



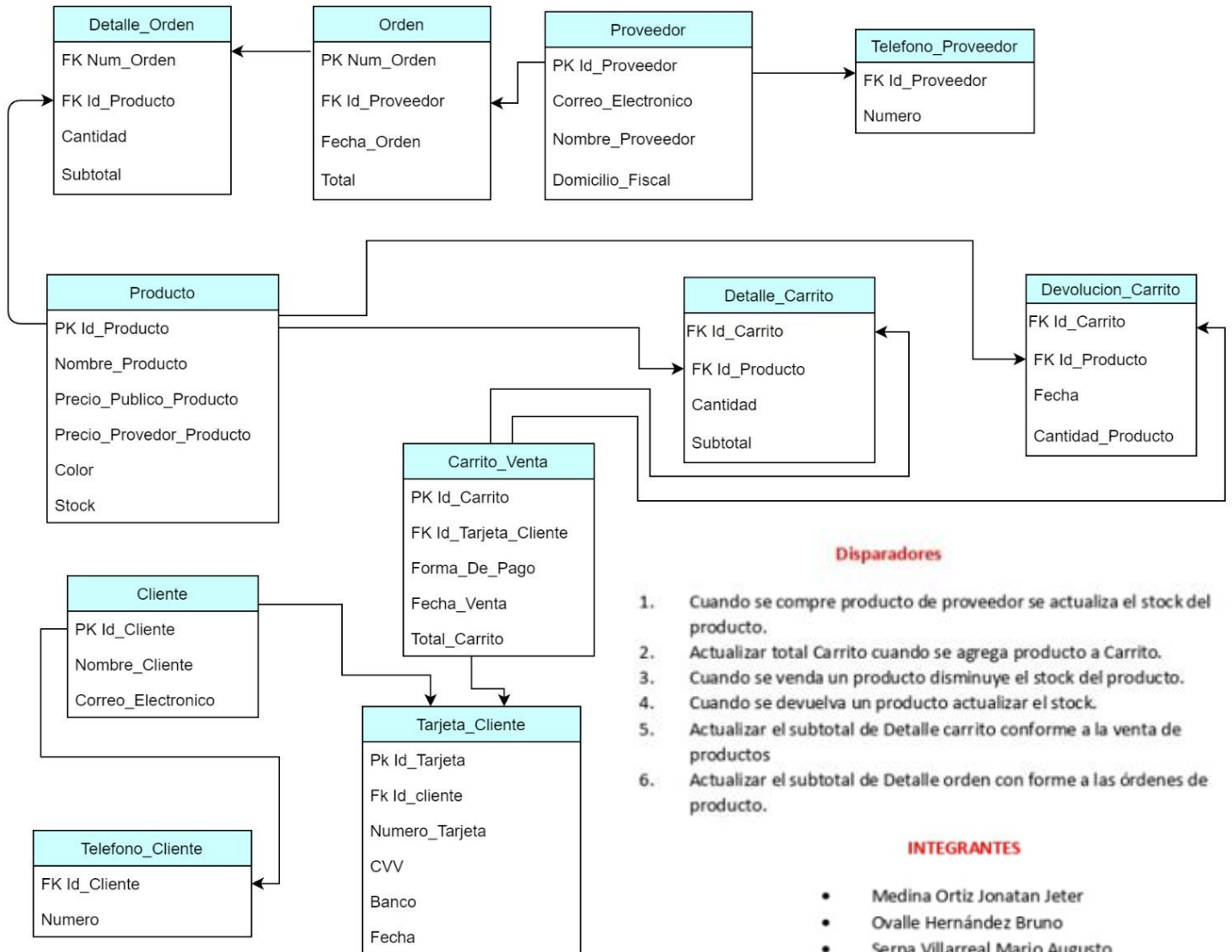
Medina Ortiz Jonatan Jeter  
Serna Villarreal Mario Augusto  
Ovalle Hernández Bruno

Manual del Programador

231301

2022-2023/II

# COMPRAS EN LINEA



## Clase Producto

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Artículo;

import Datos.Tarjeta_Cliente;

import Datos.Telefono_Cliente;

import Menu.Menu;

import java.sql.Connection;

import java.awt.Point;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

/*
 vista en la cual puedes ver, modificar, agregar o eliminar productos de la base de datos
 */

public class Producto extends javax.swing.JFrame {

    private String HOST="localhost"; //host

    private String PUERTO="5432"; //puerto

    private String DB="ComprasEnLinea"; //base de datos

    private String USER="postgres"; //usuario

    private String PASS="postgres"; //contraseña

    public String url="jdbc:postgresql://" + HOST + ":" + PUERTO + "/" + DB; // direccion url

    private Connection conexion=null; // conexion con la base de datos

    public String id; // id para cada producto
```

```

/**
 * Creates new form Producto
 */
public Producto(String u) {
    initComponents();
    muestra();
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    usuario(u);
}

public void usuario(String u){
    if(u.equals("pablo")){
        Inserta.setEnabled(false);
        Modifica.setEnabled(false);
        Elimina.setEnabled(false);
    }
}

// método para conectar con la base de datos
public Connection conectaBD(){
    Connection cone=null;

    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }

    return cone;
}

// inserta un producto en la base de datos
public void inserta(){
    //verifica si todos los campos del producto están completos
    if(Nombre.getText().equalsIgnoreCase("") || PrecioPublico.getText().equalsIgnoreCase("")
        || PrecioProveedor.getText().equalsIgnoreCase("") || Color.getText().equalsIgnoreCase("")
        || Stock.getText().equalsIgnoreCase("")){
        JOptionPane.showMessageDialog(this, "Casillas vacias");
    }else{
        try{

```

```

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // inserta un nuevo producto en la tabla Artículo.Producto

        String Query="INSERT INTO
Artículo.Producto(Nombre_Producto,Precio_Publico_Producto,Precio_Publico_Proveedor,Color,Stock)"
        + "VALUES('" +Nombre.getText()+"',CAST('"+PrecioPublico.getText()+"' AS DECIMAL(10,2))
,CAST('"+PrecioProveedor.getText()+"' AS DECIMAL(10,2)), '"+Color.getText()+"', '"+Stock.getText()+"'";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close();// cierra la conexión con la base ded atos

        //JOptionPane.showMessageDialog(null, "INSERTO");

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//muestra todos los productos en la base de datos

public void muestra(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener los productos de la base de datos, ordenados de manera ascendente según su Id

        String Query="Select id_producto,nombre_producto,precio_publico_producto\n" +

        ",precio_publico_proveedor\n" +

        ",color,stock FROM Artículo.Producto\n" +

        "ORDER BY Id_Producto ASC";

        String[]datos =new String[6];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        DefaultTableModel model=new DefaultTableModel();

        //crea las columnas en el jTable

        model.addColumn("Id Producto");

        model.addColumn("Nombre Producto");

        model.addColumn("Precio Público Producto");

        model.addColumn("Precio Proveedor Producto");

        model.addColumn("Color");

        model.addColumn("Stock");

```

```

Productos.setModel(model);

//llena las columnas del jTable
while(columnas.next()){

    datos[0]=columnas.getString(1);

    datos[1]=columnas.getString(2);

    datos[2]=columnas.getString(3);//publico

    datos[3]=columnas.getString(4);//proveedor

    datos[4]=columnas.getString(5);

    datos[5]=columnas.getString(6);

    model.addRow(datos);

}

// obtiene la informacion del producto al hacer clic en el renglón del jTable
Productos.addMouseListener(new MouseAdapter() {

    public void mousePressed(MouseEvent Mouse_evt){

        JTable table=(JTable) Mouse_evt.getSource();

        Point point = Mouse_evt.getPoint();

        int row=table.rowAtPoint(point);

        // llena los jTextField con los datos del producto

        if(Mouse_evt.getClickCount()==1){

            Nombre.setText(Productos.getValueAt(Productos.getSelectedRow(),1).toString());

            PrecioPublico.setText(Productos.getValueAt(Productos.getSelectedRow(),2).toString());

            PrecioProveedor.setText(Productos.getValueAt(Productos.getSelectedRow(),3).toString());

            Color.setText(Productos.getValueAt(Productos.getSelectedRow(),4).toString());

            Stock.setText(Productos.getValueAt(Productos.getSelectedRow(),5).toString());

            id=Productos.getValueAt(Productos.getSelectedRow(), 0).toString();

        }

    }

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

// método para eliminar un producto

```

```

public void elimina(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para eliminar un producto de la tabla Artículo.Producto

        String Query="DELETE FROM Artículo.Producto WHERE Id_Producto="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexión con la base de datos

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//actualiza los datos del producto

public void modifica(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para actualizar los datos del producto

        String Query="UPDATE Artículo.Producto SET
Nombre_Producto='"+Nombre.getText()+"',Precio_Publico_Producto='"+PrecioPublico.getText()+"',Precio_Publico_Proveedor='"+
PrecioProveedor.getText()+"', "

        + "Color='"+Color.getText()+"',Stock='"+Stock.getText()+" WHERE Id_Producto="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexión con la base de datos

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@Override
@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

```

```
jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

Nombre = new javax.swing.JTextField();

PrecioPublico = new javax.swing.JTextField();

PrecioProveedor = new javax.swing.JTextField();

Color = new javax.swing.JTextField();

Stock = new javax.swing.JTextField();

Inserta = new javax.swing.JButton();

Modifica = new javax.swing.JButton();

Elimina = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

Productos = new javax.swing.JTable();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Producto");


jLabel1.setFont(new java.awt.Font("Tahoma", 2, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 204, 204));
jLabel1.setText("Producto");


jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel2.setText("Nombre");


jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel3.setText("Precio Público");


jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel4.setText("Precio Proveedor");


jLabel5.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel5.setText("Color");


jLabel6.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```



```
jLabel6.setText("Stock");
```

```
Nombre.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
Nombre.setToolTipText("");
```

```
PrecioPublico.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
PrecioProveedor.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
Color.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
Stock.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
Inserta.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
Inserta.setText("INSERTA");
```

```
Inserta.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        InsertaActionPerformed(evt);  
    }  
});
```

```
Modifica.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
Modifica.setText("MODIFICA");
```

```
Modifica.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ModificaActionPerformed(evt);  
    }  
});
```

```
Elimina.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
Elimina.setText("ELIMINA");
```

```
Elimina.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EliminaActionPerformed(evt);  
    }  
});
```

```
Productos.setModel(new javax.swing.table.DefaultTableModel(  

```

```

new Object [][] {

},

new String [] {

    "Title 1", "Title 2", "Title 3", "Title 4"

}

});

jScrollPane1.setViewportViewView(Productos);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 990, Short.MAX_VALUE)
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel2, javax.swing.GroupLayout.Alignment.LEADING)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.LEADING)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel5, javax.swing.GroupLayout.Alignment.LEADING)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel6, javax.swing.GroupLayout.Alignment.LEADING)
                    .addGap(27, 27, 27)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(Nombre, javax.swing.GroupLayout.PREFERRED_SIZE, 325, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(ContainerGap, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(Stock, javax.swing.GroupLayout.DEFAULT_SIZE, 95, Short.MAX_VALUE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(Color, javax.swing.GroupLayout.Alignment.LEADING)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(PrecioProveedor)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(PrecioPublico))
                    )
                )
            )
        )
    )

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(Inserta, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 117, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(Elimina, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 107, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(75, 75, 75)

        .addComponent(Modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(67, 67, 67))))))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

            .addContainerGap()

            .addComponent(jLabel1)

            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel2)

                .addComponent(Nombre, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup())

                    .addGap(111, 111, 111)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                        .addComponent(jLabel6)

                        .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

                .addGroup(layout.createSequentialGroup())

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(layout.createSequentialGroup())

                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                .addComponent(Modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)

                                .addComponent(Inserta, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE))

                            .addGap(30, 30, 30)

                            .addComponent(Elimina, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))

                        .addGroup(layout.createSequentialGroup())

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent(PrecioPublico, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

            .addComponent(jLabel4)

            .addComponent(PrecioProveedor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(Color, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 346,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap()

    );

    pack();
} // </editor-fold>

```

```

private void InsertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();

    Nombre.setText("");

    PrecioPublico.setText("");

    PrecioProveedor.setText("");

    Color.setText("");

    Stock.setText("");

    muestra();

}

```

```

private void EliminaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    elimina();

    Nombre.setText("");

    PrecioPublico.setText("");

```

```
PrecioProveedor.setText("");

Color.setText("");

Stock.setText("");

muestra();
}
```

```
private void ModificaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    modifica();

    Nombre.setText("");

    PrecioPublico.setText("");

    PrecioProveedor.setText("");

    Color.setText("");

    Stock.setText("");

    muestra();
}
```

```
// Variables declaration - do not modify

private javax.swing.JTextField Color;

private javax.swing.JButton Elimina;

private javax.swing.JButton Inserta;

private javax.swing.JButton Modifica;

private javax.swing.JTextField Nombre;

private javax.swing.JTextField PrecioProveedor;

private javax.swing.JTextField PrecioPublico;

private javax.swing.JTable Productos;

private javax.swing.JTextField Stock;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JScrollPane jScrollPane1;

// End of variables declaration
}
```

## Clase Tarjeta\_Cliente

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Datos;
```

```
  
import Usuario.Cliente;  
  
import java.sql.Connection;  
  
import java.awt.Point;  
  
import java.awt.event.MouseAdapter;  
  
import java.awt.event.MouseEvent;  
  
import java.sql.Connection;  
  
import java.sql.DriverManager;  
  
import java.sql.ResultSet;  
  
import java.sql.SQLException;  
  
import java.util.Calendar;  
  
import java.util.Date;  
  
import java.util.LinkedList;  
  
import java.util.Locale;  
  
import java.util.logging.Level;  
  
import java.util.logging.Logger;  
  
import javax.swing.ComboBoxModel;  
  
import javax.swing.JComboBox;  
  
import javax.swing.JFrame;  
  
import javax.swing.JOptionPane;  
  
import javax.swing.JTable;  
  
import javax.swing.JTextField;  
  
import javax.swing.table.DefaultTableModel;
```

```
/*  
  
vista donde puedes ver, agregar, modificar o eliminar las tarjetas del cliente  
  
*/
```

```

public class Tarjeta_Cliente extends javax.swing.JFrame {

    /*
    conjunto de variables necesarias para realizar la conexión a la
    base de datos
    */

    private String HOST="localhost"; //host
    private String PUERTO="5432"; // puerto
    private String DB="ComprasEnLinea"; // base de datos
    private String USER="postgres"; // usuario
    private String PASS="postgres"; // contraseña

    public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //direccion creada al juntar el host, puerto y base de datos
    private Connection conexion=null; // variable para la conexión

    public String id; //Variable para id de Cliente
    public String numeros; //Variable para guardar los numeros de la tarjeta
    public LinkedList<String> ids=new LinkedList<>();
    public LinkedList<String> fechas=new LinkedList<>();

    /**
     * Creates new form Tarjeta_Cliente
     */
    public Tarjeta_Cliente(String U) {
        initComponents();
        clientes();
        muestra();
        fecha();
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        usuario(U);
    }

    public void usuario(String u){
        if(u.equals("pablo")){
            inserta.setEnabled(false);
            modifica.setEnabled(false);
            elimina.setEnabled(false);
        }
    }

    // método para desplegar el calendario

```

```

public void fecha(){

    Calendar fechaActual = Calendar.getInstance();

    fechaActual.add(Calendar.MONTH, 5);

    Date.setSelectableDateRange(fechaActual.getTime(), null);

}

// metodo que obtiene los clientes de la base de datos ordenados de forma ascendente según el Id

public void clientes(){

    try{

        conexion=connectaBD(); // realiza la conexión a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener los datos del cliente, ordenados de forma ascendente según su Id

        String Query="SELECT Nombre_Cliente,Id_Cliente,Correo_Electronico FROM Usuario.Cliente ORDER BY Id_Cliente ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query);

        while(columnas.next()){

            // inserta el cliente

            Cliente.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            ids.addLast(columnas.getString(2));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// conecta a la base de datos a travez del driver, url, usuario y password

public Connection connectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

// inserta una tarjeta en la base de datos segun el Id del cliente seleccionado

public void inserta(){

    try{

```



```

conexion=connectaBD(); //crea la conexión a la base de datos

java.sql.Statement corrida=conexion.createStatement();

int id=Cliente.getSelectedIndex(); // selecciona el id del cliente

// query que realiza la inserción de una nueva tarjeta en la tabla Datos.tarjeta_cliente

String Query="INSERT INTO Datos.Tarjeta_Cliente(Id_Cliente,Numero_Tarjeta,CVV,Banco,Fecha)"

+
"VALUES('"+ids.get(Cliente.getSelectedIndex())+"','"+Numero.getText()+"','"+cvv.getText()+"','"+Banco.getText()+"','"+((JTextField)
Date.getDateEditor().getUiComponent()).getText()+"')";

corrida.executeUpdate(Query); // ejecuta el query

corrida.close();

conexion.close(); //cierra la conexion

//JOptionPane.showMessageDialog(null, "INSERTO");

}catch(Exception ex){

JOptionPane.showMessageDialog(null,"Numero De Tarjeta Duplicado");

}

}

//muestra en el jtable las tarjetas que estén relacionadas según el Id del cliente seleccionado

public void muestra(){

try{

conexion=connectaBD(); // crea la conexión con la base de datos

java.sql.Statement corrida=conexion.createStatement();

//query que muestra en el jtable las tarjetas del cliente, relacionando las tablas Datos.Tarjeta_cliente y Usuario.Cliente,
ordenadas de forma ascendente

String Query="SELECT
T.Id_Cliente,T.Numero_Tarjeta,T.CVV,T.Banco,to_char(Fecha,'MM/yyyy'),C.Nombre_Cliente,C.Correo_Electronico,Fecha FROM
Datos.Tarjeta_Cliente T,Usuario.Cliente C WHERE T.Id_Cliente=C.Id_Cliente ORDER BY T.Id_Cliente ASC";

String[]datos =new String[8];

corrida=conexion.createStatement();

ResultSet columnas=corrida.executeQuery(Query);

DefaultTableModel model=new DefaultTableModel();

//se agregan las columnas al jtable

model.addColumn("Id Cliente");

model.addColumn("Número Tarjeta");

model.addColumn("CVV");

model.addColumn("Banco");

model.addColumn("Fecha");

model.addColumn("Nombre Cliente - Correo Electrónico");

Ttarjeta.setModel(model);

//se llenan las columnas del jtable

while(columnas.next()){

```

```

        datos[0]=columnas.getString(1);

        datos[1]=columnas.getString(2);

        datos[2]=columnas.getString(3);

        datos[3]=columnas.getString(4);

        datos[4]=columnas.getString(5);

        datos[5]=columnas.getString(6)+" - "+columnas.getString(7);

        model.addRow(datos);

    }

    //obtiene los datos de la tarjeta al hacer clic en el renglón del jTable

    Ttarjeta.addMouseListener(new MouseAdapter() {

        public void mousePressed(MouseEvent Mouse_evt){

            JTable table=(JTable) Mouse_evt.getSource();

            Point point = Mouse_evt.getPoint();

            int row=table.rowAtPoint(point);

            if(Mouse_evt.getClickCount()==1){

                //se llenan los jTextField con los datos de la tarjeta seleccionada

                Numero.setText(Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(),1).toString());

                cvv.setText(Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(),2).toString());

                Banco.setText(Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(),3).toString());

                for (int i =0; i < ids.size(); i++){

                    if(ids.get(i).equals(Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(),0).toString())){

                        Cliente.setSelectedIndex(i);

                    }

                }

                try{

                    conexion=connectaBD(); //crea la conexión con la base de datos

                    java.sql.Statement corrida;

                    //query para obtener la fecha de la tarjeta convertida en char/cadena de la tabla Datos.Tarjeta_Cliente

                    String Query="SELECT to_char(Fecha,'yyyy/MM/dd') FROM Datos.Tarjeta_Cliente WHERE

Numero_Tarjeta='"+Numero.getText()+"'";

                    corrida=conexion.createStatement();

                    ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

                    while(columnas.next()){

                        ((JTextField)Date.getDateEditor().getUiComponent()).setText(columnas.getString(1));

                    }

                    corrida.close();

                    conexion.close(); //cierra la conexion

```

```

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    // obtiene el id del cliente al momento de seleccionar el renglo del jtable
    id=Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(), 0).toString();

    // obtiene los numeros de la tarjeta al seleccionar el renglón dentro del jtable
    numeros=Ttarjeta.getValueAt(Ttarjeta.getSelectedRow(), 1).toString();

    }

    }

    });

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

//método para elimiar una tupla de la tabla Datos.Tarjeta_cliente
public void elimina(){

    try{

        conexion=connectaBD(); //crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para eliminar la tupla que corresponda al numero de tarjeta de la tabla Datos.Tarjeta_cliente
        String Query="DELETE FROM Datos.Tarjeta_Cliente WHERE Numero_Tarjeta='"+Numero.getText()+"'";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//método para modificar una tupla de la tabla Datos.Tarjeta_Cliente
public void modifica(){

    try{

        conexion=connectaBD(); //crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para actualizar los datos de la tupla relacionada con el Id del cliente y el numero de la tarjeta

        String Query="UPDATE Datos.Tarjeta_Cliente SET
Numero_Tarjeta='"+Numero.getText()+"',Id_Cliente='"+ids.get(Cliente.getSelectedIndex())+"', "

```

```

+
"CVV="+cvv.getText()+""+Banco=""+Banco.getText()+""+Fecha=""+"((JTextField)Date.getDateEditor().getUiComponent()).getText()"+
""

        + "WHERE Id_Cliente="+id+" AND Numero_Tarjeta="+numeros+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "Numero De Tarjeta Duplicado");

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    Numero = new javax.swing.JTextField();
    cvv = new javax.swing.JTextField();
    Banco = new javax.swing.JTextField();
    Cliente = new javax.swing.JComboBox();
    Date = new com.toedter.calendar.JDateChooser();
    jScrollPane1 = new javax.swing.JScrollPane();
    Ttarjeta = new javax.swing.JTable();
    inserta = new javax.swing.JButton();
    elimina = new javax.swing.JButton();
    modifica = new javax.swing.JButton();

```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);  
setTitle("TarjetaCliente");
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 2, 18)); // NOI18N  
jLabel1.setText("Tarjeta Cliente");
```

```
jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel2.setText("Número De Tarjeta");
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel3.setText("CVV");
```

```
jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel4.setText("Banco");
```

```
jLabel5.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel5.setText("Fecha");
```

```
jLabel6.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel6.setText("Cliente");
```

```
Numero.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N  
Numero.setToolTipText("");
```

```
cvv.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
```

```
Banco.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N  
Banco.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        BancoActionPerformed(evt);  
    }  
});
```

```
Cliente.setModel(new javax.swing.DefaultComboBoxModel(new String[] { }));  
Cliente.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ClienteActionPerformed(evt);  
    }  
});
```

```
    }  
});
```

```
Date.setDateFormatString("yyyy/MM/d");
```

```
Ttarjeta.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {  
  
    },  
    new String [] {  
        "Title 1", "Title 2", "Title 3", "Title 4"  
    }  
));  
jScrollPane1.setViewportViewView(Ttarjeta);
```

```
inserta.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
inserta.setText("INSERTA");  
inserta.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        insertaActionPerformed(evt);  
    }  
});
```

```
elimina.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
elimina.setText("ELIMINA");  
elimina.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        eliminaActionPerformed(evt);  
    }  
});
```

```
modifica.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
modifica.setText("MODIFICA");  
modifica.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        modificaActionPerformed(evt);  
    }  
});
```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                .addGroup(layout.createSequentialGroup()

                    .addContainerGap()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                        .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(layout.createSequentialGroup()

                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                .addGroup(layout.createSequentialGroup()

                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                                        .addGroup(layout.createSequentialGroup()

                                            .addComponent(jLabel2)

                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                                            .addComponent(Número, javax.swing.GroupLayout.PREFERRED_SIZE, 204,
javax.swing.GroupLayout.PREFERRED_SIZE))

                                    .addGroup(layout.createSequentialGroup()

                                        .addComponent(jLabel5)

                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                                        .addComponent(Date, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                                        .addGap(153, 153, 153))

                                    .addGroup(layout.createSequentialGroup()

                                        .addComponent(jLabel6)

                                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                                        .addComponent(Cliente, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                                    .addGroup(layout.createSequentialGroup()

                                        .addComponent(jLabel3)

                                        .addGap(14, 14, 14)

                                        .addComponent(cvv, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)))

                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                                    .addComponent(jLabel4)

                                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                                    .addComponent(Banco, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGap(186, 186, 186)))

    .addGap(45, 45, 45)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(inserta, javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)

        .addComponent(elimina, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

    .addGap(55, 55, 55)

    .addComponent(modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(0, 602, Short.MAX_VALUE))))

    .addGroup(layout.createSequentialGroup())

    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 121, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(0, 0, Short.MAX_VALUE)))

    .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addComponent(jLabel1)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(Numero, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(layout.createSequentialGroup())

        .addGap(6, 6, 6)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 49,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(inserta, javax.swing.GroupLayout.PREFERRED_SIZE, 49,
javax.swing.GroupLayout.PREFERRED_SIZE))))

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel3)

        .addComponent(cvv, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```



```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(Banco, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jLabel5)

            .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(11, 11, 11)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel6)

            .addComponent(Cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(layout.createSequentialGroup())

        .addGap(29, 29, 29)

        .addComponent(elimina, javax.swing.GroupLayout.PREFERRED_SIZE, 58,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 285,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap()

    );

    pack();
} // </editor-fold>

```

```

private void BancoActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void ClienteActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void insertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();
}

```

```

        muestra();

        Numero.setText("");

        cvv.setText("");

        Banco.setText("");
    }

    private void modificaActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        modifica();

        muestra();

        Numero.setText("");

        cvv.setText("");

        Banco.setText("");
    }

    private void eliminaActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        elimina();

        muestra();

        Numero.setText("");

        cvv.setText("");

        Banco.setText("");
    }

    // Variables declaration - do not modify
    private javax.swing.JTextField Banco;
    private javax.swing.JComboBox Cliente;
    private com.toedter.calendar.JDateChooser Date;
    private javax.swing.JTextField Numero;
    private javax.swing.JTable Ttarjeta;
    private javax.swing.JTextField cvv;
    private javax.swing.JButton elimina;
    private javax.swing.JButton inserta;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;

```

```
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton modifica;  
// End of variables declaration  
}
```

## Clase Telefono\_Cliente

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Datos;
```

```
  
import Datos.Tarjeta_Cliente;  
import Datos.Telefono_Cliente;  
  
import Usuario.Cliente;  
import Usuario.Proveedor;  
  
import java.sql.Connection;  
  
import java.awt.Point;  
  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.util.LinkedList;  
import javax.swing.JFrame;  
import javax.swing.JOptionPane;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;
```

```
/*  
  
vista donde puedes ver, agregar, modificar o eliminar los teléfonos del cliente  
*/
```

```
  
public class Telefono_Cliente extends javax.swing.JFrame {  
  
    /*  
  
    conjunto de variables necesarias para realizar la conexión a la  
  
    base de datos  
  
    */  
  
    private String HOST="localhost";//host  
  
    private String PUERTO="5432";// puerto
```

```

private String DB="ComprasEnLinea";// base de datos

private String USER="postgres";// usuario

private String PASS="postgres";// contraseña

public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //direccion creada al juntar el host, puerto y base de datos

private Connection conexion=null;// variable para la conexión

public String id;//Variable para id de Cliente

public String numeros;//Variable para guardar los numeros de teléfono

public LinkedList<String> ids=new LinkedList<>();

/**
 * Creates new form Telefono_Cliente
 */
public Telefono_Cliente(String U) {
    initComponents();
    clientes();
    muestra();
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    usuario(U);
}

public void usuario(String u){
    if(u.equals("pablo")){
        Inserta.setEnabled(false);
        Modifica.setEnabled(false);
        Elimina.setEnabled(false);
    }
}

// metodo que obtiene los clientes de la base de datos ordenados de forma ascendente según el Id
public void clientes(){
    try{
        conexion=connectaBD();
        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener los datos del cliente, ordenados de forma ascendente según su Id
        String Query="SELECT Nombre_Cliente,Id_Cliente,Correo_Electronico FROM Usuario.Cliente ORDER BY Id_Cliente ASC";
        corrida=conexion.createStatement();
        ResultSet columnas=corrida.executeQuery(Query);
        while(columnas.next()){
            //inserta el cliente
            Cliente.addItem(columnas.getString(1)+" - "+columnas.getString(3));
        }
    }
}

```

```

        ids.addLast(columnas.getString(2));
    }
} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, ex);
}
}

// conecta a la base de datos a travez del driver, url, usuario y password
public Connection conectaBD(){
    Connection cone=null;

    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }

    return cone;
}

// inserta un teléfono en la base de datos segun el Id del cliente seleccionado
public void inserta(){
    try{
        conexion=conectaBD(); // crea la conexión a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        int id=Cliente.getSelectedIndex(); // selecciona el id del cliente

        // query que realiza la inserción de un nuevo teléfono en la tabla Datos.Teléfono_Cliente
        String Query="INSERT INTO Datos.Telefono_Cliente(Id_Cliente,Numero)"
            + "VALUES('"+ids.get(Cliente.getSelectedIndex())+"','"+Numero.getText()+"')";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

        //JOptionPane.showMessageDialog(null, "INSERTO");
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, "Número Duplicado");
    }
}

//muestra en el jTable los teléfonos que estén relacionadas según el Id del cliente seleccionado

```

```

public void muestra(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query que muestra en el jTable los teléfonos del cliente, relacionando las tablas Datos.Telefono_cliente y Usuario.Cliente,
        ordenadas de forma ascendente

        String Query="SELECT D.Id_Cliente,D.Numero,P.Nombre_Cliente,P.Correo_Electronico FROM Datos.Telefono_Cliente
        D,Usuario.Cliente P WHERE D.Id_Cliente=P.Id_Cliente ORDER BY D.Id_Cliente ASC";

        String[]datos =new String[5];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query);

        DefaultTableModel model=new DefaultTableModel();

        //se agregan las columnas al jTable

        model.addColumn("Id Cliente");

        model.addColumn("Número");

        model.addColumn("Nombre Cliente - Correo Electrónico");

        JTelefonoC.setModel(model);

        //se llenan las columnas del jTable

        while(columnas.next()){

            datos[0]=columnas.getString(1);

            datos[1]=columnas.getString(2);

            datos[2]=columnas.getString(3)+" - "+columnas.getString(4);

            model.addRow(datos);

        }

        //obtiene los datos del teléfono al hacer clic en el renglón del jTable

        JTelefonoC.addMouseListener(new MouseAdapter() {

            public void mousePressed(MouseEvent Mouse_evt){

                JTable table=(JTable) Mouse_evt.getSource();

                Point point = Mouse_evt.getPoint();

                int row=table.rowAtPoint(point);

                if(Mouse_evt.getClickCount()==1){

                    Numero.setText(JTelefonoC.getValueAt(JTelefonoC.getSelectedRow(),1).toString());

                    for (int i =0; i < ids.size(); i++){

                        if(ids.get(i).equals(JTelefonoC.getValueAt(JTelefonoC.getSelectedRow(),0).toString())){

                            Cliente.setSelectedIndex(i);

                        }

                    }

                }

                // obtiene el id del renglón seleccionado

```

```

        id=JTelefonoC.getValueAt(JTelefonoC.getSelectedRow(), 0).toString();

        // obtiene el numero del renglón seleccionado

        numeros=JTelefonoC.getValueAt(JTelefonoC.getSelectedRow(), 1).toString();

    }

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

//método para eliminar una tupla de la tabla Datos.Telefono_Cliente

public void elimina(){

    try{

        conexion=connectaBD(); //crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para eliminar la tupla que corresponda al numero de teléfono de la tabla Datos.Telefono_Cliente

        String Query="DELETE FROM Datos.Telefono_Cliente WHERE Numero='"+Numero.getText()+"'";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

}

//método para modificar una tupla de la tabla Datos.Telefono_cliente

public void modifica(){

    try{

        conexion=connectaBD(); //crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para actualizar los datos de la tupla relacionada con el Id del cliente y el numero de telefono

        String Query="UPDATE Datos.Telefono_Cliente SET
Numero='"+Numero.getText()+"',Id_Cliente='"+ids.getCliente.getSelectedIndex()+"'

        + "WHERE Id_Cliente='"+id+" AND Numero='"+numeros+"'";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

```



```

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "Número Duplicado");

    }

}

/**

 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    Inserta = new javax.swing.JButton();

    Modifica = new javax.swing.JButton();

    Elimina = new javax.swing.JButton();

    jScrollPane1 = new javax.swing.JScrollPane();

    JTelefonoC = new javax.swing.JTable();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    Numero = new javax.swing.JTextField();

    Cliente = new javax.swing.JComboBox();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Teléfono Cliente");

    Inserta.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    Inserta.setText("INSERTA");

    Inserta.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            InsertaActionPerformed(evt);

        }

    });

    Modifica.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    Modifica.setText("MODIFICA");

    Modifica.addActionListener(new java.awt.event.ActionListener() {

```

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    ModificaActionPerformed(evt);  
}  
});
```

```
Elimina.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N  
Elimina.setText("ELIMINA");  
Elimina.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        EliminaActionPerformed(evt);  
    }  
});
```

```
JTelefonoC.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {  
  
    },  
    new String [] {  
        "Title 1", "Title 2", "Title 3", "Title 4"  
    }  
));  
jScrollPane1.setViewportViewView(JTelefonoC);
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 2, 12)); // NOI18N  
jLabel1.setForeground(new java.awt.Color(0, 204, 204));  
jLabel1.setText("Teléfono Cliente");
```

```
jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel2.setText("Número");
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  
jLabel3.setText("Cliente");
```

```
Numero.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
Cliente.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 103, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE))

    .addGroup(layout.createSequentialGroup())

        .addContainerGap()

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jScrollPane1)

            .addGroup(layout.createSequentialGroup())

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                    .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                    .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(Número, javax.swing.GroupLayout.PREFERRED_SIZE, 181,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(Cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 287,
javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGap(33, 33, 33)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                    .addComponent(Modifica, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                    .addComponent(Inserta, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

                .addGap(75, 75, 75)

                .addComponent(Elimina)

                .addGap(119, 119, 119)))

        .addContainerGap()

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup())

                .addComponent(jLabel1)

                .addGap(33, 33, 33)

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(Numero, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent(Cliente, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(56, 56, 56)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 283,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(layout.createSequentialGroup())

        .addGap(36, 36, 36)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

            .addComponent(Inserta, javax.swing.GroupLayout.DEFAULT_SIZE, 35, Short.MAX_VALUE)

            .addComponent(Elimina, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

        .addGap(18, 18, 18)

        .addComponent(Modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addContainerGap(13, Short.MAX_VALUE))

    );

    pack();
}

```

```

private void InsertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();

    muestra();

    Numero.setText("");

}

```

```

private void EliminaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    elimina();

    muestra();

    Numero.setText("");

}

```

```
private void ModificaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    modifica();  
    muestra();  
    Numero.setText("");  
  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JComboBox Cliente;  
private javax.swing.JButton Elimina;  
private javax.swing.JButton Inserta;  
private javax.swing.JTable JTelefonoC;  
private javax.swing.JButton Modifica;  
private javax.swing.JTextField Numero;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JScrollPane jScrollPane1;  
  
// End of variables declaration  
}
```

## Clase Telefono\_Proveedor

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Datos;

import Datos.Tarjeta_Cliente;
import Datos.Telefono_Cliente;
import Usuario.Proveedor;
import java.sql.Connection;
import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.LinkedList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/*
 vista donde puedes ver, agregar, modificar o eliminar los teléfonos del proveedor
 */

public class Telefono_Proveedor extends javax.swing.JFrame {

    /*
    conjunto de variables necesarias para realizar la conexión a la
    base de datos
    */

    private String HOST="localhost"; //host
    private String PUERTO="5432"; // puerto
    private String DB="ComprasEnLinea"; // base de datos
    private String USER="postgres"; // usuario
    private String PASS="postgres"; // contraseña
```

```

public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //direccion creada al juntar el host, puerto y base de datos

private Connection conexion=null; // variable para la conexión

public String id; //Variable para id del Proveedor

public String numeros; //Variable para guardar los numeros de teléfono

public LinkedList<String> ids=new LinkedList<>();

/**
 * Creates new form Telefono_Proveedor
 */
public Telefono_Proveedor(String U) {

    initComponents();

    muestra();

    proveedores();

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    usuario(U);
}

public void usuario(String u){

    if(u.equals("pablo")){

        Inserta.setEnabled(false);

        Modifica.setEnabled(false);

        Elimina.setEnabled(false);

    }

}

// metodo que obtiene los Proveedores de la base de datos ordenados de forma ascendente según el Id

public void proveedores(){

    try{

        conexion=connectaBD(); // crea la conexión a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener los datos del proveedor, ordenados de forma ascendente según su Id

        String Query="SELECT Nombre_Proveedor,Id_Proveedor,Correo_Electronico FROM Usuario.Proveedor ORDER BY
Id_Proveedor ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecutar el query

        while(columnas.next()){

            Proveedor.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            ids.addLast(columnas.getString(2));

        }

    }

}

```

```

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// conecta a la base de datos a travez del driver, url, usuario y password
public Connection conectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

// inserta un teléfono en la base de datos segun el Id del proveedor seleccionado
public void inserta(){

    try{

        conexion=conectaBD(); // crea la conexion a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        int id=Proveedor.getSelectedIndex();// selecciona el id del proveedor

        // query que realiza la inserción de un nuevo teléfono en la tabla Datos.Teléfono_Proveedor

        String Query="INSERT INTO Datos.Teléfono_Proveedor(Id_Proveedor,Numero)"

            + "VALUES('" +ids.get(Proveedor.getSelectedIndex())+"','"+Numero.getText()+"')";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close();// cierra la conexion con la base de datos

        //JOptionPane.showMessageDialog(null, "INSERTO");

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "Número Duplicado");

    }

}

//muestra en el jTable los teléfonos que estén relacionadas según el Id del proveedor seleccionado
public void muestra(){

    try{

```



```

conexion=connectaBD(); //crea la conexion con la base de datos

java.sql.Statement corrida=conexion.createStatement();

//query que muestra en el jTable los teléfonos del proveedor, relacionando las tablas Datos.Telefono_Proveedor y
Usuario.Proveedor, ordenadas de forma ascendente

String Query="SELECT D.Id_Proveedor,D.Numero,P.Nombre_Proveedor,P.Correo_Electronico FROM
Datos.Telefono_Proveedor D,Usuario.Proveedor P WHERE D.Id_Proveedor=P.Id_Proveedor ORDER BY D.Id_Proveedor ASC";

String[] datos =new String[5];

corrida=conexion.createStatement();

ResultSet columnas=corrida.executeQuery(Query);

DefaultTableModel model=new DefaultTableModel();

//se agregan las columnas al jTable

model.addColumn("Id Proveedor");

model.addColumn("Número");

model.addColumn("Nombre Proveedor - Correo Electrónico");

JTelefonoP.setModel(model);

//se llenan las columnas del jTable

while(columnas.next()){

    datos[0]=columnas.getString(1);

    datos[1]=columnas.getString(2);

    datos[2]=columnas.getString(3)+" - "+columnas.getString(4);

    model.addRow(datos);

}

//obtiene los datos del teléfono al hacer clic en el renglón del jTable

JTelefonoP.addMouseListener(new MouseAdapter() {

    public void mousePressed(MouseEvent Mouse_evt){

        JTable table=(JTable) Mouse_evt.getSource();

        Point point = Mouse_evt.getPoint();

        int row=table.rowAtPoint(point);

        if(Mouse_evt.getClickCount()==1){

            Numero.setText(JTelefonoP.getValueAt(JTelefonoP.getSelectedRow(),1).toString());

            for (int i =0; i < ids.size(); i++){

                if(ids.get(i).equals(JTelefonoP.getValueAt(JTelefonoP.getSelectedRow(),0).toString())){

                    Proveedor.setSelectedIndex(i);

                }

            }

            // obtiene el id del renglón seleccionado

            id=JTelefonoP.getValueAt(JTelefonoP.getSelectedRow(), 0).toString();

            // obtiene el numero del renglón seleccionado

```

```

        numeros=JTelefonoP.getValueAt(JTelefonoP.getSelectedRow(), 1).toString();

    }

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

//método para elimiar una tupla de la tabla Datos.Telefono_Proveedor

public void elimina(){

    try{

        conexion=connectaBD();//crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para eliminar la tupla que corresponda al numero de teléfono de la tabla Datos.Telefono_Proveedor

        String Query="DELETE FROM Datos.Telefono_Proveedor WHERE Numero='"+Numero.getText()+"'";

        corrida.executeUpdate(Query); // ejecutar query

        corrida.close();

        conexion.close(); // cierra la conexión con la base de dayos

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//método para modificar una tupla de la tabla Datos.Telefono_Proveedor

public void modifica(){

    try{

        conexion=connectaBD();//crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para actualizar los datos de la tupla relacionada con el Id del proveedor y el numero de telefono

        String Query="UPDATE Datos.Telefono_Proveedor SET
Numero='"+Numero.getText()+"',Id_Proveedor='"+ids.get(Proveedor.getSelectedIndex())+"'
        + "WHERE Id_Proveedor="+id+" AND Numero='"+numeros+"'";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexion con la base de datos

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "Numero Duplicado");

    }

}

```

```

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel3 = new javax.swing.JLabel();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    Numero = new javax.swing.JTextField();

    Proveedor = new javax.swing.JComboBox();

    jScrollPane1 = new javax.swing.JScrollPane();

    JTelefonoP = new javax.swing.JTable();

    Inserta = new javax.swing.JButton();

    Modifica = new javax.swing.JButton();

    Elimina = new javax.swing.JButton();

    jLabel4 = new javax.swing.JLabel();

    jLabel3.setText("jLabel3");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Teléfono Proveedor");

    jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
    jLabel1.setText("Número");

    jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
    jLabel2.setText("Proveedor");

    Proveedor.setToolTipText("");

    JTelefonoP.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

```

```

    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
});

jScrollPane1.setViewportViewView(JTelefonoP);

Inserta.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Inserta.setText("INSERTA");
Inserta.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        InsertaActionPerformed(evt);
    }
});

Modifica.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Modifica.setText("MODIFICA");
Modifica.setToolTipText("");
Modifica.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ModificaActionPerformed(evt);
    }
});

Elimina.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Elimina.setText("ELIMINA");
Elimina.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EliminaActionPerformed(evt);
    }
});

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel4.setForeground(new java.awt.Color(0, 153, 204));
jLabel4.setText("Teléfono Proveedor");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```

```

getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel4)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 912, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(18, 18, 18)
                    .addComponent(Numero)
                    .addGap(112, 112, 112)
                    .addComponent(Inserta, javax.swing.GroupLayout.PREFERRED_SIZE, 106, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(Elimina, javax.swing.GroupLayout.PREFERRED_SIZE, 95, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(Modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(167, 167, 167))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(Proveedor, javax.swing.GroupLayout.PREFERRED_SIZE, 347,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        );

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel4)
            .addGap(15, 15, 15)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(Numero, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(Inserta)

```

```

        .addComponent(Elimina)

        .addComponent(Modifica, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(12, 12, 12)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

        .addComponent(Proveedor, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(33, 33, 33)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 275,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void InsertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();

    muestra();

    Numero.setText("");

}

```

```

private void EliminaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    elimina();

    muestra();

    Numero.setText("");

}

```

```

private void ModificaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    modifica();

    muestra();

}

```

```
// Variables declaration – do not modify

private javax.swing.JButton Elimina;

private javax.swing.JButton Inserta;

private javax.swing.JTable JtelefonoP;

private javax.swing.JButton Modifica;

private javax.swing.JTextField Numero;

private javax.swing.JComboBox Proveedor;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JScrollPane jScrollPane1;

// Clase47 variables declaration
}
```

## Clase Menu

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package Menu;
```

```
  
import Artículo.Producto;  
import Operaciones.Carrito_Venta;  
import Operaciones.Orden;  
import Reportes.Reporte1;  
import Reportes.Reporte2;  
import Usuario.Cambio_Usuario;  
import Usuario.Cliente;  
import Usuario.Proveedor;  
import java.sql.Connection;  
import javax.swing.JOptionPane;
```

```
/*  
Menú inicial de Compras en Línea  
*/  
  
public class Menu extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Menu  
     */  
  
    public String user;  
  
    public Menu(String u) {  
        initComponents();  
        usuario(u);  
        user=u;  
    }  
  
    public void usuario(String u){  
        if(u.equals("andrea")){  
            Orden.setEnabled(false);  
            Carrito.setEnabled(false);  
        }  
    }  
}
```



```

    }
}
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();

    Cliente = new javax.swing.JButton();

    Producto = new javax.swing.JButton();

    Carrito = new javax.swing.JButton();

    Proveedor = new javax.swing.JButton();

    Orden = new javax.swing.JButton();

    Usuario = new javax.swing.JButton();

    Reporte1 = new javax.swing.JButton();

    Reporte2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Menú");

    setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

    jLabel1.setFont(new java.awt.Font("Kristen ITC", 0, 36)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(0, 204, 204));
    jLabel1.setText("MENÚ");

    Cliente.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
    Cliente.setText("Cliente");

    Cliente.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ClienteActionPerformed(evt);
        }
    });

    Producto.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N

```

```
Producto.setText("Producto");

Producto.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ProductoActionPerformed(evt);

    }

});
```

```
Carrito.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
Carrito.setText("Carrito");

Carrito.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        CarritoActionPerformed(evt);

    }

});
```

```
Proveedor.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
Proveedor.setText("Proveedor");

Proveedor.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ProveedorActionPerformed(evt);

    }

});
```

```
Orden.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
Orden.setText("Orden");

Orden.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        OrdenActionPerformed(evt);

    }

});
```

```
Usuario.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
Usuario.setText("Usuario");

Usuario.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        UsuarioActionPerformed(evt);

    }

});
```

```

Reporte1.setFont(new java.awt.Font("sansserif", 0, 24)); // NOI18N
Reporte1.setText("Reporte 1");
Reporte1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Reporte1ActionPerformed(evt);
    }
});

Reporte2.setFont(new java.awt.Font("sansserif", 0, 24)); // NOI18N
Reporte2.setText("Reporte 2");
Reporte2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Reporte2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(217, 217, 217)
            .addComponent(jLabel1)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(74, 74, 74)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(Reporte1)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(Carrito, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(Producto, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(Cliente, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 168, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(Orden, javax.swing.GroupLayout.PREFERRED_SIZE, 101, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(Proveedor)

        .addComponent(Usuario)

        .addComponent(Reporte2))

    .addGap(73, 73, 73))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

        .addContainerGap()

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 42, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(26, 26, 26)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(Cliente, javax.swing.GroupLayout.PREFERRED_SIZE, 42, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(Proveedor, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(47, 47, 47)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(Producto, javax.swing.GroupLayout.PREFERRED_SIZE, 44, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(Orden, javax.swing.GroupLayout.PREFERRED_SIZE, 43, javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGap(31, 31, 31)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(Carrito, javax.swing.GroupLayout.PREFERRED_SIZE, 50, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(Usuario))

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21, Short.MAX_VALUE)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                        .addComponent(Reporte1)

                        .addComponent(Reporte2)))

            );

    pack();

} // </editor-fold>

```

```

private void ProductoActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    Producto p=new Producto(user);

    p.setVisible(true);

}

```

```
private void ClienteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Cliente c=new Cliente(user);  
    c.setVisible(true);  
}
```

```
private void ProveedorActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Proveedor pr=new Proveedor(user);  
    pr.setVisible(true);  
}
```

```
private void OrdenActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Orden pr=new Orden(user);  
    pr.setVisible(true);  
}
```

```
private void CarritoActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Carrito_Venta c= new Carrito_Venta(user);  
    c.setVisible(true);  
}
```

```
private void UsuarioActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Cambio_Usuario c=new Cambio_Usuario();  
    c.setVisible(true);  
    this.setVisible(false);  
}
```

```
private void Reporte1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Reporte1 c=new Reporte1();  
    c.setVisible(true);  
}
```

```

private void Reporte2ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    Reporte2 c=new Reporte2();

    c.setVisible(true);

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    }

    //</editor-fold>

    /* Create and display the form */

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new Menu("").setVisible(true);

        }

    }

```

```
});  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JButton Carrito;  
private javax.swing.JButton Cliente;  
private javax.swing.JButton Orden;  
private javax.swing.JButton Producto;  
private javax.swing.JButton Proveedor;  
private javax.swing.JButton Reporte1;  
private javax.swing.JButton Reporte2;  
private javax.swing.JButton Usuario;  
private javax.swing.JLabel jLabel1;  
  
// End of variables declaration  
}
```

## Clase Carrito\_Venta

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Operaciones;

import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

/**
 * vista donde puedes ver las ventas del carrito, agregar, modificar o eliminar.
 */

public class Carrito_Venta extends javax.swing.JFrame {

    private String HOST="localhost"; //host

    private String PUERTO="5432"; //puerto

    private String DB="ComprasEnLinea"; //base de datos

    private String USER="postgres"; //usuario

    private String PASS="postgres"; // contraseña

    public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url

    private Connection conexion=null; // variable para la base de datos

    public LinkedList<String> ids=new LinkedList<>();
```



```

public String user; // usuario

/**
 * Creates new form Carrito_Venta
 */
public Carrito_Venta(String U) {

    initComponents();

    muestra();

    fecha();

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    clientes();

    usuario(U);

    user=U;

}

public void usuario(String u){

    if(u.equals("pablo")){

        INSERTAR.setEnabled(false);

        MODIFICAR.setEnabled(false);

        ELIMINAR.setEnabled(false);

    }

}

/ // muestra los clientes que esten realizando compras

public void clientes(){

    try{

        conexion=connectaBD(); //crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query que obtiene todos los clientes que esten realizando compras, junto con su forma de pago

        String Query="SELECT U.Nombre_Cliente,T.Id_Tarjeta,Correo_Electronico FROM Usuario.Cliente U,Datos.Tarjeta_Cliente T
WHERE U.Id_Cliente=T.Id_Cliente ORDER BY U.Id_Cliente ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

        //agrega la informacion al jTable

        while(columnas.next()){

            Clientes.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            ids.addLast(columnas.getString(2));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);
    }
}

```

```

    }
}

//conecta con la base de datos
public Connection conectaBD(){
    Connection cone=null;

    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }

    return cone;
}

//da formato a la fecha
public void fecha() {
    Calendar fechaActual = Calendar.getInstance();

    // Formatear la fecha actual a un formato de String
    SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
    String fechaFormateada = formatoFecha.format(fechaActual.getTime());
    fecha.setText(fechaFormateada);
}

// muestra todas las compras en el jTable
public void muestra(){
    try{
        conexion=conectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para mostrar las compras junto con el nombre del client, la forma de pago y fecha de la compra
        String Query="SELECT
C.Id_Carrito,Cl.Nombre_Cliente,C.Forma_De_Pago,to_char(Fecha_Venta,'yyyy/MM/dd'),C.Total_Carrito,T.Banco,RIGHT(T.Numero_Tarjeta,4),Cl.Id_Cliente,Cl.Correo_Electronico\n" +
"      FROM Operaciones.Carrito_Venta C,Usuario.Cliente Cl,Datos.Tarjeta_Cliente T\n" +
"      WHERE C.Id_Tarjeta_Cliente=T.Id_Tarjeta AND T.Id_Cliente=Cl.Id_Cliente\n" +
"      ORDER BY C.Id_Carrito\n" +
"      ";

        String[]datos =new String[7];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

        DefaultTableModel model=new DefaultTableModel();

```

```

// crea las columnas para almacenar la información
model.addColumn("Id Carrito");
model.addColumn("Nombre Cliente - Correo Electrónico");
model.addColumn("Forma De Pago");
model.addColumn("Fecha Venta");
model.addColumn("Total");
model.addColumn("Banco");
model.addColumn("Número Tarjeta");

// llena las columnas con los datos obtenidos
carritos.setModel(model);

while(columnas.next()){
    datos[0]=columnas.getString(1);
    datos[1]=columnas.getString(2)+" - "+columnas.getString(9);
    datos[2]=columnas.getString(3);
    datos[3]=columnas.getString(4);
    datos[4]=columnas.getString(5);
    datos[5]=columnas.getString(6);
    datos[6]=columnas.getString(7);

    model.addRow(datos);
}

int columnWidth = 150; // Tamaño de columna en píxeles
int columnIndex = 1; // Índice de la columna a modificar
carritos.getColumnModel().getColumn(columnIndex).setPreferredWidth(columnWidth);

// obtiene los datos de la compra al hacer clic en el renglon del jTable
carritos.addMouseListener(new MouseAdapter() {
    //al hacer clic dentro del jTable, obtiene los datos del rengón
    public void mousePressed(MouseEvent Mouse_evt){
        JTable table=(JTable) Mouse_evt.getSource();
        Point point = Mouse_evt.getPoint();
        int row=table.rowAtPoint(point); // numero del renglon
        if(Mouse_evt.getClickCount()==1){
            String valor = carritos.getValueAt(carritos.getSelectedRow(), 6).toString();
            String id = busqueda(valor);
            for (int i =0; i < ids.size(); i++){
                if(ids.get(i).equals(id)){
                    Clientes.setSelectedIndex(i);
                }
            }
        }
    }
});

```

```

    }

    }

    // obtiene los datos segun el id

    id=carritos.getValueAt(carritos.getSelectedRow(), 0).toString();

    total.setText(carritos.getValueAt(carritos.getSelectedRow(), 4).toString());

    if(carritos.getValueAt(carritos.getSelectedRow(), 2).toString().equals("Debito")){

        Pago.setSelectedIndex(0);

    }else{

        Pago.setSelectedIndex(1);

    }

    try{

        conexion=connectaBD(); // hace la conexion con la base de datos

        java.sql.Statement corrida;

        //query que obtiene la fecha convertida en char

        String Query="SELECT to_char(Fecha_Venta,'yyyy/MM/dd') FROM Operaciones.Carrito_Venta WHERE
Id_Carrito="+id+"";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query);

        while(columnas.next()){

            fecha.setText(columnas.getString(1));

        }

        corrida.close();

        conexion.close();

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

}

//obitene la compra segun la tarjeta

public String busqueda(String cadena){

    String Query1="";

```

```

try{

    conexion=connectaBD(); // crea la conexion con la base de datos

    java.sql.Statement corrida;

    // query que obtiene todas las compras realizadas con el numero de tarjeta

    Query1="SELECT Id_Tarjeta FROM Datos.Tarjeta_Cliente WHERE RIGHT(Numero_Tarjeta,4)='"+cadena+"'";

    corrida=conexion.createStatement();

    ResultSet columnas=corrida.executeQuery(Query1); // ejecuta el query

    while(columnas.next()){

        Query1=columnas.getString(1);

    }

    corrida.close();

    conexion.close();// cierra la conexion con la base de datos

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

return Query1;

}

// inserta una nueva compra/venta

public void inserta(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para insertar una nueva venta en la base de datos

        String Query="INSERT INTO Operaciones.Carrito_Venta(Id_Tarjeta_Cliente,Forma_De_Pago,Fecha_Venta,Total_Carrito)"

            +

            "VALUES('"+ids.getClientes.getSelectedIndex()+"','"+Pago.getSelectedItem()+"','"+fecha.getText()+"','"+total.getText()+"')";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// elimina una venta del carrito

public void elimina(){

    try{

```

```

        conexion=connectaBD(); // crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para eliminar una venta del carrito

        String Query="DELETE FROM Operaciones.Carrito_Venta WHERE Id_Carrito="+carritos.getValueAt(carritos.getSelectedRow(),
0).toString()+" ";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "No se puede eliminar porque tiene productos en el");

    }

}

// modifica la venta del carrito

public void modifica(){

    try{

        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        Pattern patron = Pattern.compile("\\d+");

        Matcher matcher = patron.matcher(carritos.getValueAt(carritos.getSelectedRow(), 0).toString());

        String num="";

        while (matcher.find()) {

            num = matcher.group();

        }

        //query para actualizar la venta del carito

        String Query="UPDATE Operaciones.Carrito_Venta SET "

            + "Id_Tarjeta_Cliente=" +ids.get(Clientes.getSelectedIndex())+" , Fecha_Venta="+fecha.getText()+"", Total_Carrito=
""+carritos.getValueAt(carritos.getSelectedRow(), 4).toString()+" "

            + "WHERE Id_Carrito="+num+"";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close();// cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    INSERTAR = new javax.swing.JButton();

    MODIFICAR = new javax.swing.JButton();

    DEVOLUCION = new javax.swing.JButton();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jLabel4 = new javax.swing.JLabel();

    jLabel5 = new javax.swing.JLabel();

    ELIMINAR = new javax.swing.JButton();

    DETALLE = new javax.swing.JButton();

    jScrollPane1 = new javax.swing.JScrollPane();

    carritos = new javax.swing.JTable();

    Clientes = new javax.swing.JComboBox();

    Pago = new javax.swing.JComboBox();

    total = new javax.swing.JLabel();

    fecha = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Carrito Venta");

    INSERTAR.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
    INSERTAR.setText("INSERTAR");

    INSERTAR.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            INSERTARActionPerformed(evt);
        }
    });

    MODIFICAR.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N

```

```
MODIFICAR.setText("MODIFICAR");

MODIFICAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        MODIFICARActionPerformed(evt);

    }

});

DEVOLUCION.setBackground(new java.awt.Color(0, 0, 0));

DEVOLUCION.setForeground(new java.awt.Color(0, 255, 255));

DEVOLUCION.setText("DEVOLUCIÓN");

DEVOLUCION.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DEVOLUCIONActionPerformed(evt);

    }

});

jLabel1.setForeground(new java.awt.Color(255, 0, 51));

jLabel1.setText("Carrito Venta");

jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel2.setText("Cliente");

jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel3.setText("Forma De Pago");

jLabel4.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel4.setText("Fecha");

jLabel5.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel5.setText("Total");

ELIMINAR.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
ELIMINAR.setText("ELIMINAR");

ELIMINAR.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ELIMINARActionPerformed(evt);

    }

});
```



```

DETALLE.setBackground(new java.awt.Color(0, 0, 0));

DETALLE.setForeground(new java.awt.Color(0, 255, 255));

DETALLE.setText("DETALLE CARRITO");

DETALLE.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DETALLEActionPerformed(evt);

    }

});

carritos.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        },

    new String [] {

        "Title 1", "Title 2", "Title 3", "Title 4"

    }

));

jScrollPane1.setViewportView(carritos);

Clientes.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ClientesActionPerformed(evt);

    }

});

Pago.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Debito", "Credito" }));

total.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
total.setForeground(new java.awt.Color(255, 51, 51));
total.setText("$0.00");

fecha.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
fecha.setText("fecha/Hola");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 163, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(total))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(fecha))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addComponent(Pago, javax.swing.GroupLayout.PREFERRED_SIZE, 221,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 132, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(Clientes, javax.swing.GroupLayout.PREFERRED_SIZE, 387,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGap(107, 107, 107)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(INSERTAR, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(MODIFICAR)

.addComponent(ELIMINAR, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGap(84, 84, 84)

.addComponent(DEVOLUCION, javax.swing.GroupLayout.PREFERRED_SIZE, 137,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(45, 45, 45)

```

```

        .addComponent(DETALLE)

        .addGap(245, 245, 245))

        .addComponent(jScrollPane1)

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup())

                .addGap(18, 18, 18)

                .addComponent(INSERTAR)

                .addGap(18, 18, 18)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(MODIFICAR)

                    .addComponent(DEVOLUCION, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(DETALLE, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE))

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                .addComponent(ELIMINAR))

            .addGroup(layout.createSequentialGroup())

                .addComponent(jLabel1)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 16,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(Clientes, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGap(7, 7, 7)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jLabel3)

                    .addComponent(Pago, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(fecha)

                    .addComponent(jLabel4))

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addGroup(layout.createSequentialGroup())

```

```

        .addGap(10, 10, 10)

        .addComponent(jLabel5))

        .addGroup(layout.createSequentialGroup())

        .addGap(3, 3, 3)

        .addComponent(total)))

        .addGap(0, 0, Short.MAX_VALUE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 282,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(14, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void ClientesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void DETALLEActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    Detalle_Carrito d=new Detalle_Carrito(user);

    d.setVisible(true);

    this.setVisible(false);
}

```

```

private void DEVOLUCIONActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    Devolucion_Carrito d=new Devolucion_Carrito(user);

    d.setVisible(true);

    this.setVisible(false);
}

```

```

private void INSERTARActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    inserta();

    muestra();

    total.setText("$0.00");
}

```

```
}
```

```
private void ELIMINARActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    elimina();
```

```
    muestra();
```

```
    total.setText("$0.00");
```

```
}
```

```
private void MODIFICARActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    modifica();
```

```
    muestra();
```

```
    total.setText("$0.00");
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JComboBox Clientes;
```

```
private javax.swing.JButton DETALLE;
```

```
private javax.swing.JButton DEVOLUCION;
```

```
private javax.swing.JButton ELIMINAR;
```

```
private javax.swing.JButton INSERTAR;
```

```
private javax.swing.JButton MODIFICAR;
```

```
private javax.swing.JComboBox Pago;
```

```
private javax.swing.JTable carritos;
```

```
private javax.swing.JLabel fecha;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JLabel jLabel5;
```

```
private javax.swing.JScrollPane jScrollPane1;
```

```
private javax.swing.JLabel total;
```

```
// End of variables declaration
```

```
}
```

## Clase Detalle\_Carrito

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Operaciones;

import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/**
 * vista donde se muestran los detalles del carrito
 */
public class Detalle_Carrito extends javax.swing.JFrame {

    private String HOST="localhost";//host
    private String PUERTO="5432";//puerto
    private String DB="ComprasEnLinea";//base de datos
    private String USER="postgres";//usuario
    private String PASS="postgres";// contraseña
    public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url
    private Connection conexion=null;// variable para la base de datos
    public LinkedList<String> idsProductos=new LinkedList<>();
```

```

public LinkedList<String> idsCarritos=new LinkedList<>();

public String user;// usuario

/**
 * Creates new form Detalle_Carrito
 */

public Detalle_Carrito(String U) {

    initComponents();

    muestra();

    productos();

    carritos();

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    cierra();

    usuario(U);

    user=U;

}

public void usuario(String u){

    if(u.equals("pablo")){

        INSERTA.setEnabled(false);

        MODIFICA.setEnabled(false);

        ELIMINA.setEnabled(false);

    }

}

//conecta con la base de datos

public Connection conectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

// cierra la vista

public void cierra(){

    this.addWindowListener(new WindowAdapter() {

        public void windowClosing(WindowEvent e) {

            Carrito_Venta d=new Carrito_Venta(user);

```

```

        d.setVisible(true);
    }
});
}

// muestra los carritos en el jTable

public void carritos(){
    try{
        conexion=connectaBD(); // crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener todos los carritos de la base de datos de forma ascendente según su Id

        String Query="SELECT C.Nombre_Cliente,O.Id_Carrito \n" +

        "FROM Operaciones.Carrito_Venta O,Usuario.Cliente C,Datos.Tarjeta_Cliente T \n" +

        "WHERE C.Id_Cliente=T.Id_Cliente AND T.Id_Tarjeta=O.Id_Tarjeta_Cliente \n" +

        "ORDER BY O.Id_Carrito ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

        while(columnas.next()){

            Carritos.addItem(columnas.getString(2)+" - "+columnas.getString(1));

            idsCarritos.addLast(columnas.getString(2));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//obtiene los productos de la base de datos

public void productos(){

    try{

        conexion=connectaBD(); // crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para obtener los productos de la base de datos

        String Query="SELECT Nombre_Producto,Id_Producto,Precio_Publico_Producto FROM Artículo.Producto ORDER BY

Id_Producto ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecutar el query

        while(columnas.next()){

            Productos.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            idsProductos.addLast(columnas.getString(2));

        }

    }

}

```



```

    }

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, ex);

}

}

// inserta un nuevo detalle carrito

public void inserta() {

    try {

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para insertar un nuevo detalle carrito

        String Query="INSERT INTO Operaciones.Detalle_Carrito(Id_Carrito,Id_Producto,Cantidad,SubTotal)"

            +

            "VALUES(" + idsCarritos.get(Carritos.getSelectedIndex()) + "," + idsProductos.get(Productos.getSelectedIndex()) + "," + Cantidad.getText(
            ) + ",NULL)";

        corrida.executeUpdate(Query); // ejecutar el query

        corrida.close();

        conexion.close(); // cierra la conexión

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Sobrepasa los productos en existencia");

    }

}

// muestra todos los detalles carrito en el jTable

public void muestra() {

    try {

        conexion=connectaBD(); // crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener el detalle de cada carrito, concatena el id del carrito, los datos del cliente con su forma de pago y los
        // detalles del producto

        String Query="SELECT CONCAT(C.Id_Carrito,' - ' + Cl.Nombre_Cliente),D.Cantidad,D.SubTotal,P.Nombre_Producto,P.Precio_Publico_Producto \n" +

            "FROM Operaciones.Detalle_Carrito D,Operaciones.Carrito_Venta C,Usuario.Cliente Cl,Datos.Tarjeta_Cliente

            T,Articulo.Producto P\n" +

            "WHERE C.Id_Carrito=D.Id_Carrito AND C.Id_Tarjeta_Cliente=T.Id_Tarjeta AND T.Id_Cliente=Cl.Id_Cliente AND

            P.Id_Producto=D.Id_Producto\n" +

            "ORDER BY D.Id_Carrito ASC";

        String[] datos = new String[7];

        corrida=conexion.createStatement();

```

```

ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

DefaultTableModel model=new DefaultTableModel();

//agrega las columnas en el jTable

model.addColumn("Carrito Cliente");

model.addColumn("Cantidad");

model.addColumn("SubTotal");

model.addColumn("Nombre Producto - Precio Público Producto");

// inserta la informacion en el jTable

Detalles.setModel(model);

while(columnas.next()){

    datos[0]=columnas.getString(1);

    datos[1]=columnas.getString(2);

    datos[2]=columnas.getString(3);

    datos[3]=columnas.getString(4)+" - "+columnas.getString(5);


    model.addRow(datos);

}

// obtiene los datos del renglón al hacer clic dentro del jTable

Detalles.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent Mouse_evt){

    JTable table=(JTable) Mouse_evt.getSource();

    Point point = Mouse_evt.getPoint();

    int row=table.rowAtPoint(point);

    if(Mouse_evt.getClickCount()==1){

        // llena los jTextField de cantidad y subtotal

        Cantidad.setText(Detalles.getValueAt(Detalles.getSelectedRow(),1).toString());

        subtotal.setText(Detalles.getValueAt(Detalles.getSelectedRow(),2).toString());

        String valor = Detalles.getValueAt(Detalles.getSelectedRow(), 3).toString();

        String id = buscaProducto(valor);

        for (int i =0; i < idsProductos.size(); i++){

            if(idsProductos.get(i).equals(id)){

                Productos.setSelectedIndex(i);

            }

        }

        String valor2 = Detalles.getValueAt(Detalles.getSelectedRow(), 0).toString();

        Pattern patron = Pattern.compile("\\d+");

        Matcher matcher = patron.matcher(valor2);

```

```

        String num="";

        while (matcher.find()) {

            num = matcher.group();

        }

        for (int i =0; i < idsCarritos.size(); i++){

            if(idsCarritos.get(i).equals(num)){

                Carritos.setSelectedIndex(i);

            }

        }

    }

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

// método que obtiene un producto de la tabla Artículo.Producto

public String buscaProducto(String cadena){

    String Query1=""; // variable para el query

    try{

        conexion=connectaBD(); // conexion de la base de datos

        java.sql.Statement corrida;

        // query que obtiene los datos del producto segun su nombre

        Query1="SELECT Id_Producto FROM Artículo.Producto WHERE Nombre_Producto='"+cadena+"'";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query1);

        while(columnas.next()){

            Query1=columnas.getString(1);

        }

        corrida.close();

        conexion.close();// cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return Query1;

}

```

```

//elimina detalle del carrito

public void elimina(){

    try{

        conexion=connectaBD(); // cra conexion a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para eliminar carrito donde el Id y el Id del producto coincidan en la base de datos

        String Query="DELETE FROM Operaciones.Detalle_Carrito WHERE
Id_Carrito="+idsCarritos.get(Carritos.getSelectedIndex())+" AND "

        + "Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+" "

        + "AND Cantidad="+Cantidad.getText()+" AND SubTotal='"+subtotal.getText()+"'";

        corrida.executeUpdate(Query); //ejecutar query

        corrida.close();

        conexion.close(); // cierra la conexión

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// modificar el detalle del carrito

public void modifica(){

    try{

        conexion=connectaBD(); //crea la conexion

        java.sql.Statement corrida=conexion.createStatement();

        String i=Detalles.getValueAt(Detalles.getSelectedRow(), 0).toString();

        Pattern patron = Pattern.compile("\\d+");

        Matcher matcher = patron.matcher(i);

        String num="";

        while (matcher.find()) {

            num = matcher.group();

        }

        String Ca=Detalles.getValueAt(Detalles.getSelectedRow(),1).toString();

        String valor = Detalles.getValueAt(Detalles.getSelectedRow(), 3).toString();

        String idProducto = buscaProducto(valor);

        //query para modificar el carrito, obitene los datos al hacer clic en el renglom dentro del jTable

        String Query="UPDATE Operaciones.Detalle_Carrito SET "

        +

        "Id_Carrito="+idsCarritos.get(Carritos.getSelectedIndex())+" ,Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+" ,Can
tidad="+Cantidad.getText()+" ,SubTotal=NULL "

        + " WHERE Id_Carrito="+num+" AND Cantidad="+Ca+" AND Id_Producto="+idProducto+"";

```

```

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion con la base de datos

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null,"Sobrepasa los productos en existencia");

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    subtotal = new javax.swing.JLabel();
    INSERTA = new javax.swing.JButton();
    MODIFICA = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    Detalles = new javax.swing.JTable();
    ELIMINA = new javax.swing.JButton();
    Carritos = new javax.swing.JComboBox();
    Productos = new javax.swing.JComboBox();
    Cantidad = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Detalle Carrito");

    jLabel1.setFont(new java.awt.Font("sansserif", 3, 14)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(0, 0, 204));
    jLabel1.setText("Detalle Carrito");

```

```
jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel2.setText("Carrito");
```

```
jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel3.setText("Producto");
```

```
jLabel4.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel4.setText("Cantidad");
```

```
jLabel5.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel5.setText("SubTotal");
```

```
subtotal.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
subtotal.setForeground(new java.awt.Color(255, 51, 51));
subtotal.setText("$0.00");
```

```
INSERTA.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
INSERTA.setText("INSERTA");
INSERTA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        INSERTAActionPerformed(evt);
    }
});
```

```
MODIFICA.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
MODIFICA.setText("MODIFICA");
MODIFICA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        MODIFICAActionPerformed(evt);
    }
});
```

```
Detalles.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        ,
    },
    new String [] {
```

```

        "Title 1", "Title 2", "Title 3", "Title 4"
    }
});

jScrollPane1.setViewportViewView(Detalles);

ELIMINA.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N
ELIMINA.setText("ELIMINA");
ELIMINA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ELIMINAActionPerformed(evt);
    }
});

Cantidad.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(subtotal)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 251, Short.MAX_VALUE)
                    .addComponent(jLabel5)
                )
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel4)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                )
            )
            .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 120, Short.MAX_VALUE)
            .addComponent(jLabel2)
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    )
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(subtotal)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 251, Short.MAX_VALUE)
                        .addComponent(jLabel5)
                    )
                )
            )
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(subtotal)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 251, Short.MAX_VALUE)
                            .addComponent(jLabel5)
                        )
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel4)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, 83,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 120, Short.MAX_VALUE)
                            .addComponent(jLabel2)
                        )
                    )
                )
            )
        )
    );

```

```

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel2)

        .addGap(37, 37, 37)

        .addComponent(Carritos, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel3)

        .addGap(18, 18, 18)

        .addComponent(Productos, javax.swing.GroupLayout.PREFERRED_SIZE, 217,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(MODIFICA)

        .addComponent(INSERTA, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(ELIMINA, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE))))

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1122,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap())

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

        .addComponent(Carritos, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(INSERTA))

        .addGap(8, 8, 8)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel3)

        .addComponent(Productos, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel4)

```



```

        .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(subtotal)))

        .addGroup(layout.createSequentialGroup()

            .addComponent(MODIFICA)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(ELIMINA)))

        .addGap(36, 36, 36)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 320,
javax.swing.GroupLayout.PREFERRED_SIZE))

    };

    pack();
} // </editor-fold>

```

```

private void INSERTAActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    inserta();

    muestra();

    Cantidad.setText("");
}

```

```

private void MODIFICAActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    modifica();

    muestra();

    Cantidad.setText("");
}

```

```

private void ELIMINAActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    elimina();

    muestra();

    Cantidad.setText("");
}

```

```
// Variables declaration - do not modify

private javax.swing.JTextField Cantidad;

private javax.swing.JComboBox Carritos;

private javax.swing.JTable Detalles;

private javax.swing.JButton ELIMINA;

private javax.swing.JButton INSERTA;

private javax.swing.JButton MODIFICA;

private javax.swing.JComboBox Productos;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JLabel subtotal;

// End of variables declaration
}
```

## Clase Detalle\_Orden

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Operaciones;

import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.LinkedList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

/*
 vista donde se muestra los detalles de las ordenes creadas
 */

public class Detalle_Orden extends javax.swing.JFrame {

    private String HOST="localhost";//host
    private String PUERTO="5432";//puerto
    private String DB="ComprasEnLinea";//base de datos
    private String USER="postgres";//usuario
    private String PASS="postgres";// contraseña
    public String url="jdbc:postgresql://" +HOST+": "+PUERTO+"/" +DB; //dirección url
    private Connection conexion=null;// variable para la base de datos

    public String idFuerte;

    public LinkedList<String> idsProductos=new LinkedList<>();
    public LinkedList<String> idsOrdenes=new LinkedList<>();
```

```

public LinkedList<String> NomProductos=new LinkedList<>();

public String user;// usuario

/**
 * Creates new form Detalle_Orden
 */

public Detalle_Orden(String U) {

    initComponents();

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    muestra();

    productos();

    ordenes();

    cierra();

    usuario(U);

    user=U;

}

public void usuario(String u){

    if(u.equals("pablo")){

        inserta.setEnabled(false);

        modifica.setEnabled(false);

        elimina.setEnabled(false);

    }

}

//cierra la vista

public void cierra(){

    this.addWindowListener(new WindowAdapter() {

        public void windowClosing(WindowEvent e) {

            Orden d=new Orden(user);

            d.setVisible(true);

        }

    });

}

//obtiene los productos de la base de datos

public void productos(){

    try{

        conexion=connectaBD();// crea la coenxion a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para obtener los productos de la base de datos

```

```
String Query="SELECT Nombre_Producto,Id_Producto,Precio_Publico_Proveedor FROM Articulo.Producto ORDER BY Id_Producto ASC";
```

```
    corrida=conexion.createStatement();
```

```
    ResultSet columnas=corrida.executeQuery(Query);// ejecutar el query
```

```
    while(columnas.next()){
```

```
        Productos.addItem(columnas.getString(1)+" - "+columnas.getString(3));
```

```
        idsProductos.addLast(columnas.getString(2));
```

```
    }
```

```
    }catch(Exception ex){
```

```
        JOptionPane.showMessageDialog(null, ex);
```

```
    }
```

```
}
```

```
// método que obtiene un producto de la tabla Articulo.Producto
```

```
public String buscaProducto(String cadena){
```

```
    String Query1="";// variable para el query
```

```
    try{
```

```
        conexion=connectaBD();// conexion de la base de datos
```

```
        java.sql.Statement corrida;
```

```
        // query que obtiene los datos del producto segun su nombre
```

```
        Query1="SELECT Id_Producto FROM Articulo.Producto WHERE Nombre_Producto='"+cadena+"'";
```

```
        corrida=conexion.createStatement();
```

```
        ResultSet columnas=corrida.executeQuery(Query1);
```

```
        while(columnas.next()){
```

```
            Query1=columnas.getString(1);
```

```
        }
```

```
        corrida.close();
```

```
        conexion.close();// cierra la conexion
```

```
    }catch(Exception ex){
```

```
        JOptionPane.showMessageDialog(null, ex);
```

```
    }
```

```
    return Query1;
```

```
}
```

```
// obtiene las ordenes en la base de datos
```

```
public void ordenes(){
```

```
    try{
```

```
        conexion=connectaBD();// conexion de la base de datos
```

```
        java.sql.Statement corrida=conexion.createStatement();
```

```
        // query que obtiene las ordenes y los datos del proveedor
```

```

String Query="SELECT O.Num_Orden,O.Id_Proveedor,P.Nombre_Proveedor FROM Operaciones.Orden
O,Usuario.Proveedor P "

    + "WHERE O.Id_Proveedor=P.Id_Proveedor ORDER BY O.Num_Orden ASC";

corrida=conexion.createStatement();

ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

while(columnas.next()){

    Ordenes.addItem(columnas.getString(1) + " " + columnas.getString(3));

    idsOrdenes.addLast(columnas.getString(1));

}

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

//conecta con la base de datos

public Connection conectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

// inserta un nuevo detalle orden

public void inserta(){

    try{

        conexion=conectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para insertar en la tabla Operaciones.Detalle_Orden una nueva orden

        String Query="INSERT INTO Operaciones.Detalle_Orden(Num_Orden,Id_Producto,Cantidad,SubTotal)"

            +

"VALUES("+idsOrdenes.get(Ordenes.getSelectedIndex())+" "+idsProductos.get(Productos.getSelectedIndex())+" "+Cantidad.getText()+",NULL)";

        corrida.executeUpdate(Query);// ejecutar el query

        corrida.close();

        conexion.close();// cierra la conexión

        //JOptionPane.showMessageDialog(null, "INSERTO");

    }catch(Exception ex){

```

```

JOptionPane.showMessageDialog(null,ex);

}

}

//elimina una orden

public void elimina(){

    try{

        conexion=connectaBD();// crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query que elimina una tupla de la tabla Operaciones.Detalle_Orden

        String Query="DELETE FROM Operaciones.Detalle_Orden WHERE
Num_Orden="+idsOrdenes.get(Ordenes.getSelectedIndex())+" AND "

        + "Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+"""

        + " AND Cantidad="+Cantidad.getText()+" AND SubTotal
=""+DetallesOrden.getValueAt(DetallesOrden.getSelectedRow(),2).toString()+""";

        corrida.executeUpdate(Query);// ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexión

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// modificar orden

public void modifica(){

    try{

        conexion=connectaBD();// crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //obtiene los nuevos datos de los textfield

        String i=DetallesOrden.getValueAt(DetallesOrden.getSelectedRow(), 0).toString();

        String Ca=DetallesOrden.getValueAt(DetallesOrden.getSelectedRow(),1).toString();

        String idProducto = buscaProducto(NomProductos.get(DetallesOrden.getSelectedRow()));

        //query que actualiza la informacion en la tabla Operaciones.Detalle_Orden

        String Query="UPDATE Operaciones.Detalle_Orden SET "

        +

        "Num_Orden="+idsOrdenes.get(Ordenes.getSelectedIndex())+"",Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+",
Cantidad="+Cantidad.getText()+"",SubTotal=NULL "

        + " WHERE Num_Orden="+i+" AND Cantidad="+Ca+" AND Id_Producto="+idProducto+"";

        corrida.executeUpdate(Query);// ejecuta el query

        corrida.close();

```

```

        conexion.close();// cierra la conexión
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null,ex);
    }
}

// muestra los detalles de las ordenes
public void muestra(){
    try{
        conexion=connectaBD(); // crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para obtener las ordenes creadas segun el Id del proveedor y se ordenan de forma ascendente
        String Query="SELECT D.Num_Orden,D.Cantidad,D.SubTotal,P.Nombre_Producto,P.Precio_Publico_Proveedor FROM
Articulo.Producto P,Operaciones.Detalle_Orden D "

            + "WHERE D.Id_Producto=P.Id_Producto ORDER BY D.Num_Orden ASC";

        String[]datos =new String[5];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

        DefaultTableModel model=new DefaultTableModel();

        // crea las columnas en el jTable

        model.addColumn("Número De Orden");

        model.addColumn("Cantidad");

        model.addColumn("SubTotal");

        model.addColumn("Nombre Producto -Precio Proveedor Producto");

        DetallesOrden.setModel(model);

        //llena las columnas con los datos

        while(columnas.next()){

            datos[0]=columnas.getString(1);

            datos[1]=columnas.getString(2);

            datos[2]=columnas.getString(3);

            datos[3]=columnas.getString(4)+" - "+columnas.getString(5);

            NomProductos.add(columnas.getString(4));

            model.addRow(datos);

        }

        // obtiene los datos del renglón al hacer clic dentro del jTable

        DetallesOrden.addMouseListener(new MouseAdapter() {

            public void mousePressed(MouseEvent Mouse_evt){

                JTable table=(JTable) Mouse_evt.getSource();

```



```

        Point point = Mouse_evt.getPoint();

        int row=table.rowAtPoint(point);

        if(Mouse_evt.getClickCount()==1){

            Cantidad.setText(DetallesOrden.getValueAt(DetallesOrden.getSelectedRow(),1).toString());


            String id =buscaProducto(NomProductos.get(DetallesOrden.getSelectedRow()));


            for (int i =0; i < idsProductos.size(); i++){

                if(idsProductos.get(i).equals(id)){

                    Productos.setSelectedIndex(i);

                }

            }


            String valor2 = DetallesOrden.getValueAt(DetallesOrden.getSelectedRow(), 0).toString();

            for (int i =0; i < idsOrdenes.size(); i++){

                if(idsOrdenes.get(i).equals(valor2)){

                    Ordenes.setSelectedIndex(i);

                }

            }

        }

    }

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

/**

 * This method is called from within the constructor to initialize the form.

 * WARNING: Do NOT modify this code. The content of this method is always

 * regenerated by the Form Editor.

 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();

```

```

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

DetallesOrden = new javax.swing.JTable();

inserta = new javax.swing.JButton();

elimina = new javax.swing.JButton();

modifica = new javax.swing.JButton();

Productos = new javax.swing.JComboBox();

Ordenes = new javax.swing.JComboBox();

Cantidad = new javax.swing.JTextField();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Detalle Orden");


jLabel1.setFont(new java.awt.Font("sansserif", 3, 14)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 0, 51));
jLabel1.setText("Detalle Orden");


jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel2.setText("Producto");


jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel3.setText("Orden");


jLabel4.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel4.setText("Cantidad");


DetallesOrden.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));

jScrollPane1.setViewportViewView(DetallesOrden);

```



```

        .addComponent(jLabel4)

        .addComponent(jLabel3))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addGroup(layout.createSequentialGroup()

            .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, 153,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 68, Short.MAX_VALUE)

            .addComponent(elimina, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGroup(layout.createSequentialGroup()

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                    .addComponent(Productos, 0, 203, Short.MAX_VALUE)

                    .addComponent(Ordenes, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                .addGap(18, 18, 18)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                    .addComponent(inserta, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
                        javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(modifica))))))

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 94, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel2)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1154,
            javax.swing.GroupLayout.PREFERRED_SIZE))

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jLabel1)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jLabel2)

                    .addComponent(Productos, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(inserta))

                .addGap(10, 10, 10)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jLabel3)

                    .addComponent(Ordenes, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(modifica))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(elimina))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 206,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void OrdenesActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void modificaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    modifica();

    NomProductos.clear();

    muestra();

    Cantidad.setText("");

}

```

```

private void eliminaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    elimina();

    NomProductos.clear();

    muestra();

    Cantidad.setText("");

}

```

```

private void insertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();
}

```

```
NomProductos.clear();  
  
muestra();  
  
Cantidad.setText("");  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JTextField Cantidad;  
  
private javax.swing.JTable DetallesOrden;  
  
private javax.swing.JComboBox Ordenes;  
  
private javax.swing.JComboBox Productos;  
  
private javax.swing.JButton elimina;  
  
private javax.swing.JButton inserta;  
  
private javax.swing.JLabel jLabel1;  
  
private javax.swing.JLabel jLabel2;  
  
private javax.swing.JLabel jLabel3;  
  
private javax.swing.JLabel jLabel4;  
  
private javax.swing.JScrollPane jScrollPane1;  
  
private javax.swing.JButton modifica;  
  
// End of variables declaration  
}
```

## Clase Devolucion\_Carrito

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Operaciones;

import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

/*
 vista donde se muestran las devoluciones de cada carrito y los productos a devolver
 */

public class Devolucion_Carrito extends javax.swing.JFrame {

    private String HOST="localhost";//host

    private String PUERTO="5432";//puerto

    private String DB="ComprasEnLinea";//base de datos

    private String USER="postgres";//usuario
```

```

private String PASS="postgres";// contraseña

public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url

private Connection conexion=null;// variable para la base de datos

public LinkedList<String> idsProductos=new LinkedList<>();

public LinkedList<String> idsCarritos=new LinkedList<>();

public String user;// usuario

/**
 * Creates new form Devolucion_Carrito
 */
public Devolucion_Carrito(String U) {

    initComponents();

    muestra();

    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    cierra();

    fecha();

    carritos();

    productos();

    usuario(U);

    user=U;
}

public void usuario(String u){

    if(u.equals("pablo")){

        INSERTA.setEnabled(false);

        MODIFICA.setEnabled(false);

        ELIMINA.setEnabled(false);

    }

}

// da formato a la fecha

public void fecha() {

    Calendar fechaActual = Calendar.getInstance();

    // Formatear la fecha actual a un formato de String

    SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");

    String fechaFormateada = formatoFecha.format(fechaActual.getTime());

    fecha.setText(fechaFormateada);
}

// conexión con la base de datos

```



```

public Connection conectaBD(){
    Connection cone=null;

    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }

    return cone;
}

// cierra la vista
public void cierra(){
    this.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            Carrito_Venta d=new Carrito_Venta(user);
            d.setVisible(true);
        }
    });
}

//obtiene los carritos de la base de datos
public void carritos(){
    try{
        conexion=conectaBD();// crea la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query que obtiene el nombre del cliente y el carrito relacionado al cliente
        String Query="SELECT C.Nombre_Cliente,O.Id_Carrito \n" +
            "FROM Operaciones.Carrito_Venta O,Usuario.Cliente C,Datos.Tarjeta_Cliente T \n" +
            "WHERE C.Id_Cliente=T.Id_Cliente AND T.Id_Tarjeta=O.Id_Tarjeta_Cliente AND O.Total_Carrito> '0'::money\n" +
            "ORDER BY O.Id_Carrito ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        while(columnas.next()){
            Carritos.addItem(columnas.getString(2)+" "+columnas.getString(1));

            idsCarritos.addLast(columnas.getString(2));
        }
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }
}

```

```

    }

    //obtiene los productos de la base de datos
public void productos(){

    Carritos.addActionListener(new ActionListener() {

        @Override

        public void actionPerformed(ActionEvent e) {

            try{

                conexion=connectaBD();// crea la coenxion a la base de datos

                java.sql.Statement corrida=conexion.createStatement();

                Productos.removeAllItems();

                // obtiene los productos relacionados al carrito

                String Query="SELECT P.Nombre_Producto,P.Id_Producto,P.Precio_Publico_Producto FROM Artículo.Producto
P,Operaciones.Detalle_Carrito D"

                + " WHERE D.Id_Carrito="+idsCarritos.get(Carritos.getSelectedIndex())+" AND "

                + "P.Id_Producto=D.Id_Producto AND D.Cantidad>0 ORDER BY P.Id_Producto ASC";

                corrida=conexion.createStatement();

                ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

                while(columnas.next()){

                    Productos.addItem(columnas.getString(1)+" - "+columnas.getString(3));

                    idsProductos.addLast(columnas.getString(2));

                }

            }catch(Exception ex){

                JOptionPane.showMessageDialog(null, ex);

            }

        }

    });

}

//muestra segun el carrito los productos a devolver
public void muestra(){

    try{

        conexion=connectaBD(); // crea la conexión a la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query que obtiene el carrito y los datos del prodcuto relacionado al carrito

        String Query="SELECT O.Id_Carrito,O.Cantidad_Producto,O.Fecha,P.Nombre_Producto,P.Precio_Publico_Producto\n" +

        "FROM Operaciones.Devolucion_Carrito O,Artículo.Producto P\n" +

        "WHERE O.Id_Producto=P.Id_Producto";

```

```

String[] datos = new String[7];

corrida = conexion.createStatement();

ResultSet columnas = corrida.executeQuery(Query); //ejecuta el query

DefaultTableModel model = new DefaultTableModel();

//agrega las columnas al jTable

model.addColumn("Id Carrito");

model.addColumn("Cantidad");

model.addColumn("Fecha");

model.addColumn("Nombre Producto - Precio Público Producto");

// ingresa los datos al jTable

Devoluciones.setModel(model);

while(columnas.next()){

    datos[0] = columnas.getString(1);

    datos[1] = columnas.getString(2);

    datos[2] = columnas.getString(3);

    datos[3] = columnas.getString(4) + " - " + columnas.getString(5);

    model.addRow(datos);

}

// obtiene los datos del renglón al hacer clic dentro del jTable

Devoluciones.addMouseListener(new MouseAdapter() {

    public void mousePressed(MouseEvent Mouse_evt){

        JTable table = (JTable) Mouse_evt.getSource();

        Point point = Mouse_evt.getPoint();

        int row = table.rowAtPoint(point);

        if(Mouse_evt.getClickCount() == 1){

            for (int i = 0; i < idsCarritos.size(); i++){

                if(idsCarritos.get(i).equals((Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 0).toString()))){

                    Carritos.setSelectedIndex(i);

                }

            }

            String valor = Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 3).toString();

            String id = buscaProducto(valor);

            for (int i = 0; i < idsProductos.size(); i++){

                if(idsProductos.get(i).equals(id)){

                    Productos.setSelectedIndex(i);

                }

            }

        }

    }

}

```

```

try{

    conexion=connectaBD(); // inicia conexion de la base de datos

    java.sql.Statement corrida;

    // obtiene la fecha de la devolucion segun el carrito

    String Query="SELECT to_char(Fecha,'yyyy/MM/dd') FROM Operaciones.Devolucion_Carrito WHERE
Id_Carrito="+((Devoluciones.getValueAt(Devoluciones.getSelectedRow(),0).toString()))+""";

    corrida=conexion.createStatement();

    ResultSet columnas=corrida.executeQuery(Query); // ejecuta el query

    while(columnas.next()){

        fecha.setText(columnas.getString(1));

    }

    corrida.close();

    conexion.close(); // cierra la conexion

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

Cantidad.setText(((Devoluciones.getValueAt(Devoluciones.getSelectedRow(),1).toString())));

}

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

// busca el producto en la base de datos, recibiendo el nombre como parametro

public String buscaProducto(String cadena){

    String Query1=""; //variable para el query

    try{

        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida;

        //query para obtener el producto con el nombre rebicido

        Query1="SELECT Id_Producto FROM Artículo.Producto WHERE Nombre_Producto='"+cadena+"'";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query1); //ejecuta el query

        while(columnas.next()){

            Query1=columnas.getString(1);

```

```

    }

    corrida.close();

    conexion.close(); //cierra la conexion

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, ex);

}

return Query1;

}

// modifica la devolucion

public void modifica() {

    try {

        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // nuevos datos para la modificación

        String i=Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 0).toString();

        String Ca=Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 1).toString();

        String valor = Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 3).toString();

        String idProducto = buscaProducto(valor);

        String FechaBusca = Devoluciones.getValueAt(Devoluciones.getSelectedRow(), 2).toString();

        //query para actualizar la tupla de la tabla Operaciones.Devolucion_Carrito

        String Query="UPDATE Operaciones.Devolucion_Carrito SET "

            +

            "Id_Carrito="+idsCarritos.get(Carritos.getSelectedIndex())+" ,Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+" ,Can

            tidad_Producto="+Cantidad.getText()+" ,fecha='"+ fecha.getText()+" "

            + " WHERE Id_Carrito="+i+" AND Cantidad_Producto="+Ca+" AND Id_Producto="+idProducto+" AND

            Fecha='"+FechaBusca+"'";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, ex);

    }

}

// inserta una nueva devolución

public void inserta() {

    try {

        conexion=connectaBD(); // crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para crear una nueva devolucion en la tabla Operaciones.Devolucion_Carrito

```

```

        String Query="INSERT INTO Operaciones.Devolucion_Carrito(Id_Carrito,Fecha,Id_Producto,Cantidad_Producto) "
        +
        "VALUES("+idsCarritos.get(Carritos.getSelectedIndex())+","+fecha.getText()+","+idsProductos.get(Productos.getSelectedIndex())+
        ", "+Cantidad.getText()+")";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null,"No tienes tantos productos");

    }

}

//elimina una devolucion

public void elimina(){

    try{

        conexion=connectaBD(); // conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // elimina de la tabla Operaciones.Devolucion_Carrito, la devolucion seleccionada

        String Query="DELETE FROM Operaciones.Devolucion_Carrito WHERE
        Id_Carrito="+idsCarritos.get(Carritos.getSelectedIndex())+" AND "

        + "Id_Producto="+idsProductos.get(Productos.getSelectedIndex())+" "

        + "AND Cantidad_Producto="+Cantidad.getText()+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

/**

 * This method is called from within the constructor to initialize the form.

 * WARNING: Do NOT modify this code. The content of this method is always

 * regenerated by the Form Editor.

 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel5 = new javax.swing.JLabel();

    jLabel1 = new javax.swing.JLabel();

```

```
jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

Carritos = new javax.swing.JComboBox();

Productos = new javax.swing.JComboBox();

Cantidad = new javax.swing.JTextField();

MODIFICA = new javax.swing.JButton();

ELIMINA = new javax.swing.JButton();

INSERTA = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

Devoluciones = new javax.swing.JTable();

fecha = new javax.swing.JLabel();


jLabel5.setText("jLabel5");


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Devolución Carrito");


jLabel1.setFont(new java.awt.Font("sansserif", 3, 14)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 51, 255));
jLabel1.setText("Devolución Carrito");


jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel2.setText("Carrito");


jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel3.setText("Producto");


jLabel4.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel4.setText("Cantidad");


jLabel6.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
jLabel6.setText("Fecha");


Carritos.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CarritosActionPerformed(evt);
    }
});
```

```

    }
});

MODIFICA.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
MODIFICA.setText("MODIFICA");
MODIFICA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        MODIFICAActionPerformed(evt);
    }
});

ELIMINA.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
ELIMINA.setText("ELIMINA");
ELIMINA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ELIMINAActionPerformed(evt);
    }
});

INSERTA.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
INSERTA.setText("INSERTA");
INSERTA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        INSERTAActionPerformed(evt);
    }
});

Devoluciones.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));

jScrollPane1.setViewportViewView(Devoluciones);

fecha.setFont(new java.awt.Font("sansserif", 0, 14)); // NOI18N

```



```

fecha.setText("Fecha");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addGroup(layout.createSequentialGroup()
                    .addGap(19, 19, 19)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel3)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(Productos, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel4)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
                                javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabel6)
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                            .addComponent(jLabel2))
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(layout.createSequentialGroup()
                                .addGap(24, 24, 24)
                                .addComponent(fecha))
                            .addGroup(layout.createSequentialGroup()
                                .addGap(25, 25, 25)
                                .addComponent(Carritos, javax.swing.GroupLayout.PREFERRED_SIZE, 202,
                                    javax.swing.GroupLayout.PREFERRED_SIZE))))
                    .addGap(24, 24, 24)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(INSERTA, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                            Short.MAX_VALUE)
                        .addComponent(MODIFICA, javax.swing.GroupLayout.DEFAULT_SIZE, 119, Short.MAX_VALUE)

```

```

        .addComponent(ELIMINA, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))

        .addGap(0, 414, Short.MAX_VALUE))

    .addGroup(layout.createSequentialGroup())

    .addContainerGap()

    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 845, Short.MAX_VALUE)

    .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addComponent(jLabel1)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel2)

    .addComponent(Carritos, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel3)

    .addComponent(Productos, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(MODIFICA))

    .addGap(8, 8, 8)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel4)

    .addComponent(Cantidad, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(ELIMINA))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel6)

    .addComponent(fecha)))

    .addComponent(INSERTA))

    .addGap(10, 10, 10)

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 309,
javax.swing.GroupLayout.PREFERRED_SIZE))

);

```

```
    pack();  
} // </editor-fold>
```

```
private void INSERTAActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    inserta();  
    muestra();  
    Productos.removeAllItems();  
}
```

```
private void MODIFICAActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    modifica();  
    muestra();  
    Productos.removeAllItems();  
  
}
```

```
private void ELIMINAActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    elimina();  
    muestra();  
    Productos.removeAllItems();  
}
```

```
private void CarritosActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JTextField Cantidad;  
private javax.swing.JComboBox Carritos;  
private javax.swing.JTable Devoluciones;  
private javax.swing.JButton ELIMINA;  
private javax.swing.JButton INSERTA;
```

```
private javax.swing.JButton MODIFICA;  
private javax.swing.JComboBox Productos;  
private javax.swing.JLabel fecha;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JScrollPane jScrollPane1;  
  
// End of variables declaration  
}
```

## Clase Orden

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package Operaciones;  
  
import Usuario.Cliente;  
  
import java.sql.Connection;  
  
import java.awt.Point;  
  
import java.awt.event.MouseAdapter;  
  
import java.awt.event.MouseEvent;  
  
import java.sql.Connection;  
  
import java.sql.DriverManager;  
  
import java.sql.ResultSet;  
  
import java.sql.SQLException;  
  
import java.util.LinkedList;  
  
import java.util.logging.Level;  
  
import java.util.logging.Logger;  
  
import javax.swing.ComboBoxModel;  
  
import javax.swing.JComboBox;  
  
import javax.swing.JOptionPane;  
  
import javax.swing.JTable;  
  
import javax.swing.JTextField;  
  
import javax.swing.table.DefaultTableModel;  
  
import Menu.Menu;  
  
import java.util.Calendar;  
  
import java.util.regex.Matcher;  
  
import java.util.regex.Pattern;  
  
import javax.swing.JFrame;
```

```
/*  
 vista donde se muestran las ordenes creadas  
 */  
  
public class Orden extends javax.swing.JFrame {  
  
    private String HOST="localhost"; //host  
  
    private String PUERTO="5432"; //puerto
```

```

private String DB="ComprasEnLinea";//base de datos

private String USER="postgres";//usuario

private String PASS="postgres";// contraseña

public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url

private Connection conexion=null;// variable para la base de datos

public String id; // id de la orden

public String numeros; // numero de la orden

public LinkedList<String> ids=new LinkedList<>();

public LinkedList<String> fechas=new LinkedList<>();

public LinkedList<String> correos=new LinkedList<>();

public String user; // usuario

/**
 * Creates new form Orden
 */
public Orden(String U) {
    initComponents();
    proveedor();
    muestra();
    fecha();
    usuario(U);
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    user=U;
}

public void usuario(String u){
    if(u.equals("pablo")){
        inserta.setEnabled(false);
        modifica.setEnabled(false);
        elimina.setEnabled(false);
    }
}

// da formato y muestra el calendario en la vista
public void fecha(){
    Calendar fechaActual = Calendar.getInstance();
    fecha.setSelectableDateRange(fechaActual.getTime(),null);
}

// obtiene los proveedores
public void proveedor(){
    try{

```

```

        conexion=connectaBD(); // crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // obitene los datos de los proveedores

        String Query="SELECT Nombre_Proveedor,Id_Proveedor,Correo_Electronico FROM Usuario.Proveedor ORDER BY
Id_Proveedor ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        //llena los campos con la informacion del proveedor

        while(columnas.next()){

            proveedores.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            ids.addLast(columnas.getString(2));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//conecta con la base de datos

public Connection connectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

// muestra las ordenes

public void muestra(){

    try{

        conexion=connectaBD(); // crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // muestra los datos de la orden y del proveedor

        String Query="SELECT D.Num_Orden,D.Fecha_Orden,D.Total,P.Nombre_Proveedor,P.Correo_Electronico FROM
Operaciones.Orden D,Usuario.Proveedor P WHERE D.Id_Proveedor=P.Id_Proveedor ORDER BY D.Num_Orden ASC";

        String[]datos =new String[5];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

```

```

DefaultTableModel model=new DefaultTableModel();

// crea las columnas del jTable

model.addColumn("Número De Orden");

model.addColumn("Fecha De Orden");

model.addColumn("Total");

model.addColumn("Nombre Proveedor - Correo Electrónico");

//llena las columnas con los datos de la orden

Ordenes.setModel(model);

while(columnas.next()){

    datos[0]=columnas.getString(1);

    datos[1]=columnas.getString(2);

    datos[2]=columnas.getString(3);

    datos[3]=columnas.getString(4)+" - "+columnas.getString(5);

    correos.add(columnas.getString(5));

    model.addRow(datos);

}

// obtiene los datos del renglón al hacer clic dentro del jTable

Ordenes.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent Mouse_evt){

    JTable table=(JTable) Mouse_evt.getSource();

    Point point = Mouse_evt.getPoint();

    int row=table.rowAtPoint(point);

    if(Mouse_evt.getClickCount()==1){

        // busqueda de los correos

        id=busqueda(correos.get(Ordenes.getSelectedRow()));

        for (int i =0; i < ids.size(); i++){

            if(ids.get(i).equals(id)){

                proveedores.setSelectedIndex(i);

            }

        }

        // id de las ordenes

        id=Ordenes.getValueAt(Ordenes.getSelectedRow(), 0).toString();

        total.setText(Ordenes.getValueAt(Ordenes.getSelectedRow(), 2).toString());

        try{

            conexion=connectaBD(); // crea una conexion con la base de datos

            java.sql.Statement corrida;

            // obitene la fecha de la creacion de la orden

            String Query="SELECT to_char(Fecha_Orden,'yyyy/MM/dd') FROM Operaciones.Orden WHERE Num_Orden='"+id+"'";

```



```

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        while(columnas.next()){

            ((JTextField)fecha.getDateEditor().getUiComponent()).setText(columnas.getString(1));

        }

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

// busca el proveedor segun el correo que recibe como parametro
public String busqueda(String cadena){

    String Query1="";

    try{

        conexion=connectaBD(); // conexion a la base de datos

        java.sql.Statement corrida;

        // obitene el proveedor relacionado al correo

        Query1="SELECT Id_Proveedor FROM Usuario.Proveedor WHERE Correo_Electronico='"+cadena+"'";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query1); //ejecuta el query

        while(columnas.next()){

            Query1=columnas.getString(1);

        }

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return Query1;

}

```

```

// inserta una nueva orden

public void inserta(){

    try{

        conexion=connectaBD();// crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        Pattern patron = Pattern.compile("\\d+");

        Matcher matcher = patron.matcher(total.getText());

        String num="";

        while (matcher.find()) {

            num = matcher.group();

        }

        //inserta nueva orden en la tabla Operaciones.Orden

        String Query="INSERT INTO Operaciones.Orden(Id_Proveedor,Fecha_Orden,Total)"

            +
            "VALUES('"+ids.get(provedores.getSelectedIndex())+"', '"+(((JTextField)fecha.getDateEditor().getUiComponent()).getText())+"', "+nu
            m+"')";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//elimina una orden

public void elimina(){

    try{

        conexion=connectaBD();// crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query que elimina una orden seleccionada

        String Query="DELETE FROM Operaciones.Orden WHERE Num_Orden="+Ordenes.getValueAt(Ordenes.getSelectedRow(),
        0).toString()+" ";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//modifica la tupla en la base de datos

```

```

public void modifica(){

    try{

        conexion=connectaBD(); // crea la conexión con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query que actualiza la orden con los nuevos datos de los textfield

        String Query="UPDATE Operaciones.Orden SET "

            + "Id_Proveedor= "+ids.get(provedores.getSelectedIndex())+" ,
Fecha_Orden="+"(((JTextField)fecha.getDateEditor().getUiComponent()).getText())+"", Total=
"+Ordenes.getValueAt(Ordenes.getSelectedRow(), 2).toString()+" "

            + "WHERE Num_Orden="+Ordenes.getValueAt(Ordenes.getSelectedRow(), 0).toString()+"";

        corrida.executeUpdate(Query); // ejecuta el query

        corrida.close();

        conexion.close();// cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    inserta = new javax.swing.JButton();

    elimina = new javax.swing.JButton();

    modifica = new javax.swing.JButton();

    Detalle = new javax.swing.JButton();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jLabel4 = new javax.swing.JLabel();

    fecha = new com.toedter.calendar.JDateChooser();

    total = new javax.swing.JLabel();

    proveedores = new javax.swing.JComboBox();

    jScrollPane1 = new javax.swing.JScrollPane();

    Ordenes = new javax.swing.JTable();

```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);  
setTitle("Orden");
```

```
inserta.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N  
inserta.setText("INSERTA");  
inserta.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        insertaActionPerformed(evt);  
    }  
});
```

```
elimina.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N  
elimina.setText("ELIMINA");  
elimina.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        eliminaActionPerformed(evt);  
    }  
});
```

```
modifica.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N  
modifica.setText("MODIFICA");  
modifica.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        modificaActionPerformed(evt);  
    }  
});
```

```
Detalle.setFont(new java.awt.Font("sansserif", 2, 14)); // NOI18N  
Detalle.setForeground(new java.awt.Color(255, 102, 0));  
Detalle.setText("DETALLE ORDEN");  
Detalle.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        DetalleActionPerformed(evt);  
    }  
});
```

```
jLabel1.setFont(new java.awt.Font("sansserif", 3, 14)); // NOI18N
```

```
jLabel1.setText("Orden");
```

```
jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
```

```
jLabel2.setText("Fecha De Orden");
```

```
jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
```

```
jLabel3.setText("Total");
```

```
jLabel4.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
```

```
jLabel4.setText("Proveedor");
```

```
total.setFont(new java.awt.Font("sansserif", 1, 18)); // NOI18N
```

```
total.setForeground(new java.awt.Color(255, 0, 0));
```

```
total.setText("$0.00");
```

```
Ordenes.setModel(new javax.swing.table.DefaultTableModel(
```

```
new Object [][] {
```

 $\},$ 

```
new String [] {
```

"Title 1", "Title 2", "Title 3", "Title 4"

}

 $\rangle\rangle;$ 

```
jScrollPane1.setViewportView(Ordenes);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(layout.createSequentialGroup()
```

```
.addContainerGap()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 996,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addGroup(layout.createSequentialGroup())
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(jLabel1)
```

```
.addGroup(layout.createSequentialGroup())
```

```

        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 122,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(17, 17, 17)

        .addComponent(total))

    .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(fecha, javax.swing.GroupLayout.PREFERRED_SIZE, 163,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel4)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(provedores, javax.swing.GroupLayout.PREFERRED_SIZE, 259,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(inserta, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(layout.createSequentialGroup())

            .addComponent(modifica)

            .addGap(18, 18, 18)

            .addComponent(Detalle, javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(elimina, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))))

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup())

                .addContainerGap()

                .addComponent(inserta)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addComponent(elimina)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(modifica)

                    .addComponent(Detalle)))

```

```

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jLabel2)

            .addComponent(fecha, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(total, javax.swing.GroupLayout.Alignment.TRAILING)

            .addComponent(jLabel3))

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(provedores, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 315,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(38, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void insertaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();

    correos.clear();

    muestra();

    total.setText("$0.00");

}

```

```

private void DetalleActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    Detalle_Orden d=new Detalle_Orden(user);

    d.setVisible(true);

    this.setVisible(false);

}

```

```
private void eliminaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    elimina();  
  
    correos.clear();  
  
    muestra();  
  
    total.setText("$0.00");  
}
```

```
private void modificaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    modifica();  
  
    correos.clear();  
  
    muestra();  
  
    total.setText("$0.00");  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JButton Detalle;  
  
private javax.swing.JTable Ordenes;  
  
private javax.swing.JButton elimina;  
  
private com.toedter.calendar.JDateChooser fecha;  
  
private javax.swing.JButton inserta;  
  
private javax.swing.JLabel jLabel1;  
  
private javax.swing.JLabel jLabel2;  
  
private javax.swing.JLabel jLabel3;  
  
private javax.swing.JLabel jLabel4;  
  
private javax.swing.JScrollPane jScrollPane1;  
  
private javax.swing.JButton modifica;  
  
private javax.swing.JComboBox provedores;  
  
private javax.swing.JLabel total;  
  
// End of variables declaration  
}
```



## Clase Reporte1

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Reportes;

import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.LinkedList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/*
 vista de los reportes de los proveedores, mostrando las ordenes del proveedor
 */

public class Reporte1 extends javax.swing.JFrame {

    private String HOST="localhost"; //host
    private String PUERTO="5432"; //puerto
    private String DB="ComprasEnLinea"; //base de datos
    private String USER="postgres"; //usuario
    private String PASS="postgres"; // contraseña
    public String url="jdbc:postgresql://" + HOST + ":" + PUERTO + "/" + DB; //dirección url
    private Connection conexion=null; // variable para la base de datos
    public LinkedList<String> ids=new LinkedList<>();

    /**
     * Creates new form Reporte1
     */
    public Reporte1() {
        initComponents();
    }
}
```

```

proveedor();

this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

DefaultTableModel model=new DefaultTableModel();

    model.addColumn("Número De Ordenes");

    model.addColumn("Proveedor");

    carritos.setModel(model);

}

// obtiene los proveedores

public void proveedor(){

    try{

        conexion=connectaBD();// crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // obitene los datos de los proveedores

        String Query="SELECT Nombre_Proveedor,Id_Proveedor,Correo_Electronico FROM Usuario.Proveedor ORDER BY
Id_Proveedor ASC";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query);//ejecuta el query

        while(columnas.next()){

            //llena los campos con la informacion del proveedor

            proveedor.addItem(columnas.getString(1)+" - "+columnas.getString(3));

            ids.addLast(columnas.getString(2));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

// realiza la conexion con la base de datos

public Connection connectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

    return cone;

}

```

```

// muestra los reportes

public void muestra(){

    try{

        conexion=connectaBD();// crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // muestra cuantas ordenes tiene el proveedor asi como su nombre

        String Query="SELECT COUNT(O.Num_Orden), P.Nombre_Proveedor\n" +

            "FROM Operaciones.Orden O\n" +

            "INNER JOIN Usuario.Proveedor P ON O.Id_Proveedor = P.Id_Proveedor AND P.Id_Proveedor

            =" +ids.get(proveedor.getSelectedIndex())+" \n" +

            "GROUP BY P.Nombre_Proveedor;";

        String[]datos =new String[5];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        DefaultTableModel model=new DefaultTableModel();

        // crea las columnas del jtable

        model.addColumn("Número De Órdenes");

        model.addColumn("Proveedor");

        carritos.setModel(model);

        //agrega la informacion al jtable

        while(columnas.next()){

            datos[0]=columnas.getString(1);

            datos[1]=columnas.getString(2);

            model.addRow(datos);

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

```

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

private void initComponents() {

    proveedor = new javax.swing.JComboBox();

    jLabel2 = new javax.swing.JLabel();

    jButton1 = new javax.swing.JButton();

    jScrollPane1 = new javax.swing.JScrollPane();

    carritos = new javax.swing.JTable();

    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Reporte 1");

    jLabel2.setFont(new java.awt.Font("sansserif", 2, 14)); // NOI18N
    jLabel2.setText("Muestra el número de órdenes que se le han pedido a un proveedor, seleccionando al proveedor");

    jButton1.setText("Buscar");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    carritos.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Title 1", "Title 2", "Title 3", "Title 4"
        }
    ));

    jScrollPane1.setViewportViewView(carritos);

    jLabel1.setFont(new java.awt.Font("Arial Rounded MT Bold", 1, 12)); // NOI18N
    jLabel1.setText("Seleccione un Proveedor.");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 631, Short.MAX_VALUE)

        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 631, Short.MAX_VALUE)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addContainerGap()

        .addComponent(proveedor, javax.swing.GroupLayout.PREFERRED_SIZE, 359,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jButton1))

        .addComponent(jLabel1))

        .addGap(0, 0, Short.MAX_VALUE)))

        .addContainerGap())
    );

    layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 50, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(8, 8, 8)

    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 21, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(proveedor, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addComponent(jButton1))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 219,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(0, 20, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
}

```

```
muestra();  
}
```

```
// Variables declaration - do not modify  
  
private javax.swing.JTable carritos;  
private javax.swing.JButton jButton1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JComboBox proveedor;  
  
// End of variables declaration  
}
```

## Clase Reporte2

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Reportes;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.util.LinkedList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 vista de los reportes de los clientes mostrando sus telefonos
 */

public class Reporte2 extends javax.swing.JFrame {

    private String HOST="localhost"; //host

    private String PUERTO="5432"; //puerto

    private String DB="ComprasEnLinea"; //base de datos

    private String USER="postgres"; //usuario

    private String PASS="postgres"; // contraseña

    public String url="jdbc:postgresql://" + HOST + ":" + PUERTO + "/" + DB; //dirección url

    private Connection conexion=null; // variable para la base de datos

    public LinkedList<String> ids=new LinkedList<>();

    /**
     * Creates new form Reporte2
     */
    public Reporte2() {
        initComponents();

        clientes();

        DefaultTableModel model=new DefaultTableModel();

        model.addColumn("Número ");

        carritos.setModel(model);
    }
}
```

```

        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }

    // obtiene los clientes
    public void clientes(){
        try{
            conexion=connectaBD();// crea la conexion con la base de datos

            java.sql.Statement corrida=conexion.createStatement();

            // query que obtiene los datos de los clientes y los ordena de forma ascendente
            String Query="SELECT Nombre_Cliente,Id_Cliente,Correo_Electronico FROM Usuario.Cliente ORDER BY Id_Cliente ASC";

            corrida=conexion.createStatement();

            ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

            // llena los campos con la información
            while(columnas.next()){

                Clientes.addItem(columnas.getString(1)+" - "+columnas.getString(3));

                ids.addLast(columnas.getString(2));

            }

        }catch(Exception ex){

            JOptionPane.showMessageDialog(null, ex);

        }

    }

    //crea la conexión con la base de datos
    public Connection connectaBD(){

        Connection cone=null;

        try{

            Class.forName("org.postgresql.Driver");

            cone=DriverManager.getConnection(url,USER,PASS);

        }catch(Exception ex){

            JOptionPane.showMessageDialog(null, ex);

        }

        return cone;

    }

    // muestra los reportes
    public void muestra(){

        try{

            conexion=connectaBD();// crea la conexion con la base de datos

            java.sql.Statement corrida=conexion.createStatement();

            // query que muestra los telefonos del cliente seleccionado

```



```

String Query="SELECT N.Numero\n" +
"FROM Datos.Telefono_Cliente N\n" +
"WHERE N.Id_Cliente IN (SELECT C.Id_Cliente FROM Usuario.Cliente C)"
        + " AND N.Id_Cliente="+ids.get(Clientes.getSelectedIndex())+";";

String[]datos =new String[5];

corrida=conexion.createStatement();

ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

DefaultTableModel model=new DefaultTableModel();

//agrega la columna al jTable
model.addColumn("Número ");

carritos.setModel(model);

// rellena el jTable con la informacion
while(columnas.next()){

    datos[0]=columnas.getString(1);

    model.addRow(datos);

}

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

```

```

    busca = new javax.swing.JButton();

    Clientes = new javax.swing.JComboBox();

    jScrollPane1 = new javax.swing.JScrollPane();

    carritos = new javax.swing.JTable();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Reporte 2");


busca.setText("Buscar");

busca.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        buscaActionPerformed(evt);

    }

});

carritos.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        },

    new String [] {

        "Title 1", "Title 2", "Title 3", "Title 4"

    }

));

jScrollPane1.setViewportView(carritos);


jLabel1.setFont(new java.awt.Font("sansserif", 2, 14)); // NOI18N
jLabel1.setText("Muestra los números de teléfono de un cliente seleccionado");


jLabel2.setFont(new java.awt.Font("Arial Black", 1, 12)); // NOI18N
jLabel2.setText("Seleccione un cliente:");


javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addComponent(jLabel1)

            .addGap(0, 0, Short.MAX_VALUE)

            .addGroup(layout.createSequentialGroup()

                .addComponent(jLabel2)

                .addGap(0, 0, Short.MAX_VALUE)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 566,
javax.swing.GroupLayout.PREFERRED_SIZE)

            )

        )

    )

```

```

        .addGroup(layout.createSequentialGroup())

        .addComponent(Clientes, javax.swing.GroupLayout.PREFERRED_SIZE, 337,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(busca))

        .addComponent(jLabel2))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

        .addContainerGap()

        .addComponent(jLabel1)

        .addGap(33, 33, 33)

        .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(Clientes, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(busca))

            .addGap(18, 18, 18)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 266,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        );

pack();
} // </editor-fold>

```

```

private void buscaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    muestra();

}

```

```

// Variables declaration - do not modify

private javax.swing.JComboBox Clientes;

private javax.swing.JButton busca;

private javax.swing.JTable carritos;

```

```
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JScrollPane jScrollPane1;  
// End of variables declaration  
}
```

## Clase Cmabio\_Usuario

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Usuario;

import Menu.Menu;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import javax.swing.JOptionPane;

/*
 clase para el inicio de sesion
 */

public class Cambio_Usuario extends javax.swing.JFrame {

    private String HOST="localhost"; //host
    private String PUERTO="5432"; //puerto
    private String DB="ComprasEnLinea"; //base de datos
    private String USER="postgres"; //usuario
    private String PASS="postgres"; // contraseña
    private String url="jdbc:postgresql://" + HOST + ":" + PUERTO + "/" + DB; //dirección url
    private Connection conexion=null; // variable para la base de datos
    public String UserNew=""; // usuario
    public String PassNew=""; // contraseña
    /**
     * Creates new form Cambio_Usuario
     */
    public Cambio_Usuario() {
        initComponents();
        usuarios();
    }
    public String User(){
        return UserNew;
    }
}
```

```

// realiza la conexion con la base de datos

public Connection conectaBD(){

    Connection cone=null;

    try{

        Class.forName("org.postgresql.Driver");

        cone=DriverManager.getConnection(url,USER,PASS);

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, "Contraseña Incorrecta");

    }

    return cone;

}

//obtiene los usuarios registrados en la base de datos

public void usuarios(){

    try{

        conexion=conectaBD(); // crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // obtiene todos los usuarios de la base de datos

        String Query="SELECT username \n" +

            "FROM pg_user WHERE username!='postgres'";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        while(columnas.next()){

            //agrega la informacion al campo

            Usuarios.addItem(columnas.getString(1));

        }

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

//valida el usuario y contraseña para iniciar sesion

public void ingresa(){

    try{

        USER=(String) Usuarios.getSelectedItemAt(); //selecciona el usuario

        PASS=Password.getText();

        conexion=conectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // obitene todos los detalles carrito de la base de datos

```

```

        String Query="SELECT * FROM Operaciones.Detalle_Carrito";

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        UserNew=USER;

        PassNew=PASS;

    }catch(Exception ex){

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    Ingresa = new javax.swing.JButton();

    Usuarios = new javax.swing.JComboBox();

    Password = new javax.swing.JPasswordField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Login");

    setAutoRequestFocus(false);

    jLabel1.setFont(new java.awt.Font("sansserif", 0, 24)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 102, 0));

    jLabel2.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
    jLabel2.setText("Usuario");

    jLabel3.setFont(new java.awt.Font("sansserif", 0, 18)); // NOI18N
    jLabel3.setText("Contraseña");

```

[illegible]



```

        .addGap(64, 64, 64)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 43, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(Usuarios, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent>Password, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addComponent(Ingresar)

        .addGap(0, 155, Short.MAX_VALUE))

    );

    pack();
}

```

```

private void IngresaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    ingresa();

    if(UserNew!=""){

        JOptionPane.showMessageDialog(null, "Bienvenido Usuario: "+UserNew);

        Menu c=new Menu(User());

        c.setVisible(true);

        this.dispose();

    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {

```

```

        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Cambio_Usuario.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Cambio_Usuario().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton Ingresa;
private javax.swing.JPasswordField Password;
private javax.swing.JComboBox Usuarios;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;

// End of variables declaration
}

```

## Clase Cliente

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Usuario;

import Datos.Tarjeta_Cliente;
import Datos.Telefono_Cliente;
import Menu.Menu;
import java.sql.Connection;
import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;

/**
 * clase para crear un nuevo cliente
 */
public class Cliente extends javax.swing.JFrame {

    private String HOST="localhost"; //host
    private String PUERTO="5432";//puerto
    private String DB="ComprasEnLinea";//base de datos
    private String USER="postgres";//usuario
    private String PASS="postgres";// contraseña
    public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url
    private Connection conexion=null;// variable para la base de datos
    public String id; // id cliente
    public String user; // usuario
```

```

/**
 * Creates new form Cliente
 */
public Cliente(String U) {
    initComponents();
    Nombre.setText("");
    Correo.setText("");
    muestra();
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    usuario(U);
    user=U;
}

public void usuario(String u){
    if(u.equals("pablo")){
        insertar.setEnabled(false);
        Modificar.setEnabled(false);
        Eliminar.setEnabled(false);
    }
}

//realiza la conexion con la base de datos
public Connection connectaBD(){
    Connection cone=null;

    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }

    return cone;
}

// crea un nuevo cliente
public void inserta(){
    //valida que los campos esten completos
    if(Nombre.getText().equalsIgnoreCase("") || Correo.getText().equalsIgnoreCase("")){
        JOptionPane.showMessageDialog(this, "Casillas vacias");
    }else{
        try{
            conexion=connectaBD(); //crea la conexion con la base de datos

```

```

java.sql.Statement corrida=conexion.createStatement();

// query para insertar un nuevo cliente en la tabla Usuario.Cliente
String Query="INSERT INTO Usuario.Cliente(Nombre_Cliente,Correo_Electronico)"
    + "VALUES('"+Nombre.getText()+"','"+Correo.getText()+"')";

corrida.executeUpdate(Query); //ejecuta el query

corrida.close();

conexion.close();//cierra la conexion

//JOptionPane.showMessageDialog(null, "INSERTO");

}catch(Exception ex){

JOptionPane.showMessageDialog(null, "Tupla Repetida");

}

}

}

//muestra los datos del cliente

public void muestra(){

try{

    conexion=connectaBD(); //crea la conexion con la base de datos

    java.sql.Statement corrida=conexion.createStatement();

    //query para obtener todos los clientes y ordenarlos de forma ascendente
    String Query="SELECT * FROM Usuario.Cliente ORDER BY Id_Cliente ASC";

    String[]datos =new String[5];

    corrida=conexion.createStatement();

    ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

    DefaultTableModel model=new DefaultTableModel();

    //agrega las columnas al jTable

    model.addColumn("Id Cliente");

    model.addColumn("Nombre Cliente");

    model.addColumn("Correo Electrónico");

    //llena las columnas del jTable

    TCliente.setModel(model);

    while(columnas.next()){

        datos[0]=columnas.getString(1);

        datos[1]=columnas.getString(2);

        datos[2]=columnas.getString(3);

        model.addRow(datos);

    }

    // selecciona un cliente y obtiene los datos al hacer clic dentro del jTable

    TCliente.addMouseListener(new MouseAdapter() {

```

```

public void mousePressed(MouseEvent Mouse_evt){

    JTable table=(JTable) Mouse_evt.getSource();

    Point point = Mouse_evt.getPoint();

    int row=table.rowAtPoint(point);

    if(Mouse_evt.getClickCount()==1){

        Nombre.setText(TCliente.getValueAt(TCliente.getSelectedRow(),1).toString());

        Correo.setText(TCliente.getValueAt(TCliente.getSelectedRow(),2).toString());

        id=TCliente.getValueAt(TCliente.getSelectedRow(), 0).toString();

    }

}

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

```

//elimina un cliente

```

public void elimina(){

    try{

        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para borrar un cliente específico

        String Query="DELETE FROM Usuario.Cliente WHERE Id_Cliente="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

```

// acutaliza los datos del cliente

```

public void modifica(){

    try{

        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para actualizar los datos del cliente seleccionado

```

```

        String Query="UPDATE Usuario.Cliente SET
Nombre_Cliente='"+Nombre.getText()+"',Correo_Electronico='"+Correo.getText()+"'"+
        + "WHERE Id_Cliente="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jScrollPane1 = new javax.swing.JScrollPane();

    TCliente = new javax.swing.JTable();

    Nombre = new javax.swing.JTextField();

    Correo = new javax.swing.JTextField();

    insertar = new javax.swing.JButton();

    Tarjeta = new javax.swing.JButton();

    Modificar = new javax.swing.JButton();

    Telefono = new javax.swing.JButton();

    Eliminar = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    setTitle("Cliente");

    jLabel1.setFont(new java.awt.Font("Tw Cen MT", 0, 18)); // NOI18N

    jLabel1.setForeground(new java.awt.Color(0, 204, 204));

    jLabel1.setText("Cliente");

```

```
jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel2.setText("Nombre");
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel3.setText("Correo Electrónico");
```

```
TCliente.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportViewView(TCliente);
```

```
Nombre.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
Correo.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
```

```
insertar.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
insertar.setText("INSERTAR");
insertar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        insertarActionPerformed(evt);
    }
});
```

```
Tarjeta.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Tarjeta.setText("TARJETA");
Tarjeta.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TarjetaActionPerformed(evt);
    }
});
```

```
Modificar.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```





```

        .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(Nombre, javax.swing.GroupLayout.PREFERRED_SIZE, 393,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(layout.createSequentialGroup())

        .addComponent(jLabel3)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(Correo)))

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addComponent(Modificar)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9, Short.MAX_VALUE)

        .addComponent(Telefono))

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addComponent(insertar)

        .addGap(18, 18, 18)

        .addComponent(Tarjeta))

        .addGroup(layout.createSequentialGroup())

        .addGap(6, 6, 6)

        .addComponent(Eliminar)))

        .addGap(0, 0, Short.MAX_VALUE))))))

        .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addContainerGap()

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(insertar)

        .addComponent(Tarjeta))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel2)

```

```

        .addComponent(Nombre, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(Modificar)

        .addComponent(Telefono))

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel3)

                .addComponent(Correo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addGroup(layout.createSequentialGroup()

                .addGap(7, 7, 7)

                .addComponent(Eliminar)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 271,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void TarjetaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    Tarjeta_Cliente Tac=new Tarjeta_Cliente(user);

    Tac.setVisible(true);

    //this.setVisible(false);

}

```

```

private void TelefonoActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    Telefono_Cliente Tc=new Telefono_Cliente(user);

    Tc.setVisible(true);

    //this.setVisible(false);

}

```

```

private void insertarActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

```

```

        inserta();

        muestra();

        Nombre.setText("");

        Correo.setText("");
    }

    private void ModificarActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        modifica();

        muestra();

        Nombre.setText("");

        Correo.setText("");
    }

    private void EliminarActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        elimina();

        muestra();

        Nombre.setText("");

        Correo.setText("");
    }

    /**
     * @param args the command line arguments
     */

    // Variables declaration - do not modify
    private javax.swing.JTextField Correo;
    private javax.swing.JButton Eliminar;
    private javax.swing.JButton Modificar;
    private javax.swing.JTextField Nombre;
    private javax.swing.JTable TCliente;
    private javax.swing.JButton Tarjeta;
    private javax.swing.JButton Telefono;
    private javax.swing.JButton Insertar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;

```

```
private javax.swing.JScrollPane jScrollPane1;  
// End of variables declaration  
}
```

## Clase Proveedor

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package Usuario;

import Datos.Tarjeta_Cliente;
import Datos.Telefono_Cliente;
import Datos.Telefono_Proveedor;
import Menu.Menu;
import java.sql.Connection;
import java.awt.Point;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/**
 * clase de proveedor
 */
public class Proveedor extends javax.swing.JFrame {

    private String HOST="localhost"; //host
    private String PUERTO="5432";//puerto
    private String DB="ComprasEnLinea";//base de datos
    private String USER="postgres";//usuario
    private String PASS="postgres";// contraseña
    public String url="jdbc:postgresql://" +HOST+"."+PUERTO+"/" +DB; //dirección url
    private Connection conexion=null;// variable para la base de datos

    public String id;

    public String user;

    /**
```

```

* Creates new form Proveedor
*/
public Proveedor(String U) {
    initComponents();
    muestra();
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    usuario(U);
    user=U;
}

public void usuario(String u){
    if(u.equals("pablo")){
        Insertar.setEnabled(false);
        Modificar.setEnabled(false);
        Eliminar.setEnabled(false);
    }
}

//realiza la conexion con la base de datos a travez del driver
public Connection connectaBD(){
    Connection cone=null;
    try{
        Class.forName("org.postgresql.Driver");
        cone=DriverManager.getConnection(url,USER,PASS);
    }catch(Exception ex){
        JOptionPane.showMessageDialog(null, ex);
    }
    return cone;
}

// inserta un nuevo proveedor
public void inserta(){
    //valida que los campos de los datos esten completos

    if(Nombre.getText().equalsIgnoreCase("") || Correo.getText().equalsIgnoreCase("") || Domicilio.getText().equalsIgnoreCase("")){
        JOptionPane.showMessageDialog(this, "Casillas vacias");
    }else{
        try{
            conexion=connectaBD(); // crea la conexion con la base de datos
            java.sql.Statement corrida=conexion.createStatement();

            // query para insertar un nuevo proveedor en la tabla Usuario.Proveedor

```

```

String Query="INSERT INTO Usuario.Proveedor(Nombre_Proveedor,Correo_Electronico,Domicilio_Fiscal)"
    + "VALUES('" +Nombre.getText()+"','"+Correo.getText()+"','"+Domicilio.getText()+"')";

corrida.executeUpdate(Query); // ejecuta el query

corrida.close();

conexion.close(); // cierra la conexion

}catch(Exception ex){
OptionPane.showMessageDialog(null,"Tupla Repetida");
}
}
}

//obtiene los datos del proveedor

public void muestra(){
    try{
        conexion=connectaBD(); //crea la conexion con la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        // query para obtener a los proveedores y ordenarlos de forma ascendente

        String Query="SELECT * FROM Usuario.Proveedor ORDER BY Id_Proveedor ASC";

        String[]datos =new String[5];

        corrida=conexion.createStatement();

        ResultSet columnas=corrida.executeQuery(Query); //ejecuta el query

        DefaultTableModel model=new DefaultTableModel();

        //agrega las columnas al jTable

        model.addColumn("Id Proveedor");

        model.addColumn("Correo Electrónico");

        model.addColumn("Nombre Proveedor");

        model.addColumn("Domicilio Fiscal");

        //llena las columnas del jTable

        TProveedor.setModel(model);

        while(columnas.next()){

            datos[0]=columnas.getString(1);

            datos[1]=columnas.getString(2);

            datos[2]=columnas.getString(3);

            datos[3]=columnas.getString(4);

            model.addRow(datos);

        }

        // obtiene los datos del proveedor al hacer clic en el renglon dentro del jTable

        TProveedor.addMouseListener(new MouseAdapter() {

            public void mousePressed(MouseEvent Mouse_evt){

```



```

        JTable table=(JTable) Mouse_evt.getSource();

        Point point = Mouse_evt.getPoint();

        int row=table.rowAtPoint(point);

        if(Mouse_evt.getClickCount()==1){

            Nombre.setText(TProveedor.getValueAt(TProveedor.getSelectedRow(),2).toString());

            Correo.setText(TProveedor.getValueAt(TProveedor.getSelectedRow(),1).toString());

            Domicilio.setText(TProveedor.getValueAt(TProveedor.getSelectedRow(),3).toString());

            id=TProveedor.getValueAt(TProveedor.getSelectedRow(), 0).toString();

        }

    }

});

}catch(Exception ex){

    JOptionPane.showMessageDialog(null, ex);

}

}

//elimina un proveedor

public void elimina(){

    try{

        conexion=connectaBD(); //crea la conexion de la base de datos

        java.sql.Statement corrida=conexion.createStatement();

        //query para eliminar el telefono del proveedor segun su Id

        String Query="DELETE FROM Datos.Telefono_Proveedor WHERE Id_Proveedor="+id+"";

        // query que elimina al proveedor seleccionado

        String Query2="DELETE FROM Usuario.Proveedor WHERE Id_Proveedor="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.executeUpdate(Query2); //ejecuta el query

        corrida.close();

        conexion.close(); //cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

}

//acutaliza los datos del proveedor

public void modifica(){

    try{

        conexion=connectaBD(); // crea la conexion con la base de datos

```

```

        java.sql.Statement corrida=conexion.createStatement();

        // actualiza los datos del proveedor seleccionado

        String Query="UPDATE Usuario.Proveedor SET
Nombre_Proveedor='"+Nombre.getText()+"',Correo_Electronico='"+Correo.getText()+"', "
        + "Domicilio_Fiscal='"+Domicilio.getText()+"'"
        + "WHERE Id_Proveedor="+id+"";

        corrida.executeUpdate(Query); //ejecuta el query

        corrida.close();

        conexion.close(); // cierra la conexion

    }catch(Exception ex){

        JOptionPane.showMessageDialog(null, ex);

    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();

    jTable1 = new javax.swing.JTable();

    jLabel1 = new javax.swing.JLabel();

    jLabel2 = new javax.swing.JLabel();

    jLabel3 = new javax.swing.JLabel();

    jLabel4 = new javax.swing.JLabel();

    Insertar = new javax.swing.JButton();

    Modificar = new javax.swing.JButton();

    Eliminar = new javax.swing.JButton();

    Telefono = new javax.swing.JButton();

    jScrollPane2 = new javax.swing.JScrollPane();

    TProveedor = new javax.swing.JTable();

    Correo = new javax.swing.JTextField();

    Nombre = new javax.swing.JTextField();

    Domicilio = new javax.swing.JTextField();

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Proveedor");

jLabel1.setFont(new java.awt.Font("Tahoma", 2, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 204, 204));
jLabel1.setText("Proveedor");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel2.setText("Correo Electrónico");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel3.setText("Nombre");

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
jLabel4.setText("Domicilio Fiscal");

Insertar.setText("INSERTAR");
Insertar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        InsertarActionPerformed(evt);
    }
});

Modificar.setText("MODIFICAR");
Modificar.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            ModificarActionPerformed(evt);

        }

    });

```

```

Eliminar.setText("ELIMINAR");

Eliminar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        EliminarActionPerformed(evt);

    }

});

```

```

Telefono.setText("TELEFONO");

Telefono.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        TelefonoActionPerformed(evt);

    }

});

```

```

TProveedor.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        },

    new String [] {

        "Title 1", "Title 2", "Title 3", "Title 4"

    }

));

jScrollPane2.setViewportViewView(TProveedor);

```

```

Correo.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

```

```

Nombre.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

```

```

Domicilio.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addComponent(jScrollPane2)

.addContainerGap())

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(Correo, javax.swing.GroupLayout.DEFAULT_SIZE, 484, Short.MAX_VALUE))

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(Nombre))

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(Domicilio))

.addGroup(layout.createSequentialGroup())

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(0, 0, Short.MAX_VALUE)))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(Modificar)

.addComponent(Eliminar, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(Insertar, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(Telefono, javax.swing.GroupLayout.Alignment.LEADING))

.addGap(33, 33, 33)))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup())

```

```

        .addContainerGap()

        .addComponent(jLabel1)

        .addGap(12, 12, 12)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(Correo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(8, 8, 8)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel3)

            .addComponent(Nombre, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(8, 8, 8)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel4)

            .addComponent(Domicilio, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGroup(layout.createSequentialGroup()

            .addComponent(Insertar)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(Modificar)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(Eliminar)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

            .addComponent(Telefono)))

        .addGap(35, 35, 35)

        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 211,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(33, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void InsertarActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    inserta();

    muestra();

    Nombre.setText("");
}

```

```

        Correo.setText("");

        Domicilio.setText("");
    }

    private void ModificarActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        modifica();

        muestra();

        Nombre.setText("");

        Correo.setText("");

        Domicilio.setText("");
    }

    private void EliminarActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        elimina();

        muestra();

        Nombre.setText("");

        Correo.setText("");

        Domicilio.setText("");
    }

    private void TelefonoActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

        Telefono_Proveedor Tp=new Telefono_Proveedor(user);

        Tp.setVisible(true);

        //this.setVisible(false);
    }

    // Variables declaration - do not modify

    private javax.swing.JTextField Correo;

    private javax.swing.JTextField Domicilio;

    private javax.swing.JButton Eliminar;

    private javax.swing.JButton Insertar;

    private javax.swing.JButton Modificar;

    private javax.swing.JTextField Nombre;

    private javax.swing.JTable TProveedor;

```

```
private javax.swing.JButton Telefono;  
  
private javax.swing.JLabel jLabel1;  
  
private javax.swing.JLabel jLabel2;  
  
private javax.swing.JLabel jLabel3;  
  
private javax.swing.JLabel jLabel4;  
  
private javax.swing.JScrollPane jScrollPane1;  
  
private javax.swing.JScrollPane jScrollPane2;  
  
private javax.swing.JTable jTable1;  
  
// End of variables declaration  
}
```