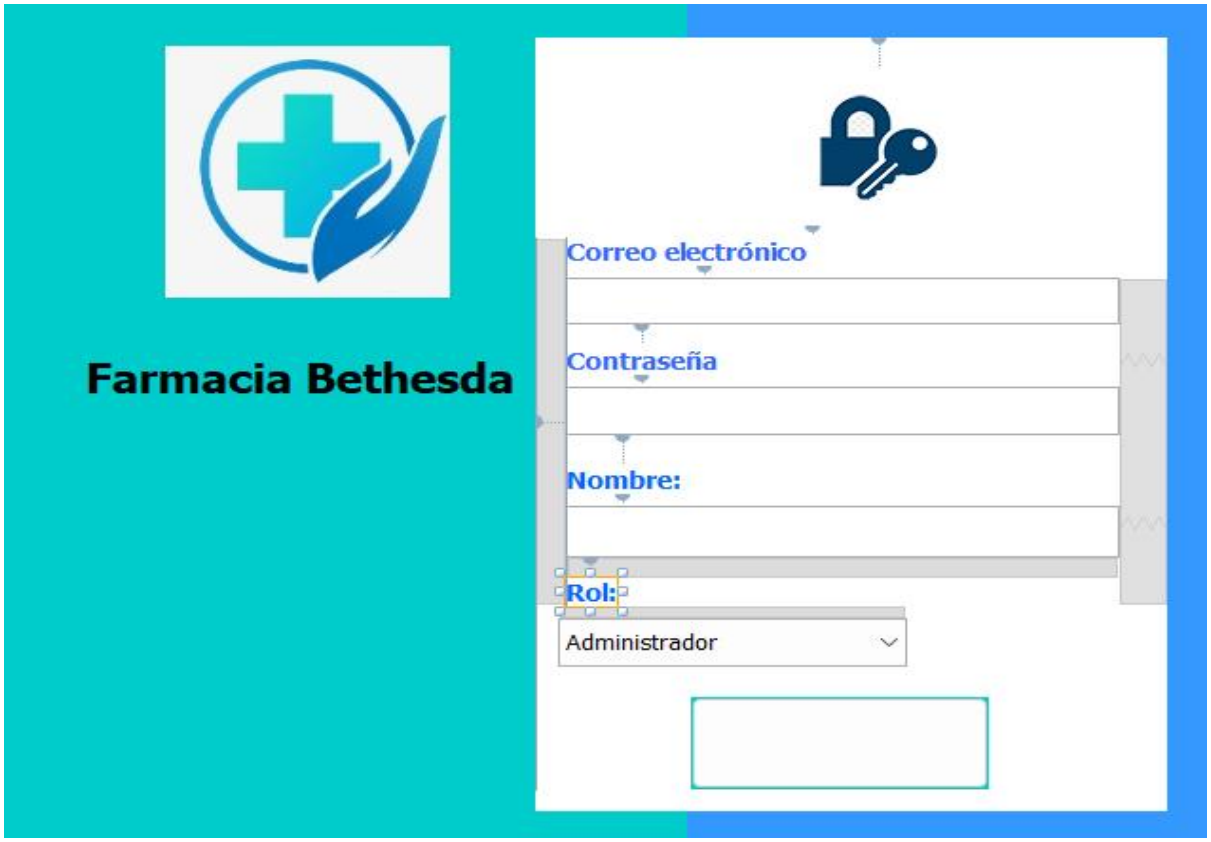
Registro

Nos permite agregar una nueva persona para que pueda ingresar al sistema esto solo lo puede hacer el Administrador.




```java
public class Registro extends javax.swing.JFrame {
    login lg = new login();
    LoginDAO login = new LoginDAO();

    public Registro() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    public void validar(){
        String correo = txtCorreo.getText();
        String pass = String.valueOf(txtPass.getPassword());
        String nom = txtNombre.getText();
        String rol = cbxRol.getSelectedItem().toString();
        if (!"".equals(correo) && !"".equals(pass) && !"".equals(nom)) {
            lg.setNombre(nom);
            lg.setCorreo(correo);
            lg.setPass(pass);
            lg.setRol(rol);
            login.Registrar(lg);
            lg = login.log(correo, pass);
                Login iniciar = new Login();
                iniciar.setVisible(true);
                dispose();
        }
    }
    /**
     * 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")
    Generated Code
```

Login

Aca estan los botones que se utilizan al iniciar el login.

```
public class Login extends javax.swing.JFrame {
    login lg = new login();
    LoginDAO login = new LoginDAO();

    public Login() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    public void validar(){
        String correo = txtCorreo.getText();
        String pass = String.valueOf(txtPass.getPassword());
        if (!"".equals(correo) || !"".equals(pass)) {

            lg = login.log(correo, pass);
            if (lg.getCorreo()!= null && lg.getPass()!= null) {
                Sistema sis = new Sistema(lg);
                sis.setVisible(true);
                dispose();
            }
            else{
                JOptionPane.showMessageDialog(null, "correo o contraseña incorrecta");
            }
        }
    }
```

A

```java
*/
public class login {
    private int id;
    private String nombre;
    private String correo;
    private String pass;
    private String rol;

    public login() {
    }

    public login(int id, String nombre, String correo, String pass, String rol) {
        this.id = id;
        this.nombre = nombre;
        this.correo = correo;
        this.pass = pass;
        this.rol = rol;
    }

    public int getId() {
        return id;
    }

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

    public String getNombre() {
        return nombre;
    }

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

    public String getCorreo() {
        return correo;
    }
}
```

```java
public class LoginDAO {
    Connection con;
    PreparedStatement ps;
    ResultSet rs;
    Conexion cn = new Conexion();

    public login log(String correo, String pass){
        login l = new login();
        String sql = "SELECT * FROM usuarios WHERE correo = ? AND pass = ?";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setString(1, correo);
            ps.setString(2, pass);
            rs = ps.executeQuery();
            if (rs.next()){
                l.setId(rs.getInt("Id"));
                l.setNombre(rs.getString("nombre"));
                l.setCorreo(rs.getString("correo"));
                l.setPass(rs.getString("pass"));
                l.setRol(rs.getString("rol"));
            }
        }catch (SQLException e){
          System.out.println(e.toString());
        }
        return l;
    }

    public boolean Registrar(login reg){
        String sql = "INSERT INTO usuarios (nombre, correo, pass, rol) VALUES (?,?,?,?)";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setString(1, reg.getNombre());
            ps.setString(2, reg.getCorreo());
            ps.setString(3, reg.getPass());
            ps.setString(4, reg.getRol());
            ps.execute();
            return true;
        }catch(SQLException e){
            System.out.println(e.toString());
            return false;
        }
    }
```

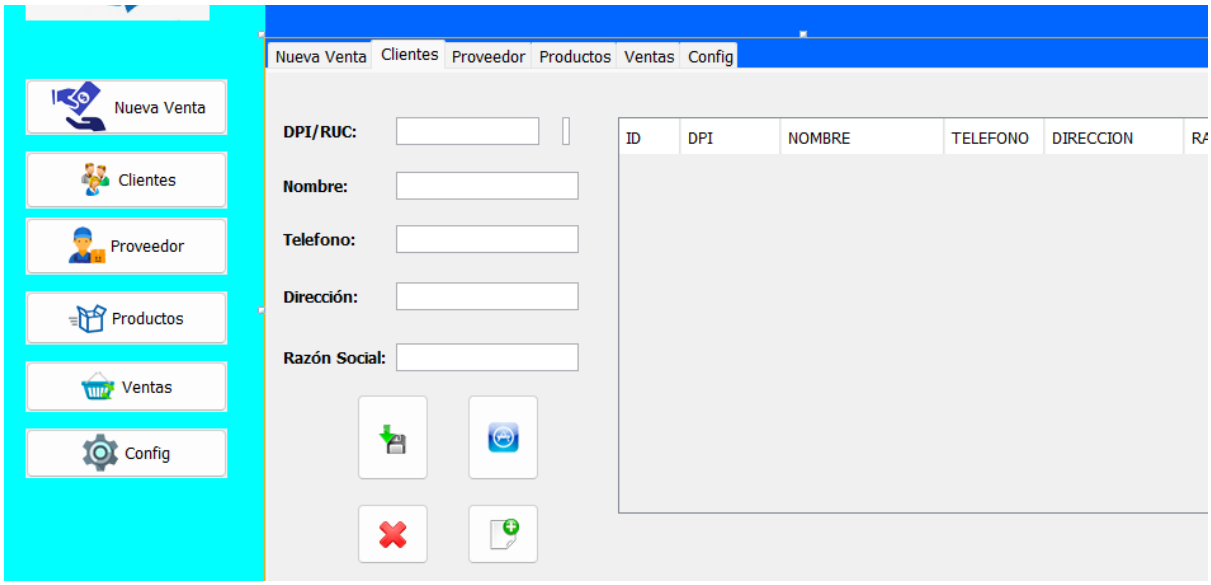
La venta principal contiene muchos botones los cuales están asignados para acción que queramos realizar dejare una breve explicación de los botones que usa cada modulo

Ventas Nuevas


```
    private void btnNuevaVentaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jTabbedPanel.setSelectedIndex(0);
    }

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

    private void btnEliminarventaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        modelo = (DefaultTableModel) TableVenta.getModel();
        modelo.removeRow(TableVenta.getSelectedRow());
        TotalPagar();
        txtCodigoVenta.requestFocus();
    }
```

```java
private void txtCodigoVentaKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode() == KeyEvent.VK_ENTER){
        if(!"".equals(txtCodigoVenta.getText())){
            String cod = txtCodigoVenta.getText();
            pro = proDao.BuscarPro(cod);
            if(pro.getNombre() != null){
                txtDescripcionVenta.setText(""+pro.getNombre());
                txtPrecioVenta.setText(""+pro.getPrecio());
                txtStockDisponible.setText(""+pro.getStock());
                txtCantidadVenta.requestFocus();
            }else{
                JOptionPane.showMessageDialog(null, "El codigo no existe");
                LimpiarVenta();
                txtCodigoVenta.requestFocus();
            }
        }else{
            JOptionPane.showMessageDialog(null, "Ingrese el codigo del Producto");
            txtCodigoVenta.requestFocus();
        }
    }
}

private void txtCantidadVentaKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode() == KeyEvent.VK_ENTER){
        if(!"".equals(txtCantidadVenta.getText())){
            String cod = txtCodigoVenta.getText();
            String descripcion = txtDescripcionVenta.getText();
            int cant = Integer.parseInt(txtCantidadVenta.getText());
            double precio = Double.parseDouble(txtPrecioVenta.getText());
            double total = cant * precio;
            int stock = Integer.parseInt(txtStockDisponible.getText());
            if(stock >= cant){
                item = item + 1;
                tmp = (DefaultTableModel) TableVenta.getModel();
                for(int i = 0; i < TableVenta.getRowCount(); i++){
                    if(TableVenta.getValueAt(i, 1).equals(txtDescripcionVenta.getText())){
                        JOptionPane.showMessageDialog(null, "El producto ya esta registrado");
                        return;
                    }
                }
```

Clientes



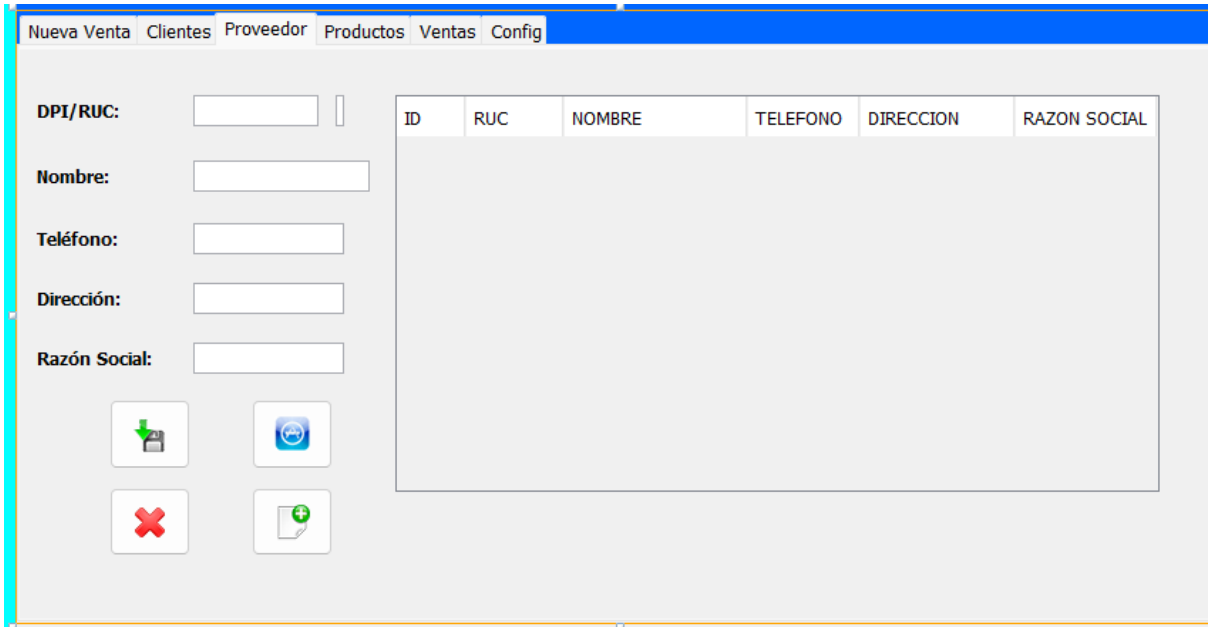
Aca estan los codigos de los botones utilizados en la ventana clientes

```
private void btnGuardarClienteActionPerformed(java.awt.event.ActionEvent evt) {
    if (!"".equals(txtRucConfig.getText()) && !"".equals(txtNombreConfig.getText()) && !"".equals(txtTelefonoConfig.getText()) && !"".equals(txtDireccionConfig.getText()) &&
        conf.setRuc(Integer.parseInt(txtRucConfig.getText()));
        conf.setNombre(txtNombreConfig.getText());
        conf.setTelefono(Integer.parseInt(txtTelefonoConfig.getText()));
        conf.setDireccion(txtDireccionConfig.getText());
        conf.setRazon(txtRazonConfig.getText());
        conf.setId(Integer.parseInt(txtIdConfig.getText()));
        proDao.ModificarDatos(conf);
        JOptionPane.showMessageDialog(null, "Cliente Registrado");
        ListarConfig();
    } else {
        JOptionPane.showMessageDialog(null, "Los campos están vacios");
    }

}
```

```
    private void btnClientesActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        LimpiarTable();
        ListarCliente();
        jTabbedPane1.setSelectedIndex(1);
    }

    private void TableClienteMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int fila = TableCliente.rowAtPoint(evt.getPoint());
        txtIdCliente.setText(TableCliente.getValueAt(fila, 0).toString());
        txtDniCliente.setText(TableCliente.getValueAt(fila, 1).toString());
        txtNombreCliente.setText(TableCliente.getValueAt(fila, 2).toString());
        txtTelefonoCliente.setText(TableCliente.getValueAt(fila, 3).toString());
        txtDirecionCliente.setText(TableCliente.getValueAt(fila, 4).toString());
        txtRazonCliente.setText(TableCliente.getValueAt(fila, 5).toString());
    }

    private void btnEliminarClienteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (!"".equals(txtIdCliente.getText())) {
            int pregunta = JOptionPane.showConfirmDialog(null, "Estas seguro de eliminar");
            if (pregunta == 0) {
                int id = Integer.parseInt(txtIdCliente.getText());
                client.EliminarCliente(id);
                LimpiarTable();
                LimpiarCliente();
                ListarCliente();
            }
        }
    }
```

```
    public boolean RegistrarCliente(Cliente cl){
        String sql = "INSERT INTO clientes (dni, nombre, telefono, direccion, razon) VALUES (?,?,?,?,?)";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setInt(1, cl.getDni());
            ps.setString(2, cl.getNombre());
            ps.setInt(3, cl.getTelefono());
            ps.setString(4, cl.getDireccion());
            ps.setString(5, cl.getRazon());
            ps.execute();
            return true;
        }catch (SQLException e){
            JOptionPane.showMessageDialog(null, e.toString());
            return false;
        }
        finally{
            try{
                con.close();
            }catch(SQLException e){
                System.out.println(e.toString());
            }
        }
```

Proveedor



Aca están los codigos de los botomes que se utilizan en la ventana Proveedor

```
private void btnguardarProveedorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (!"".equals(txtRucProveedor.getText()) || !"".equals(txtNombreProveedor.getText()) || !"".equals(txtTelefonoProveedor.getText()) || !"".equals(txtDireccionProveedor.g
        pr.setRuc(Integer.parseInt(txtRucProveedor.getText()));
        pr.setNombre(txtNombreProveedor.getText());
        pr.setTelefono(Integer.parseInt(txtTelefonoProveedor.getText()));
        pr.setDireccion(txtDireccionProveedor.getText());
        pr.setRazon(txtRazonProveedor.getText());
        PrDao.RegistrarProveedor(pr);
        JOptionPane.showMessageDialog(null, "Proveedor registrado");
        LimpiarTable();
        ListarProveedor();
        LimpiarProveedor();

    } else {
        JOptionPane.showMessageDialog(null, "Los campos están vacios");
    }
}

private void btnProveedorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    LimpiarTable();
    ListarProveedor();
    jTabbedPanel.setSelectedIndex(2);
}
```

```
private void TableProveedorMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int fila = TableProveedor.rowAtPoint(evt.getPoint());
    txtIdProveedor.setText(TableProveedor.getValueAt(fila, 0).toString());
    txtRucProveedor.setText(TableProveedor.getValueAt(fila, 1).toString());
    txtNombreProveedor.setText(TableProveedor.getValueAt(fila, 2).toString());
    txtTelefonoProveedor.setText(TableProveedor.getValueAt(fila, 3).toString());
    txtDireccionProveedor.setText(TableProveedor.getValueAt(fila, 4).toString());
    txtRazonProveedor.setText(TableProveedor.getValueAt(fila, 5).toString());
}

private void btnEliminarProveedorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (!"".equals(txtIdProveedor.getText())) {
        int pregunta = JOptionPane.showConfirmDialog(null, "Estas seguro de eliminar");
        if (pregunta == 0) {
            int id = Integer.parseInt(txtIdProveedor.getText());
            PrDao.EliminarProveedor(id);
            LimpiarTable();
            ListarProveedor();
            LimpiarProveedor();
        }
    } else {
        JOptionPane.showMessageDialog(null, "Seleccione una fila");
    }
}

private void btnEditarProveedorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if ("".equals(txtIdProveedor.getText())) {
        JOptionPane.showMessageDialog(null, "Seleccione una fila");
    } else {
        if (!"".equals(txtRucProveedor.getText()) && !"".equals(txtNombreProveedor.getText()) && !"".equals(txtTelefonoProveedor.getText()) && !"".equals(txtDireccionProveed
            pr.setRuc(Integer.parseInt(txtRucProveedor.getText()));
            pr.setNombre(txtNombreProveedor.getText());
            pr.setTelefono(Integer.parseInt(txtTelefonoProveedor.getText()));
            pr.setDireccion(txtDireccionProveedor.getText());
            pr.setRazon(txtRazonProveedor.getText());
            pr.setId(Integer.parseInt(txtIdProveedor.getText()));
            PrDao.ModificarProveedor(pr);
            LimpiarTable();
            ListarProveedor();
            LimpiarProveedor();
            JOptionPane.showMessageDialog(null, "Proveedor modificado");
```

```java
public class ProveedorDAO {
    Connection con;
    Conexion cn = new Conexion();
    PreparedStatement ps;
    ResultSet rs;
    public boolean RegistrarProveedor(Proveedor pr){
        String sql = "INSERT INTO proveedor(ruc, nombre, telefono, direccion, razon) VALUES (?,?,?,?,?)";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setInt(1, pr.getRuc());
            ps.setString(2, pr.getNombre());
            ps.setInt(3, pr.getTelefono());
            ps.setString(4, pr.getDireccion());
            ps.setString(5, pr.getRazon());
            ps.execute();
            return true;
        }catch (SQLException e){
            System.out.println(e.toString());
            return false;
        }finally{
            try {
                con.close();
            } catch (SQLException e) {
                System.out.println(e.toString());
            }
        }
    }
```

```java
    public List ListarProveedor(){
        List<Proveedor> Listapr = new ArrayList();
        String sql = "SELECT * FROM proveedor";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while (rs.next()){
                Proveedor pr = new Proveedor();
                pr.setId(rs.getInt("id"));
                pr.setRuc(rs.getInt("ruc"));
                pr.setNombre(rs.getString("nombre"));
                pr.setTelefono(rs.getInt("telefono"));
                pr.setDireccion(rs.getString("direccion"));
                pr.setRazon(rs.getString("razon"));
                Listapr.add(pr);
            }
        }catch (SQLException e){
            System.out.println(e.toString());
        }
        return Listapr;
    }

    public boolean EliminarProveedor(int id){
        String sql = "DELETE FROM proveedor WHERE id=?";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setInt(1, id);
            ps.execute();
            return true;
        }catch(SQLException e){
            System.out.println(e.toString());
            return false;
        }finally{
            try {
                con.close();
            } catch (SQLException e) {
                System.out.println(e.toString());
            }
        }
```

Productos

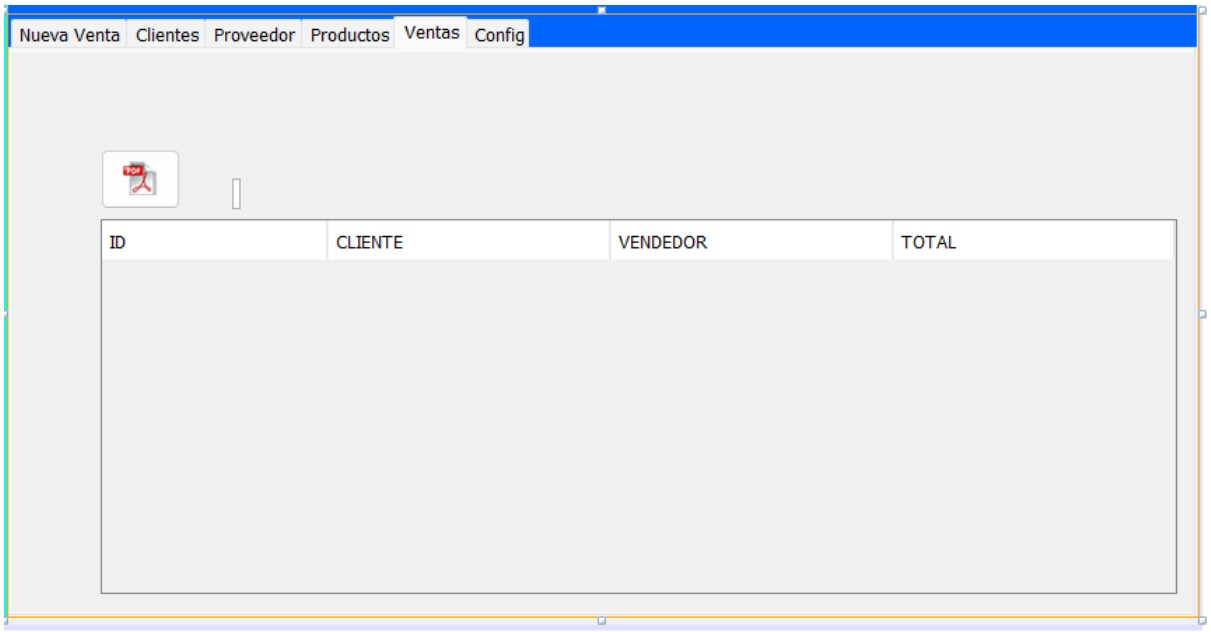
Aca estan los codigos de los botones utilizados en la ventana de productos

```
    private void btnProductosActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        LimpiarTable();
        LimpiarProductos();
        ListarProductos();
        jTabbedPanel.setSelectedIndex(3);
    }

    private void TableProductoMouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        int fila = TableProducto.rowAtPoint(evt.getPoint());
        txtIdPro.setText(TableProducto.getValueAt(fila, 0).toString());
        txtCodigoPro.setText(TableProducto.getValueAt(fila, 1).toString());
        txtDesPro.setText(TableProducto.getValueAt(fila, 2).toString());
        cbxProveedorPro.setSelectedItem(TableProducto.getValueAt(fila, 3).toString());
        txtCantPro.setText(TableProducto.getValueAt(fila, 4).toString());
        txtPrecioPro.setText(TableProducto.getValueAt(fila, 5).toString());
    }

    private void btnEliminarProActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (!"".equals(txtIdPro.getText())) {
            int pregunta = JOptionPane.showConfirmDialog(null, "Estas seguro de eliminar");
            if (pregunta == 0) {
                int id = Integer.parseInt(txtIdPro.getText());
                proDao.EliminarProductos(id);
                LimpiarTable();
                LimpiarProductos();
                ListarProductos();
            }
        }
    }
```

```
    public Config BuscarDatos(){
        Config conf = new Config();
        String sql = "SELECT * FROM config";
        try{
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            if(rs.next()){
                conf.setId(rs.getInt("id"));
                conf.setRuc(rs.getInt("ruc"));
                conf.setNombre(rs.getString("nombre"));
                conf.setTelefono(rs.getInt("telefono"));
                conf.setDireccion(rs.getString("direccion"));
                conf.setRazon(rs.getString("razon"));
            }
        }catch (SQLException e){
            System.out.println(e.toString());
        }
        return conf;
    }

        public boolean ModificarDatos(Config conf){
        String sql = "UPDATE config SET ruc=?, nombre=?, telefono=?, direccion=?, razon=? WHERE id=?";
        try{
            ps = con.prepareStatement(sql);
            ps.setInt(1, conf.getRuc());
            ps.setString(2, conf.getNombre());
            ps.setInt(3, conf.getTelefono());
            ps.setString(4, conf.getDireccion());
            ps.setString(5, conf.getRazon());
            ps.setInt(6, conf.getId());
            ps.execute();
            return true;
        }catch(SQLException e){
            System.out.println(e.toString());
            return false;
        } finally{
            try {
                con.close();
            } catch (SQLException ex) {
                System.out.println(ex.toString());
            }
        }
    }
```

```java
public Productos BuscarPorDescripcion(String descripcion) {
    Productos producto = new Productos();
    String sql = "SELECT * FROM productos WHERE nombre = ?";
    try {
        con = cn.getConnection(); // Obtener la conexión
        ps = con.prepareStatement(sql);
        ps.setString(1, descripcion);
        rs = ps.executeQuery();
        if (rs.next()) {
            producto.setCodigo(rs.getString("codigo")); // Asumiendo que tienes un método setCodigo
            producto.setNombre(rs.getString("nombre"));
            producto.setPrecio(rs.getDouble("precio"));
            producto.setStock(rs.getInt("stock"));
        }
    } catch (SQLException e) {
        System.out.println(e.toString());
    } finally {
        try {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (con != null) con.close();
        } catch (SQLException ex) {
            System.out.println(ex.toString());
        }
    }
    return producto;
}


}
```

Ventas




```java
    private void btnGenerarVentaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (TableVenta.getRowCount() > 0){
            if(!"".equals(txtNombreClienteventa.getText())){
              RegistrarVenta();
            RegistrarDetalle();
            ActualizarStock();
            pdf();
            LimpiarTableVenta();
            LimpiarClienteventa();
            }else{
                JOptionPane.showMessageDialog(null, "Debes ingresar un cliente");
            }
        }else{
            JOptionPane.showMessageDialog(null, "NO hay productos en la venta");
        }

    }
```

Config


```java
public class Config {
    private int id;
    private int ruc;
    private String nombre;
    private int telefono;
    private String direccion;
    private String razon;

    public Config(){

    }

    public Config(int id, int ruc, String nombre, int telefono, String direccion, String razon) {
        this.id = id;
        this.ruc = ruc;
        this.nombre = nombre;
        this.telefono = telefono;
        this.direccion = direccion;
        this.razon = razon;
    }

    public int getId() {
        return id;
    }

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

    public int getRuc() {
        return ruc;
    }
```

Base de datos