Prueba de caja Blanca

V. 1.0

Fecha: 01/07/2022

Proyecto: Mini Market Tecnology

Asignatura: Ingeniería de Software II

Integrantes:

Ligia Maricela Maldonado Paredes
Rommel Nicolas Zambrano Gaona
Silvia Liliana Yunga Quichimbo
Robert Denilson Sanguña Lanchimba
Saltos Cárdenas Brandon Xavier
Rubén Darío González Gudiño

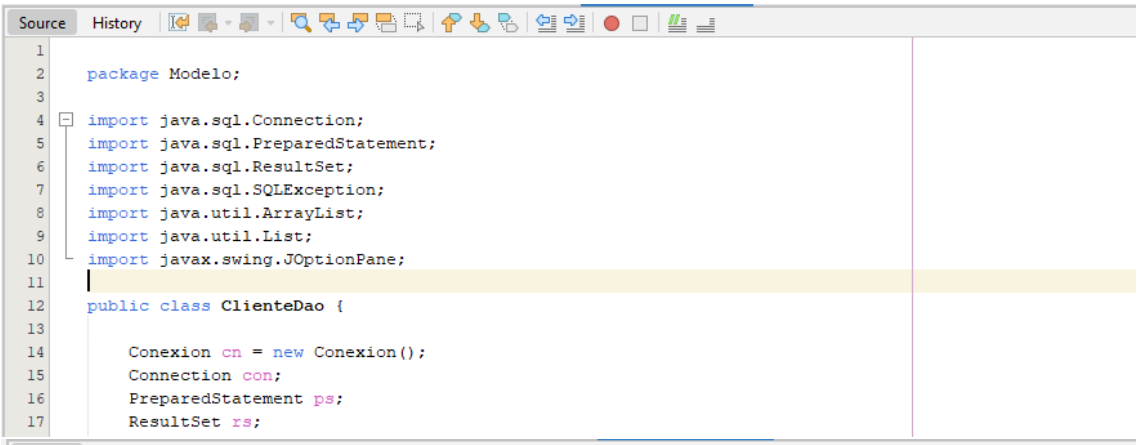
Fecha: 01/07/2022

Asignatura: Ingeniería de Software II

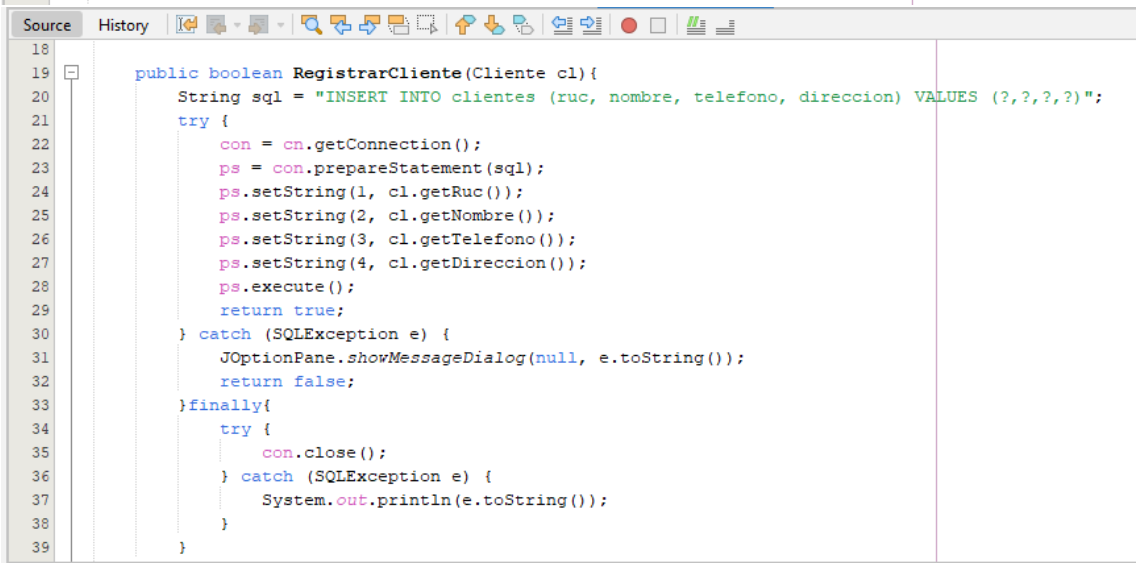
Integrantes:

Registro de clientes

1. Código fuente



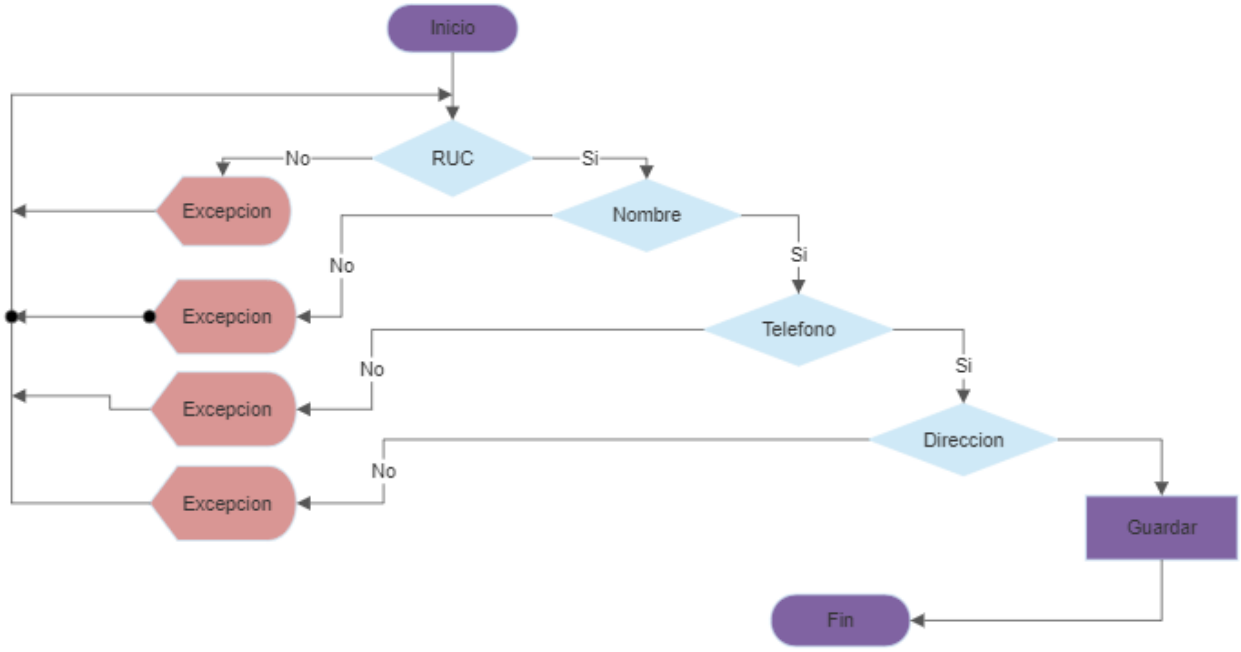
```java

package Modelo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;

public class ClienteDao {

    Conexion cn = new Conexion();
    Connection con;
    PreparedStatement ps;
    ResultSet rs;
```



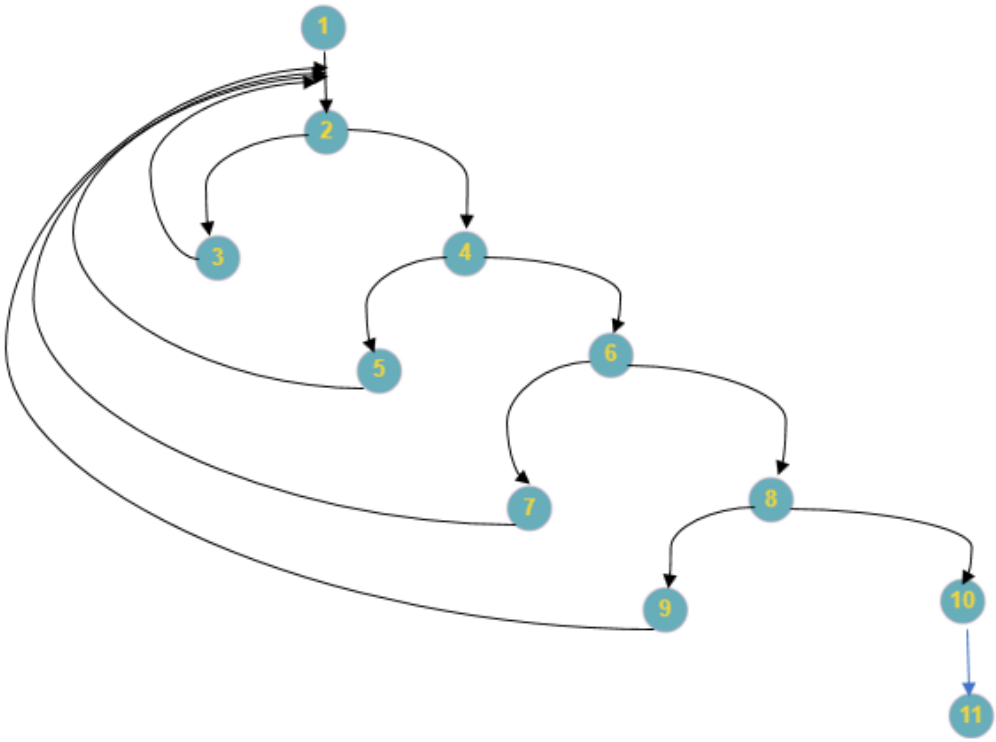
```java

    public boolean RegistrarCliente(Cliente cl){
        String sql = "INSERT INTO clientes (ruc, nombre, telefono, direccion) VALUES (?,?,?,?)";
        try {
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setString(1, cl.getRuc());
            ps.setString(2, cl.getNombre());
            ps.setString(3, cl.getTelefono());
            ps.setString(4, cl.getDireccion());
            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());
            }
        }
```

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

2. Diagrama de flujo



3. Grafo



4. Rutas

R1: 1, 2, 4, 6, 8, 10, 11

R2: 1, 2, 3, 4, 6, 8, 10, 11

R3: 1, 2, 4, 5, 2, 4, 6, 8, 10, 11

R4: 1, 2, 4, 6, 7, 2, 4, 6, 8, 10, 11

R5: 1, 2, 4, 6, 8, 9, 2, 4, 6, 8, 10, 11

5. Complejidad ciclomática

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones) + 1
  V(G) = 4 + 1 = 5
- V(G) = A – N + 2
  V(G)= 13 – 10 + 2 = 5

  DONDE:

  P: Número de nodos predicado = 5

  A: Número de aristas = 13

  N: Número de nodos = 10

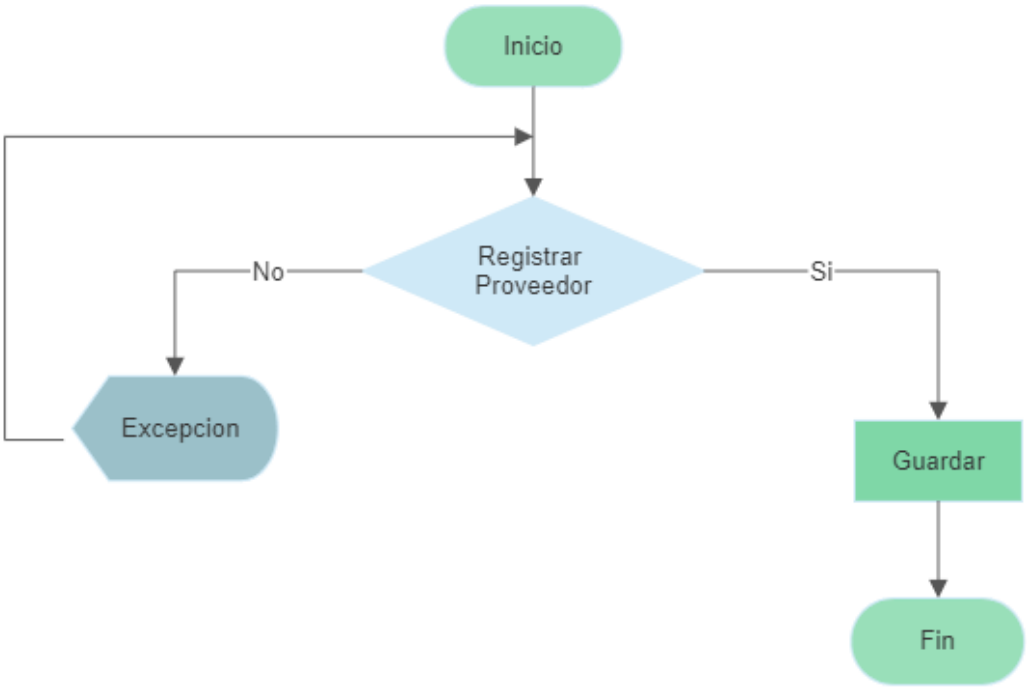
Requisito

Agregar Proveedor

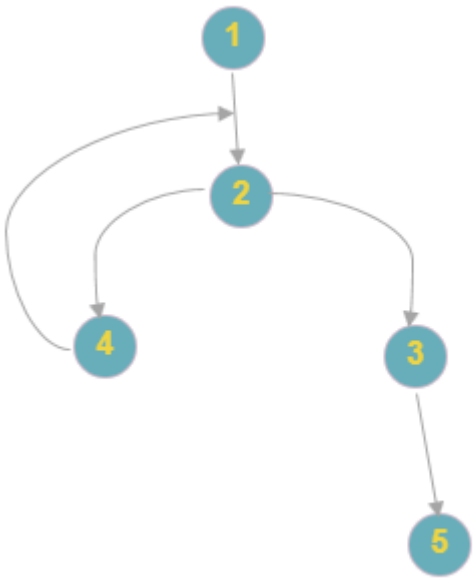
1. Código fuente

```java
package Modelo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

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) VALUES (?,?,?,?)";
        try {
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            ps.setString(1, pr.getRuc());
            ps.setString(2, pr.getNombre());
            ps.setString(3, pr.getTelefono());
            ps.setString(4, pr.getDireccion());
            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());
            }
        }
    }
```

2. Diagrama de flujo



3. Grafo



4. Rutas

R1: 1, 2, 3, 5

R2: 1, 2, 4, 2, 3, 5

5. Complejidad ciclomática

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones) + 1
  V(G) = 1+ 1 = 2
- V(G) = A – N + 2
  V(G)= 5– 5 + 2 = 2

DONDE:

P: Número de nodos predicado = 2

A: Número de aristas = 5

N: Número de nodos = 5

Requisito

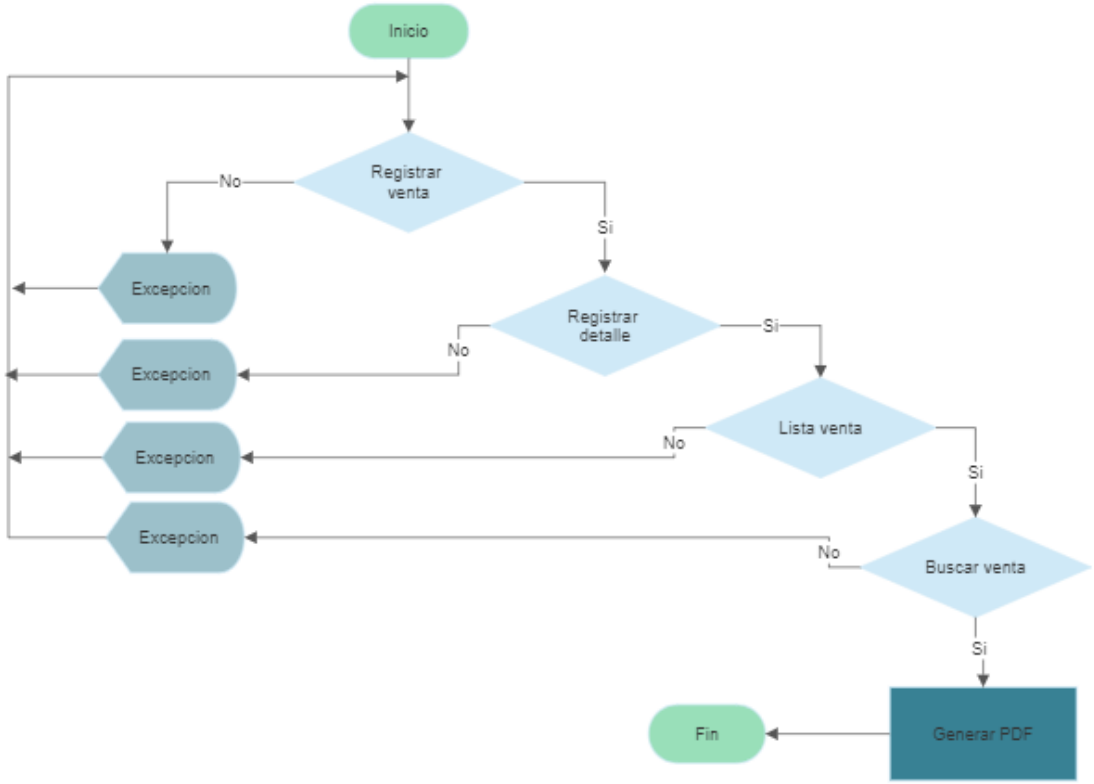
Agregar Venta

1. **Código fuente**

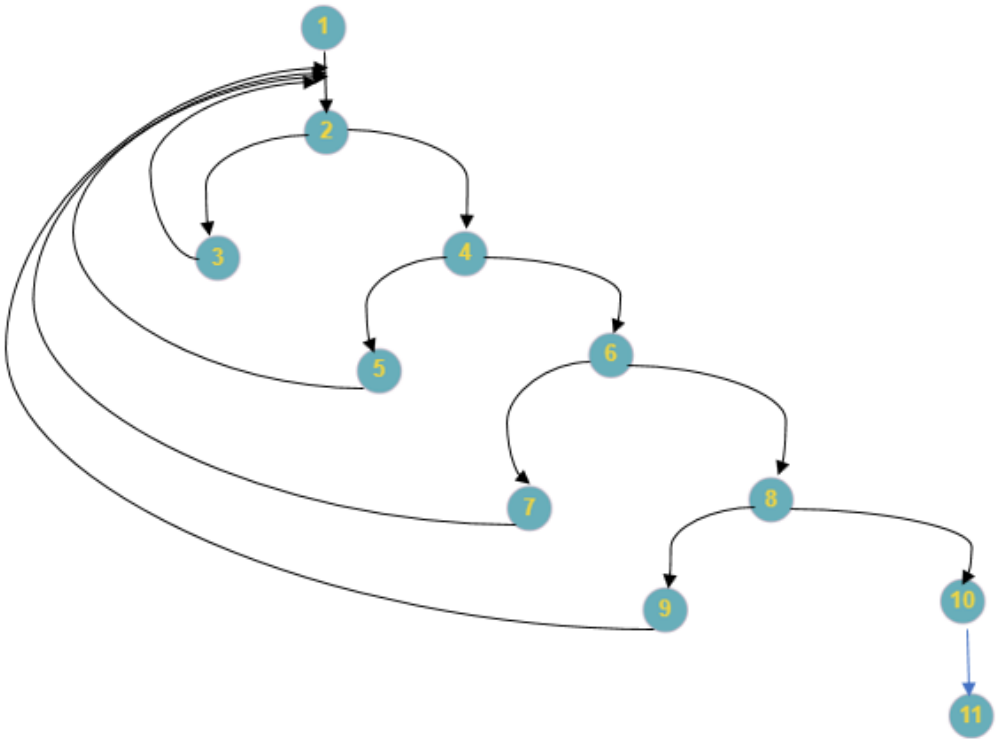
```
public class VentaDao {
    Connection con;
    Conexion cn = new Conexion();
    PreparedStatement ps;
    ResultSet rs;
    int r;

    public int IdVenta(){
        int id = 0;
        String sql = "SELECT MAX(id) FROM ventas";
        try {
            con = cn.getConnection();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            if (rs.next()) {
                id = rs.getInt(1);
            }
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
        return id;
    }
```

```java
public int RegistrarVenta(Venta v){
    String sql = "INSERT INTO ventas (cliente, vendedor, total, fecha) VALUES (?,?,?,?)";
    try {
        con = cn.getConnection();
        ps = con.prepareStatement(sql);
        ps.setInt(1, v.getCliente());
        ps.setString(2, v.getVendedor());
        ps.setDouble(3, v.getTotal());
        ps.setString(4, v.getFecha());
        ps.execute();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }finally{
        try {
            con.close();
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
    }
    return r;
}
public int RegistrarDetalle(Detalle Dv){
   String sql = "INSERT INTO detalle (id_pro, cantidad, precio, id_venta) VALUES (?,?,?,?)";
    try {
        con = cn.getConnection();
        ps = con.prepareStatement(sql);
        ps.setInt(1, Dv.getId_pro());
        ps.setInt(2, Dv.getCantidad());
        ps.setDouble(3, Dv.getPrecio());
        ps.setInt(4, Dv.getId());
        ps.execute();
    } catch (SQLException e) {
        System.out.println(e.toString());
    }finally{
        try {
            con.close();
        } catch (SQLException e) {
            System.out.println(e.toString());
        }
    }
    return r;
}
```

2. Diagrama de flujo



3. Grafo



4. Rutas

R1: 1, 2, 4, 6, 8, 10, 11

R2: 1, 2, 3, 4, 6, 8, 10, 11

R3: 1, 2, 4, 5, 2, 4, 6, 8, 10, 11

R4: 1, 2, 4, 6, 7, 2, 4, 6, 8, 10, 11

R5: 1, 2, 4, 6, 8, 9, 2, 4, 6, 8, 10, 11

5. Complejidad ciclomática

Se puede calcular de las siguientes formas:

- V(G) = número de nodos predicados(decisiones) + 1
  V(G) = 4+ 1 = 5
- V(G) = A – N + 2
  V(G)= 14 – 11 + 2 = 5

  DONDE:

  P: Número de nodos predicado = 5

  A: Número de aristas = 14

  N: Número de nodos = 11