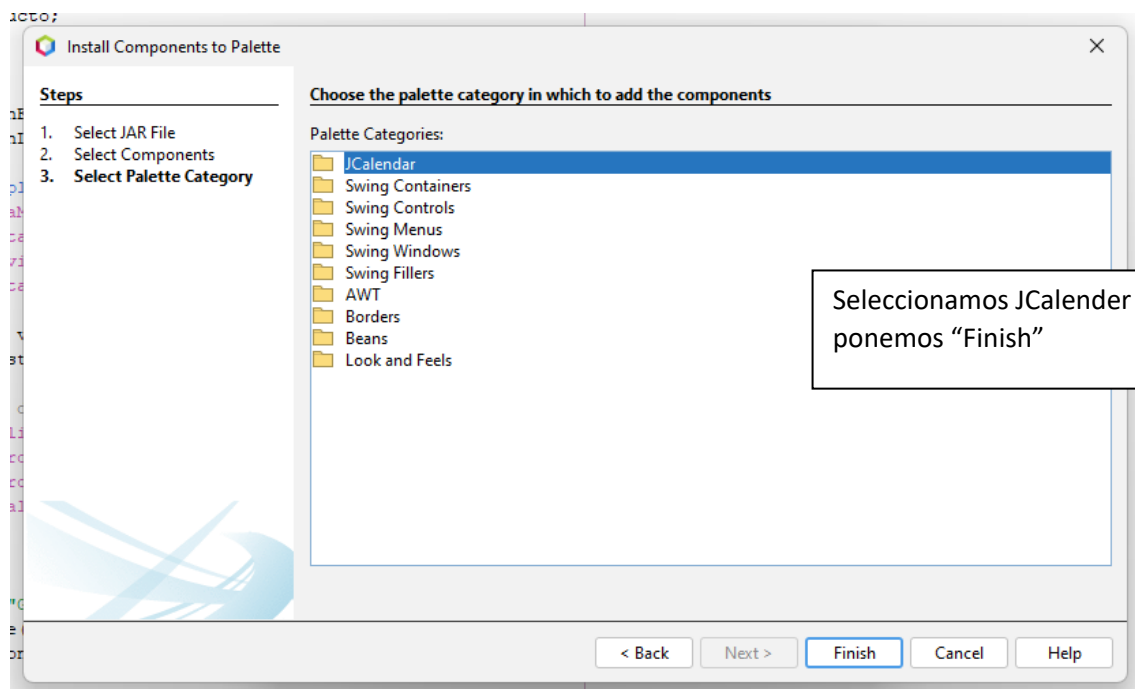
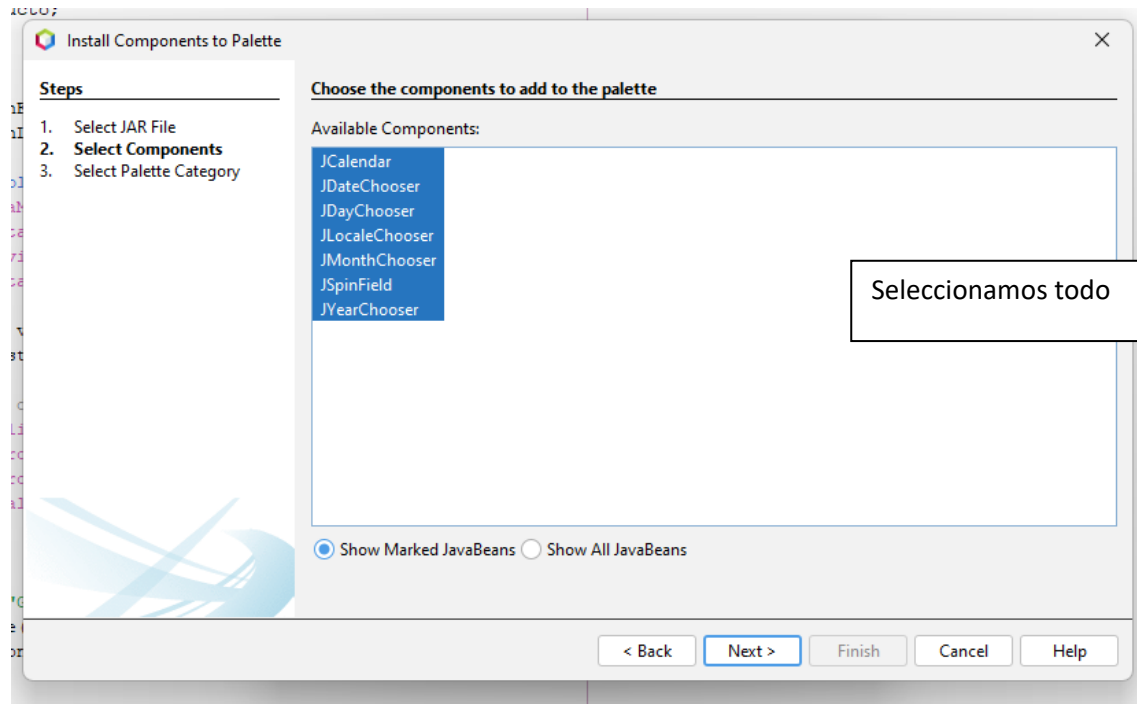
- Latest release version 1.4: [jcalendar-1.4.zip](#)


Descargamos la librería en “JCalender”, luego debemos extraerla.



Ingresamos a las siguientes opciones







```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to view the code.
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to view the code.
4  */
5  package Vista;
6  import java.util.Date;
7  /**
8   *
9   * @author Admin-ETI
10  */
11  public class Menu extends javax.swing.JFrame {
12
13      /**
14       * Creates new form Menu
15       */
16      public Menu() {
17          initComponents();
18          jDateChooser.addPropertyChangeListener("date", evt ->{
19              Date selectedDate = jDateChooser.getDate();
20          });
21      }
22  }
```

Ingresamos al formulario donde
deseamos agregar el calendario y
agregamos estas dos líneas de
código en "Source".