

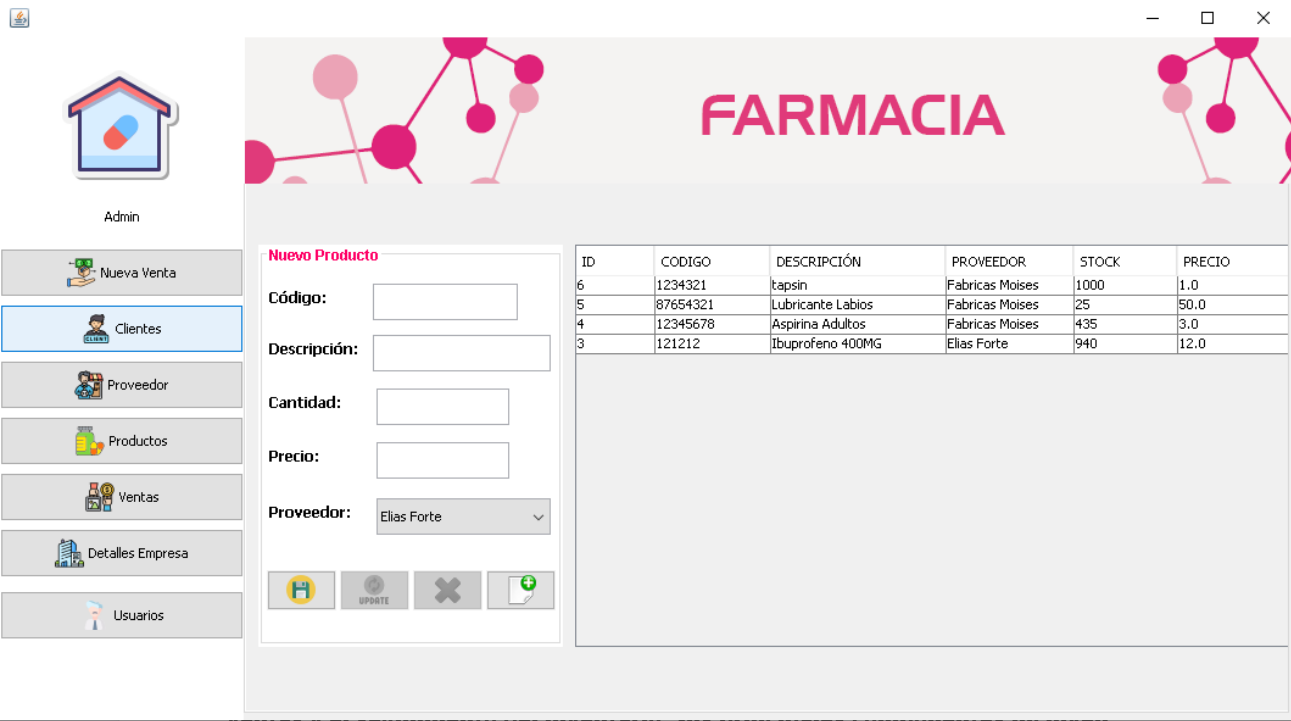
# MANUAL TECNICO: SISTEMA DE VENTAS DE FARMACIA

El proyecto de Sistema de Ventas de Farmacia tiene como objetivo principal modernizar y mejorar la eficiencia de las operaciones de venta y gestión de inventario en una farmacia. Este sistema tiene la misión de optimizar la experiencia tanto para el personal de la farmacia como para los clientes, al tiempo que garantiza un manejo efectivo de los medicamentos y productos disponibles.

## Visión General:


El sistema se desarrolla en respuesta a las necesidades de una farmacia moderna, con la intención de proporcionar un enfoque integral para la administración de ventas y el seguimiento del inventario. Sus principales componentes incluyen:

**Gestión de Inventario:** El sistema llevará un registro actualizado y detallado de todos los productos en stock, lo que facilitará la gestión de reabastecimiento y la reducción de pérdidas debido a la caducidad.



**Procesamiento de Ventas:** Se automatizarán las transacciones de venta, agilizando el proceso de compra para los clientes y brindando un control más preciso a los empleados de la farmacia.





Admin

Nueva Venta

Cientes

Proveedor

Productos

Ventas

Detalles Empresa

Usuarios

Código

Descripción

Cant

Precio

22/09/2023

Stock Disponible

ID	DESCRIPCIÓN	CANTIDAD	PRECIO U.	PRECIO TOTAL
----	-------------	----------	-----------	--------------


Dni/Ruc

Nombre:

Generar Factura

Total a Pagar: -----

**Control de Recetas Médicas:** El sistema permitirá la gestión electrónica de recetas médicas, asegurando la legalidad de las ventas de medicamentos controlados y simplificando la atención al paciente.



Admin

Nueva Venta

Cientes

Proveedor

Productos

Ventas

Detalles Empresa

Usuarios

Historial Ventas

jTextField1

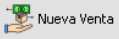
ID	CLIENTE	VENDEDOR	TOTAL
31	Jorje sanic	Admin	90.0
32	Jorje sanic	Admin	90.0
34	Jorje sanic	Admin	150.0
35	Marcos Diaz	Admin	135.0
30	Eto Mamtin	Admin	147.0
33	Eto Mamtin	Admin	150.0

**Reportes y Análisis:** Proporcionará herramientas de generación de informes que ayudarán a los propietarios de la farmacia a tomar decisiones informadas sobre el negocio.





Admin



Nueva Venta



Clientes



Proveedor



Productos



Ventas



Detalles Empresa



Usuarios

# FARMACIA



## Historial Ventas

jTextField1

ID	CLIENTE	VENDEDOR	TOTAL
31	Jorje sanic	Admin	90.0
32	Jorje sanic	Admin	90.0
34	Jorje sanic	Admin	150.0
35	Marcos Diaz	Admin	135.0
30	Eto Mantin	Admin	147.0
33	Eto Mantin	Admin	150.0

**Seguridad de Datos:** Se implementarán medidas de seguridad robustas para proteger la información sensible de los clientes y el negocio.



## Iniciar Sesión



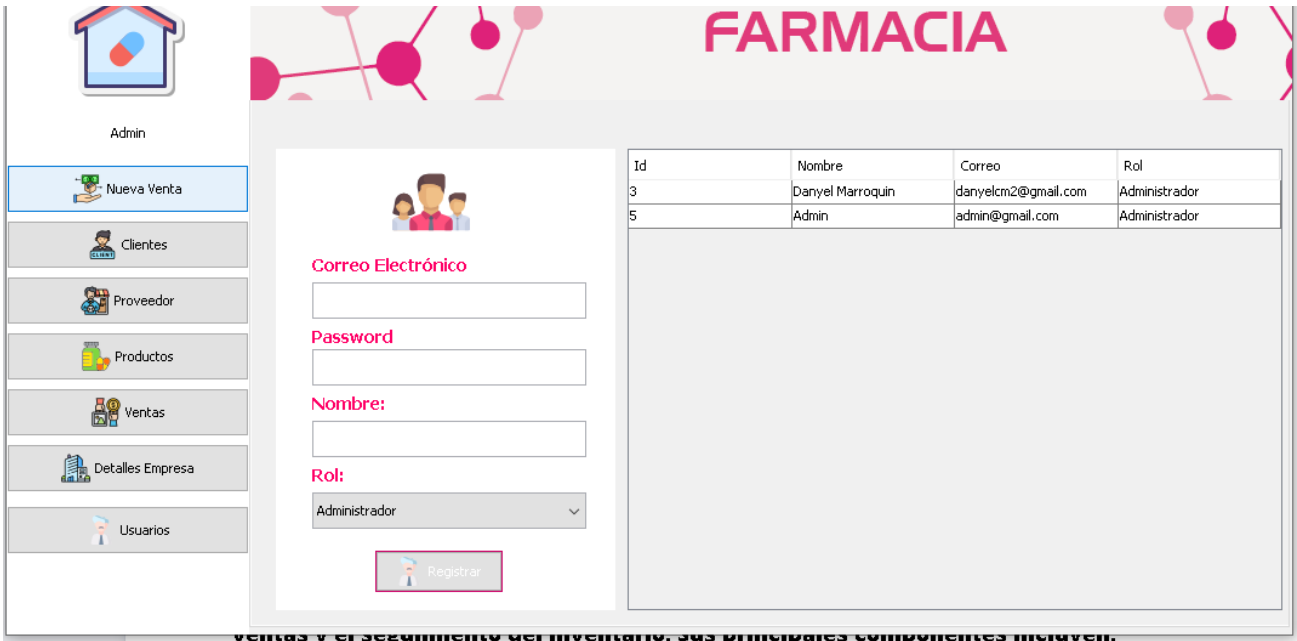
Correo Electrónico

admin@gmail.com|

Password

••••

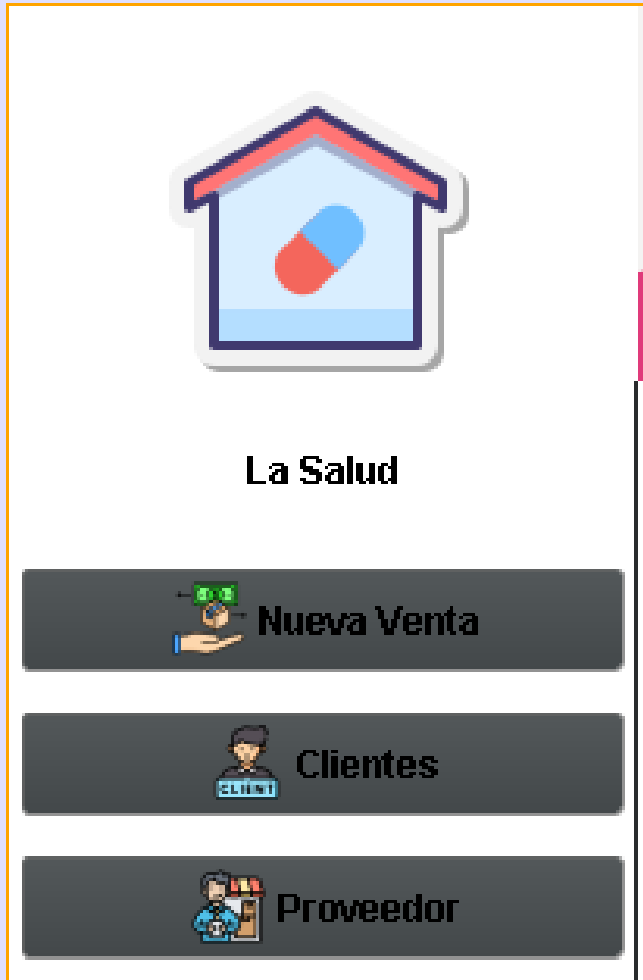
Login



ventas y el seguimiento del inventario. Sus principales componentes incluyen.

Visión General:

El sistema se desarrolla en respuesta a las necesidades de una farmacia moderna, con la intención de proporcionar un enfoque integral para la administración de ventas y el seguimiento del inventario. Sus principales componentes incluyen:



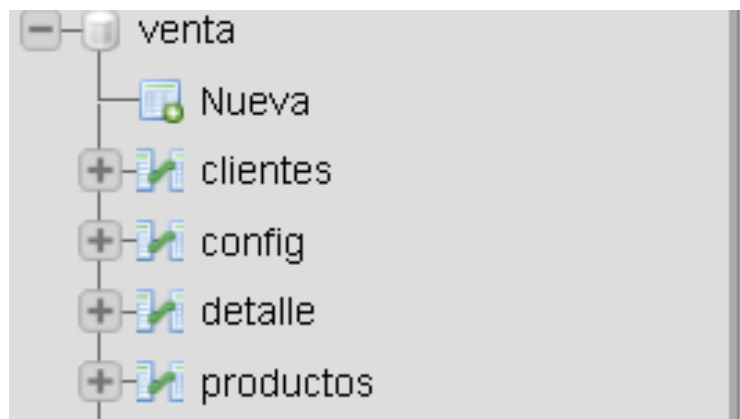


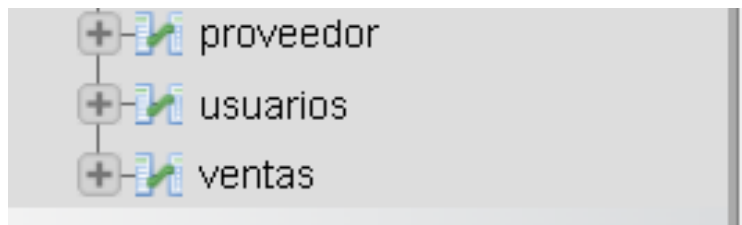
### Introducción al Sistema de Ventas de Farmacia:

En un contexto donde la gestión eficiente de farmacias es esencial, el Sistema de Ventas de Farmacia se presenta como una solución tecnológica integral para abordar los desafíos modernos de esta industria. Este proyecto tiene como objetivo principal mejorar la eficiencia operativa, garantizar el cumplimiento normativo y elevar la experiencia del cliente a través de la automatización de procesos de venta, gestión de inventario y control de recetas médicas. Este sistema representa un avance hacia una gestión farmacéutica más precisa y orientada a la satisfacción del paciente en la era digital.

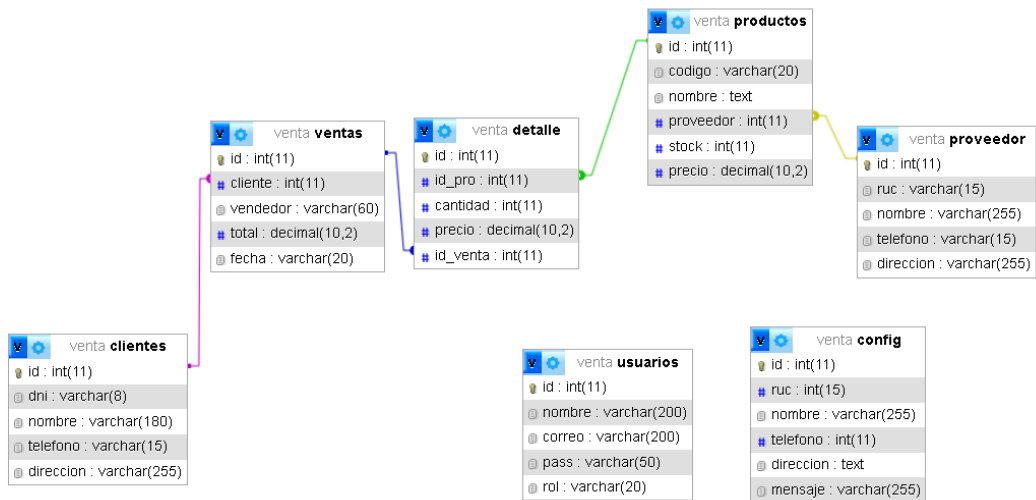
### Arquitectura del Proyecto:

Tablas de nuestra base de datos:





### Diagrama UML de la base de datos con todas sus tablas:



### Código fuente de login:


```

1 package Vista;
2
3
4 import Modelo.LoginDAO;
5 import Modelo.login;
6 import java.awt.Desktop;
7 import java.io.IOException;
8 import java.net.URI;
9 import java.net.URISyntaxException;
10 import javax.swing.JOptionPane;
11
12
13 public class Login extends javax.swing.JFrame {
14     login lg = new login();
15     LoginDAO login = new LoginDAO();
16     public Login() {
17         initComponents();
18         this.setLocationRelativeTo(null);
19         txtCorreo.setText("admin@gmail.com");
20         txtPass.setText("123");
21     }
22     public void validar(){
23         String correo = txtCorreo.getText();
24         String pass = String.valueOf(txtPass.getPassword());
25         if (!"".equals(correo) || !"".equals(pass)) {
26
27             lg = login.log(correo, pass);
28             if (lg.getCorreo() != null && lg.getPass() != null) {
29                 Sistema sis = new Sistema(lg);
30                 sis.setVisible(true);
31                 dispose();
32             }else{

```

```
33 JOptionPane.showMessageDialog(null, "Correo o la Contraseña incorrecta");
34 }
35 }
36 }
37 /**
```

## Iniciar Sesión

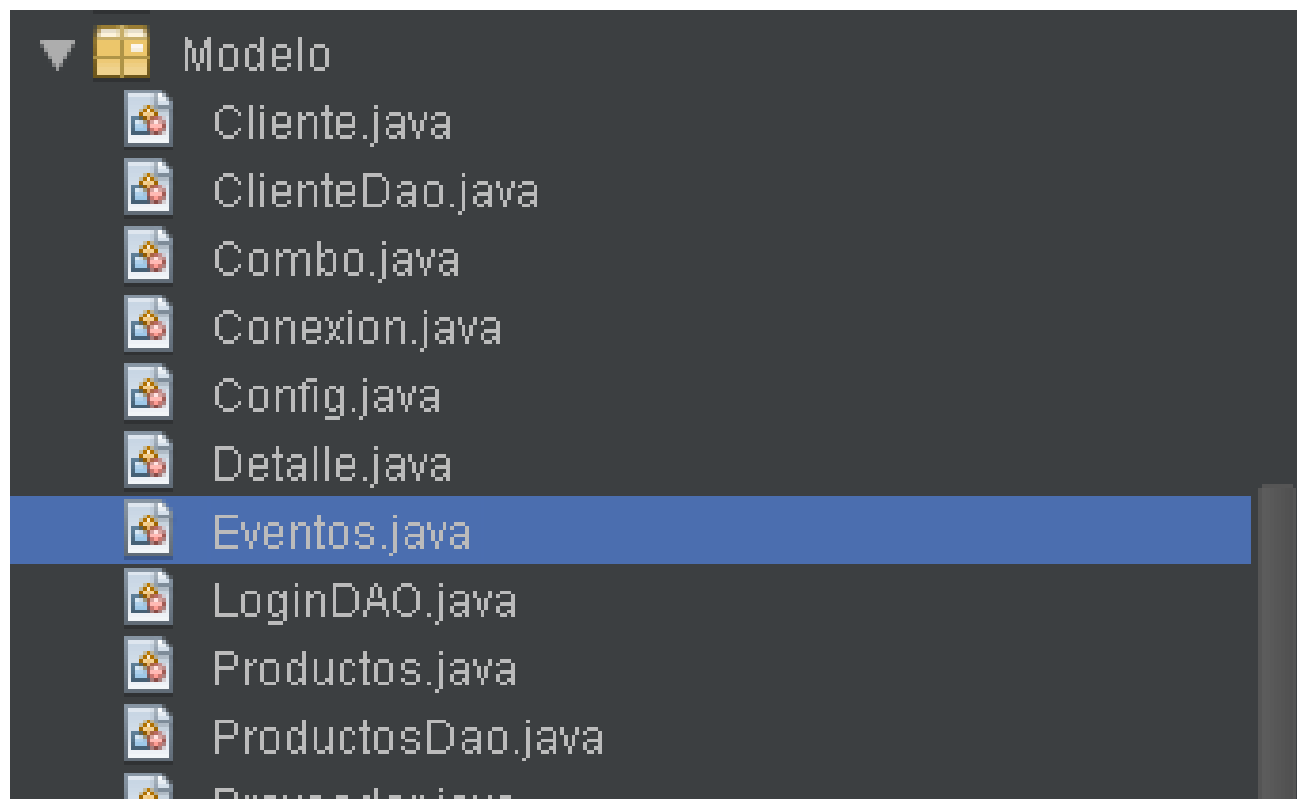


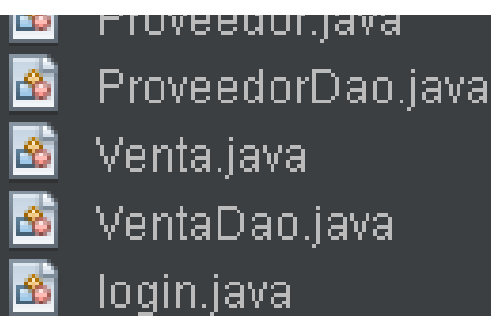
Correo Electrónico

Password

Login

### Modelos utilizados en el desarrollo





## Código general del sistema:

```

1  * @author USUARIO
2  */
3
4  public final class Sistema extends javax.swing.JFrame {
5      Date fechaVenta = new Date();
6      String fechaActual = new SimpleDateFormat("dd/MM/yyyy").format(fechaVenta);
7      Cliente cl = new Cliente();
8      ClienteDao client = new ClienteDao();
9      Proveedor pr = new Proveedor();
10     ProveedorDao PrDao = new ProveedorDao();
11     Productos pro = new Productos();
12     ProductosDao proDao = new ProductosDao();
13     Venta v = new Venta();
14     VentaDao Vdao = new VentaDao();
15     Detalle Dv = new Detalle();
16     Config conf = new Config();
17     Eventos event = new Eventos();
18     login lg = new login();
19     LoginDAO login = new LoginDAO();
20     DefaultTableModel modelo = new DefaultTableModel();
21     DefaultTableModel tmp = new DefaultTableModel();
22     int item;
23     double Totalpagar = 0.00;
24
25     public Sistema() {
26         initComponents();
27     }
28
29     public Sistema(login priv){
30         initComponents();
31         this.setLocationRelativeTo(null);
32         Mdate.setDate(fechaVenta);
33         btIdCliente.setVisible(false);
34         btIdVenta.setVisible(false);
35         btIdPro.setVisible(false);
36         btIdProducto.setVisible(false);
37         btIdProveedor.setVisible(false);
38         btIdConfig.setVisible(false);
39         btIdCV.setVisible(false);
40         ListarConfig();
41         if (priv.getRol().equals("Asistente")) {
42             btnProductos.setEnabled(false);
43             btnProveedor.setEnabled(false);
44             LabelVendedor.setText(priv.getNombre());
45         } else {
46             LabelVendedor.setText(priv.getNombre());
47         }
48     }
49
50     public void ListarCliente() {
51         List<Cliente> ListarCl = client.ListarCliente();
52         modelo = (DefaultTableModel) TableCliente.getModel();
53         Object[] ob = new Object[5];
54         for (int i = 0; i < ListarCl.size(); i++) {
55             ob[0] = ListarCl.get(i).getId();
56             ob[1] = ListarCl.get(i).getDni();
57             ob[2] = ListarCl.get(i).getNombre();
58             ob[3] = ListarCl.get(i).getTelefono();
59             ob[4] = ListarCl.get(i).getDireccion();
60             modelo.addRow(ob);
61         }
62         TableCliente.setModel(modelo);
63     }
64
65     public void ListarProveedor() {
66         List<Proveedor> ListarPr = PrDao.ListarProveedor();
67         modelo = (DefaultTableModel) TableProveedor.getModel();
68         Object[] ob = new Object[5];
69         for (int i = 0; i < ListarPr.size(); i++) {
70             ob[0] = ListarPr.get(i).getId();
71             ob[1] = ListarPr.get(i).getRuc();
72             ob[2] = ListarPr.get(i).getNombre();
73             ob[3] = ListarPr.get(i).getTelefono();
74             ob[4] = ListarPr.get(i).getDireccion();
75             modelo.addRow(ob);
76         }
77         TableProveedor.setModel(modelo);
78     }
79
80     public void ListarUsuarios() {
81         List<login> Listar = login.ListarUsuarios();
82         modelo = (DefaultTableModel) TableUsuarios.getModel();
83         Object[] ob = new Object[4];
84         for (int i = 0; i < Listar.size(); i++) {
85             ob[0] = Listar.get(i).getId();
86             ob[1] = Listar.get(i).getNombre();

```



```

        ob[2] = Listar.get(i).getCorreo();
        ob[3] = Listar.get(i).getRol();
        modelo.addRow(ob);
    }
    TableUsuarios.setModel(modelo);
}

```

```

    }

    public void ListarProductos() {
        List<Productos> ListarPro = proDao.ListarProductos();
        modelo = (DefaultTableModel) TableProducto.getModel();
        Object[] ob = new Object[6];
        for (int i = 0; i < ListarPro.size(); i++) {
            ob[0] = ListarPro.get(i).getId();
            ob[1] = ListarPro.get(i).getCodigo();
            ob[2] = ListarPro.get(i).getNombre();
            ob[3] = ListarPro.get(i).getProveedorPro();
            ob[4] = ListarPro.get(i).getStock();
            ob[5] = ListarPro.get(i).getPrecio();
            modelo.addRow(ob);
        }
        TableProducto.setModel(modelo);
    }

    public void ListarConfig() {
        conf = proDao.BuscarDatos();
        txtIdConfig.setText("" + conf.getId());
        txtRucConfig.setText("" + conf.getRuc());
        txtNombreConfig.setText("" + conf.getNombre());
        txtTelefonoConfig.setText("" + conf.getTelefono());
        txtDireccionConfig.setText("" + conf.getDireccion());
        txtMensaje.setText("" + conf.getMensaje());
    }

    public void ListarVentas() {
        List<Venta> ListarVenta = Vdao.ListarVentas();
        modelo = (DefaultTableModel) TableVentas.getModel();
        Object[] ob = new Object[4];
        for (int i = 0; i < ListarVenta.size(); i++) {
            ob[0] = ListarVenta.get(i).getId();
            ob[1] = ListarVenta.get(i).getNombre_cli();
            ob[2] = ListarVenta.get(i).getVendedor();
            ob[3] = ListarVenta.get(i).getTotal();
            modelo.addRow(ob);
        }
        TableVentas.setModel(modelo);
    }

    public void LimpiarTable() {
        for (int i = 0; i < modelo.getRowCount(); i++) {
            modelo.removeRow(i);
            i = i - 1;
        }
    }
}

```

Desarrollado por **Wincer Daniel Cordova Marroquín** carnet: **1990-22-8308** de la **Universidad Mariano Gálvez de Chimaltenango**

**PROGRAMACION II SECCION "B" : DIRIGIDA POR EL ING. HAROLDO TURCIOS**