|  |
| --- |
| Prueba de Caja Blanca |
| ***“Gestión de Ventas y Compras DIDEB”*** |
|  |
|  |
| **Integrantes:**  Sebastian Alvarez  Madely Betancourt  Luis Carvajal  Johanna Molina  Carlos Puco  **Fecha 2022-01-31** |

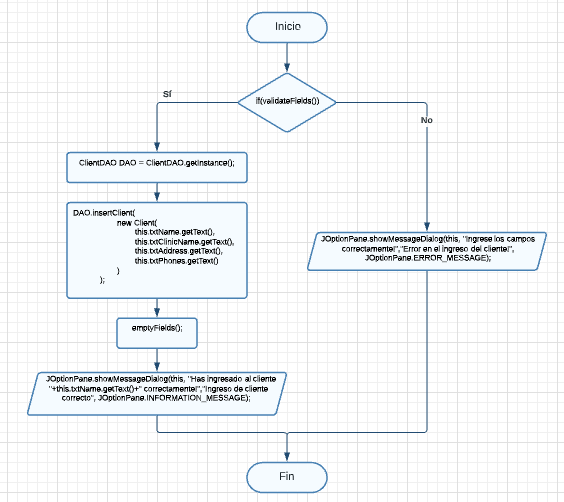
**CLIENTE**

1. Acción al presionar “Agregar” en “Agregar nuevo cliente”

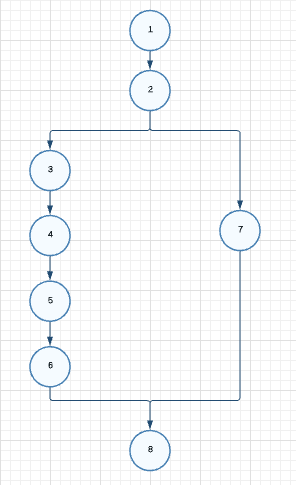
**CÓDIGO FUENTE**

| if(validateFields()){  ClientDAO DAO = ClientDAO.getInstance();  DAO.insertClient(  new Client(  this.txtName.getText(),  this.txtClinicName.getText(),  this.txtAddress.getText(),  this.txtPhones.getText()  )  );    emptyFields();  JOptionPane.showMessageDialog(this, "Has ingresado al cliente "+this.txtName.getText()+" correctamente!","Ingreso de cliente correcto", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(this, "Ingrese los campos correctamente!","Error en el ingreso del cliente!", JOptionPane.ERROR\_MESSAGE);  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****



**RUTAS**

**R1:** 1, 2, 3, 4, 5, 6, 8

**R2:** 1, 2, 7, 8

**COMPLEJIDAD CICLOMÁTICA**

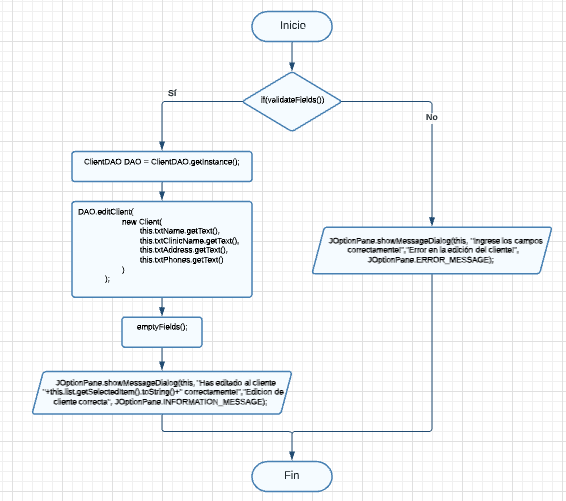
V(G)= 8 – 8 + 2 = 2

1. Acción al presionar “Modificar” en “Modificar cliente”

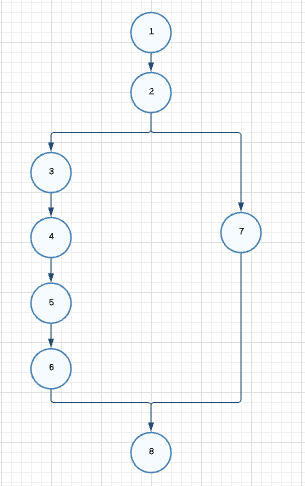
**CÓDIGO FUENTE**

| if(validateFields()){  ClientDAO DAO = ClientDAO.getInstance();  DAO.editClient(  new Client(  this.list.getSelectedItem().toString(),  this.txtClinicName.getText(),  this.txtAddress.getText(),  this.txtPhones.getText()  )  );    emptyFields();  JOptionPane.showMessageDialog(this, "Has editado al cliente "+this.list.getSelectedItem().toString()+" correctamente!","Edicion de cliente correcta", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(this, "Ingrese los campos correctamente!","Error en la edición del cliente!", JOptionPane.ERROR\_MESSAGE);  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****



**RUTAS**

**R1:** 1, 2, 3, 4, 5, 6, 8

**R2:** 1, 2, 7, 8

**COMPLEJIDAD CICLOMÁTICA**

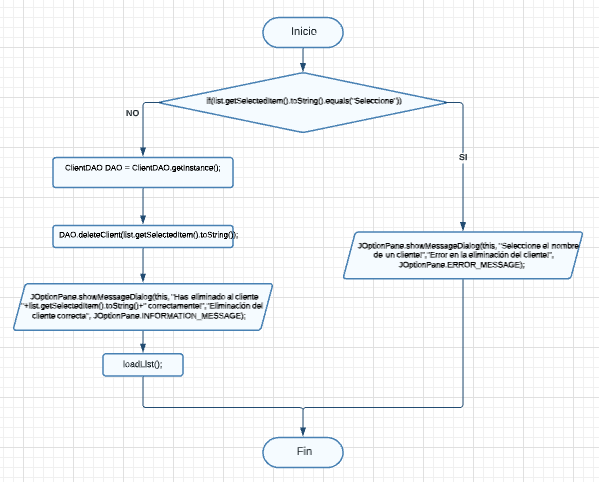
V(G)= 8 – 8 + 2 = 2

1. Acción al presionar “Eliminar” en “Eliminar cliente”

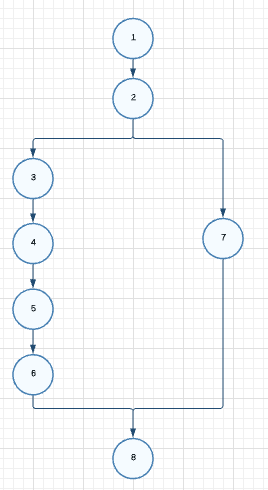
**CÓDIGO FUENTE**

| if(list.getSelectedItem().toString().equals("Seleccione")){  JOptionPane.showMessageDialog(this, "Seleccione el nombre de un cliente!","Error en la eliminación del cliente!", JOptionPane.ERROR\_MESSAGE);  } else {  ClientDAO DAO = ClientDAO.getInstance();  DAO.deleteClient(list.getSelectedItem().toString());    JOptionPane.showMessageDialog(this, "Has eliminado al cliente "+list.getSelectedItem().toString()+" correctamente!","Eliminación del cliente correcta", JOptionPane.INFORMATION\_MESSAGE);  loadList();  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 4, 5, 6, 8

**R2:** 1, 2, 7, 8

**COMPLEJIDAD CICLOMÁTICA**

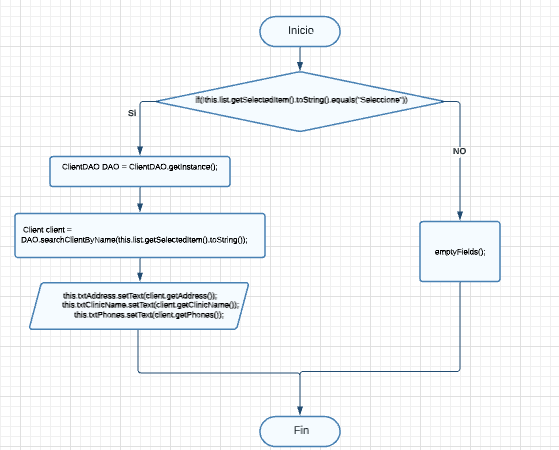
V(G)= 8 – 8 + 2 = 2

1. Acción al presionar “Buscar” en “Buscar cliente por nombre”

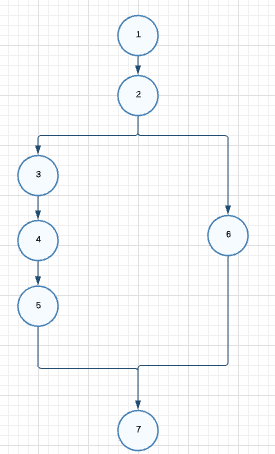
**CÓDIGO FUENTE**

| if(!this.list.getSelectedItem().toString().equals("Seleccione")){  ClientDAO DAO = ClientDAO.getInstance();  Client client = DAO.searchClientByName(this.list.getSelectedItem().toString());  this.txtAddress.setText(client.getAddress());  this.txtClinicName.setText(client.getClinicName());  this.txtPhones.setText(client.getPhones());  } else {  emptyFields();  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 4, 5, 7

**R2:** 1, 2, 6, 7

**COMPLEJIDAD CICLOMÁTICA**

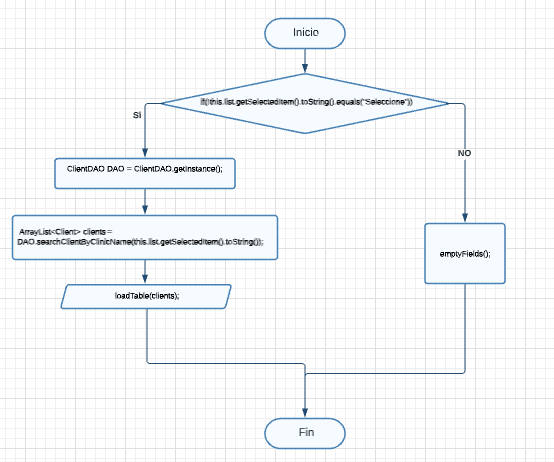
V(G)= 7 – 7 + 2 = 2

1. Acción al presionar “Buscar” en “Buscar cliente por nombre de clínica”

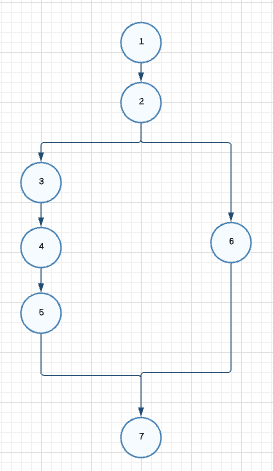
**CÓDIGO FUENTE**

| if(!this.list.getSelectedItem().toString().equals("Seleccione")){  ClientDAO DAO = ClientDAO.getInstance();  ArrayList<Client> clients = DAO.searchClientByClinicName(this.list.getSelectedItem().toString());  loadTable(clients);  } else {  emptyFields();  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****



**RUTAS**

**R1:** 1, 2, 3, 4, 5, 7

**R2:** 1, 2, 6, 7

**COMPLEJIDAD CICLOMÁTICA**

V(G)= 7 – 7 + 2 = 2

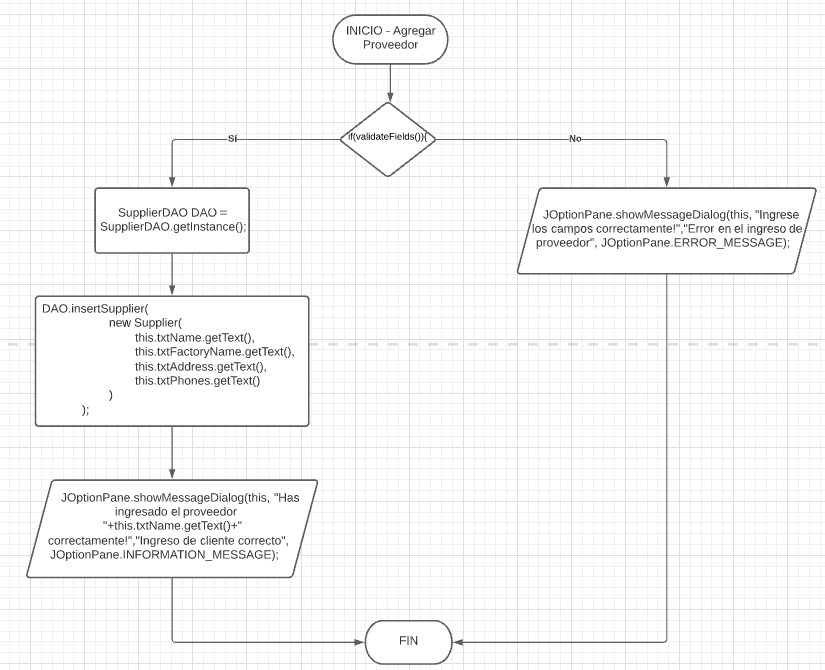
**PROVEEDOR**

**Agregar Proveedor**

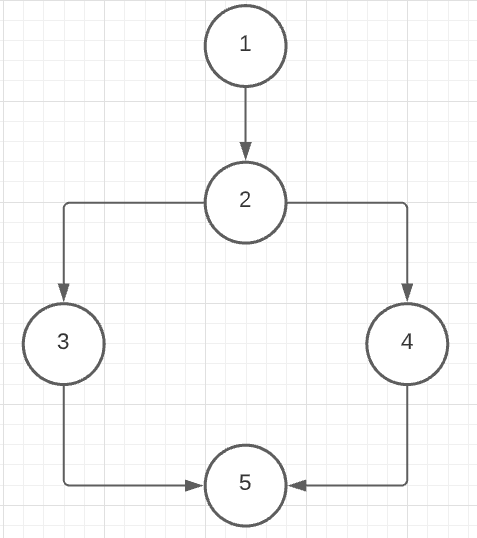
**CÓDIGO FUENTE**

| private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {  if(validateFields()){  SupplierDAO DAO = SupplierDAO.getInstance();  DAO.insertSupplier(  new Supplier(  this.txtName.getText(),  this.txtFactoryName.getText(),  this.txtAddress.getText(),  this.txtPhones.getText()  )  );    emptyFields();  JOptionPane.showMessageDialog(this, "Has ingresado el proveedor "+this.txtName.getText()+" correctamente!","Ingreso de cliente correcto", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(this, "Ingrese los campos correctamente!","Error en el ingreso de proveedor", JOptionPane.ERROR\_MESSAGE);  }  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 5.

**R2:** 1, 2, 4, 5.

**COMPLEJIDAD CICLOMÁTICA**

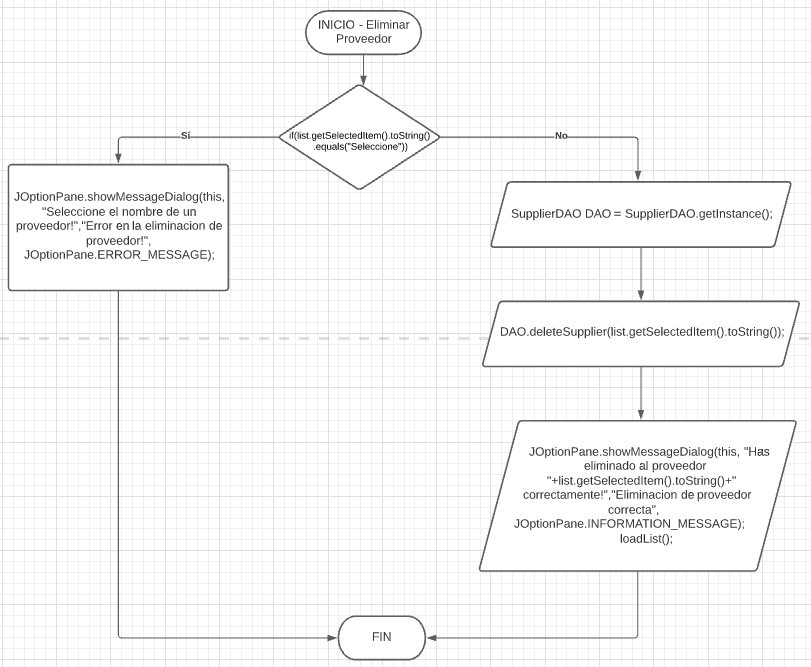
V(G)= 5 – 5 + 2 = 2

**Eliminar Proveedor**

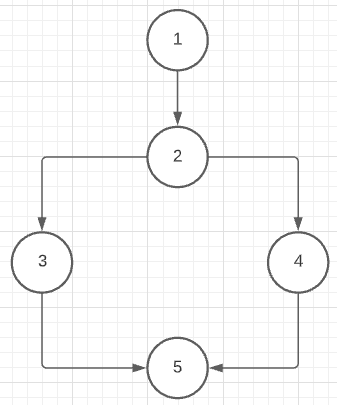
**CÓDIGO FUENTE**

| private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {  if(list.getSelectedItem().toString().equals("Seleccione")){  JOptionPane.showMessageDialog(this, "Seleccione el nombre de un proveedor!","Error en la eliminacion de proveedor!", JOptionPane.ERROR\_MESSAGE);  } else {  SupplierDAO DAO = SupplierDAO.getInstance();  DAO.deleteSupplier(list.getSelectedItem().toString());    JOptionPane.showMessageDialog(this, "Has eliminado al proveedor "+list.getSelectedItem().toString()+" correctamente!","Eliminacion de proveedor correcta", JOptionPane.INFORMATION\_MESSAGE);  loadList();  }  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 5.

**R2:** 1, 2, 4, 5.

**COMPLEJIDAD CICLOMÁTICA**

V(G)= 5 – 5 + 2 = 2

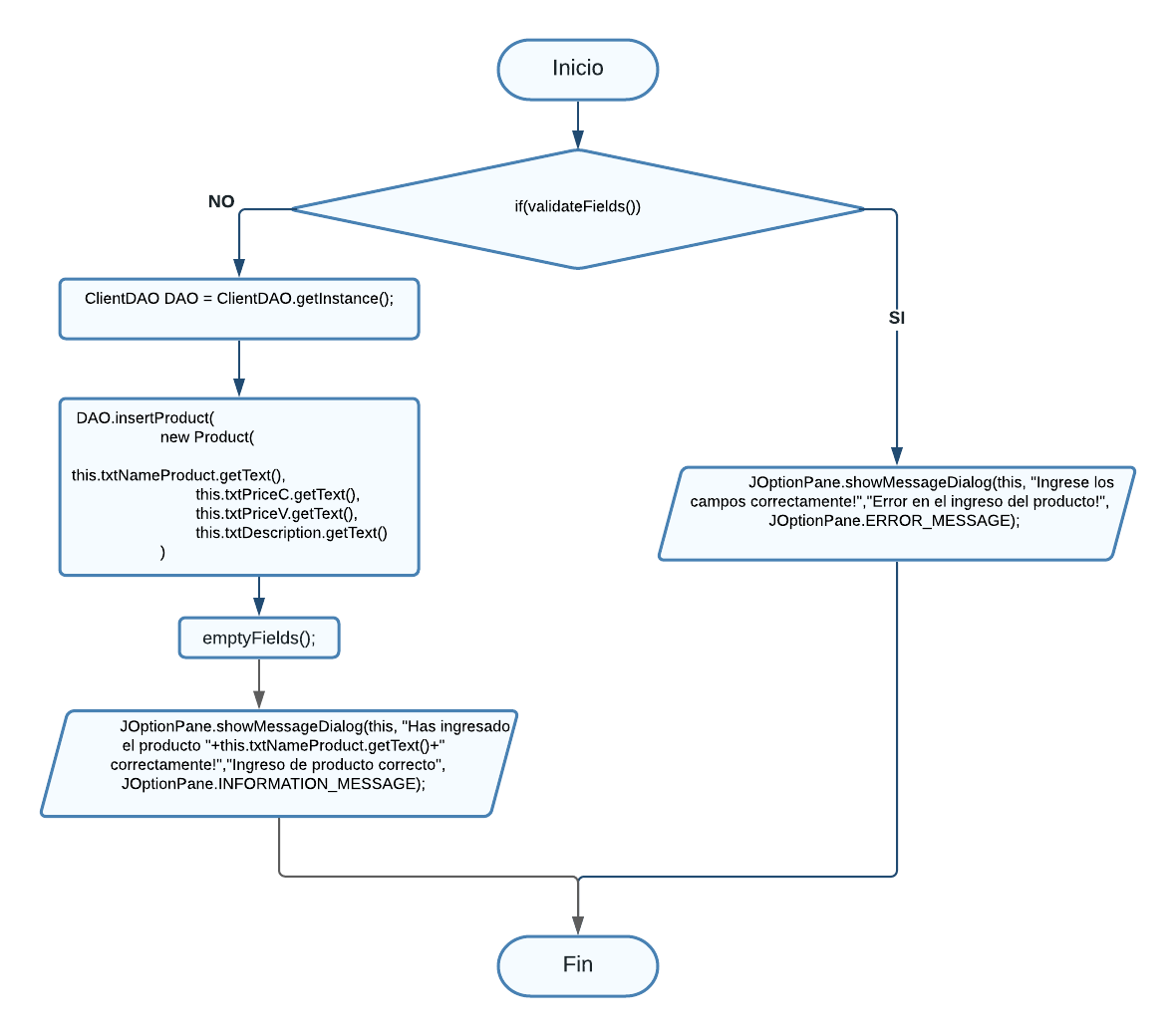
**PRODUCTO**

**Agregar producto**

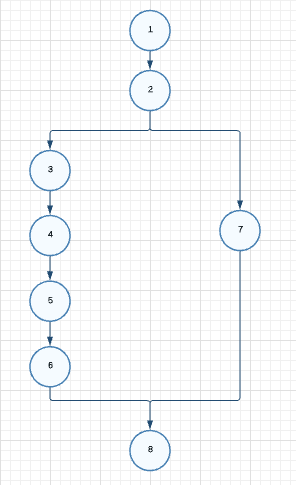
**CÓDIGO FUENTE**

| if(validateFields()){  ProductDAO DAO = ProductDAO.getInstance();  DAO.insertProduct(  new Product(  this.txtNameProduct.getText(),  this.txtPriceC.getText(),  this.txtPriceV.getText(),  this.txtDescription.getText()  )  );    emptyFields();  JOptionPane.showMessageDialog(this, "Has ingresado el producto "+this.txtNameProduct.getText()+" correctamente!","Ingreso de producto correcto", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(this, "Ingrese los campos correctamente!","Error en el ingreso del producto!", JOptionPane.ERROR\_MESSAGE);  } |
| --- |

**Diagrama de flujo**



**GRAFO**

****



**RUTAS**

**R1:** 1, 2, 3, 4, 5, 6, 8

**R2:** 1, 2, 7, 8

**COMPLEJIDAD CICLOMÁTICA**

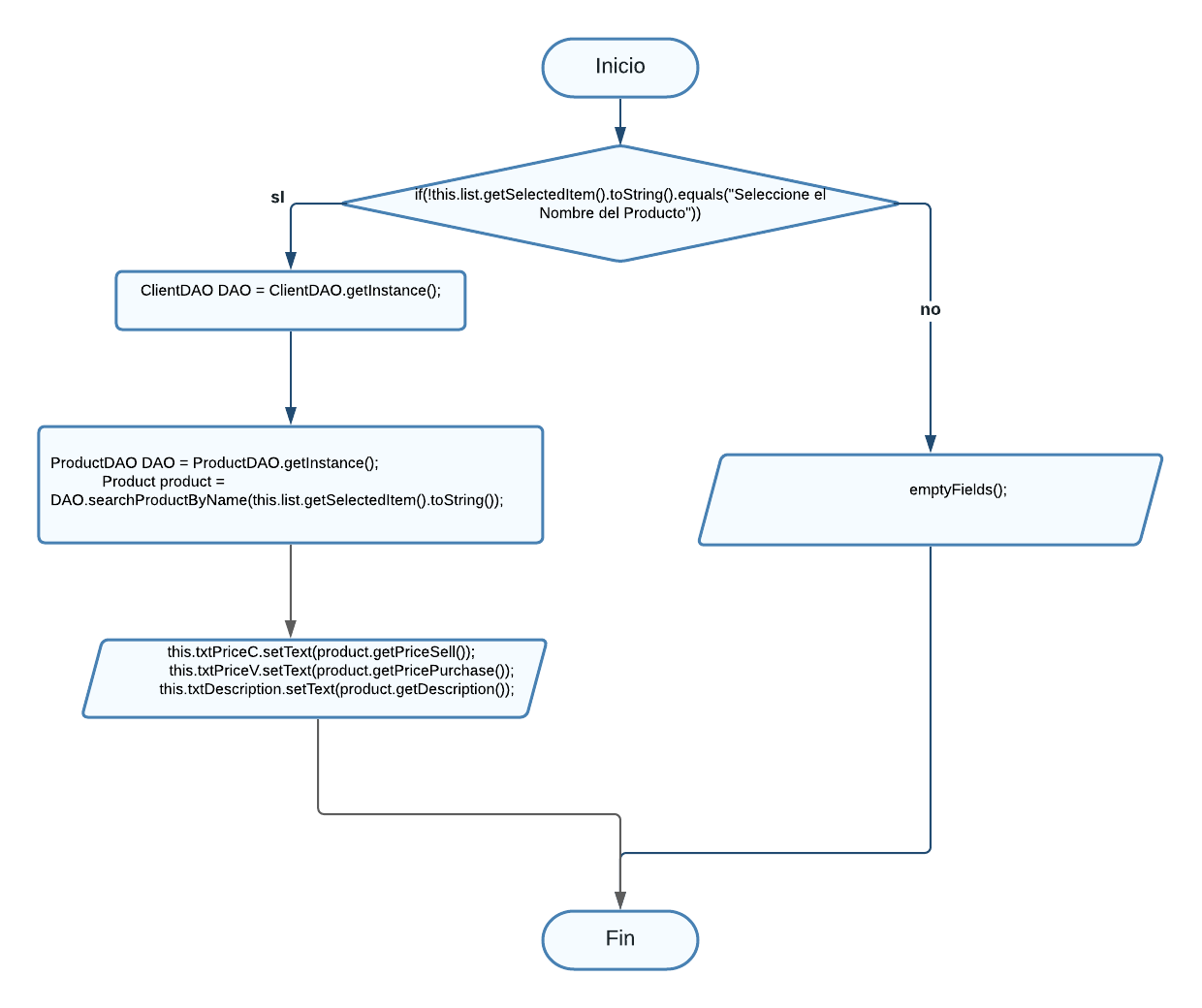
V(G)= 8 – 8 + 2 = 2

**Buscar Producto por Nombre**

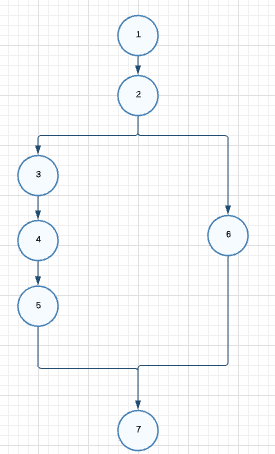
**CÓDIGO FUENTE**

| if(!this.list.getSelectedItem().toString().equals("Seleccione el Nombre del Producto")){  ProductDAO DAO = ProductDAO.getInstance();  Product product = DAO.searchProductByName(this.list.getSelectedItem().toString());  this.txtPriceC.setText(product.getPriceSell());  this.txtPriceV.setText(product.getPricePurchase());  this.txtDescription.setText(product.getDescription());  } else {  emptyFields();  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 4, 5, 7

**R2:** 1, 2, 6, 7

**COMPLEJIDAD CICLOMÁTICA**

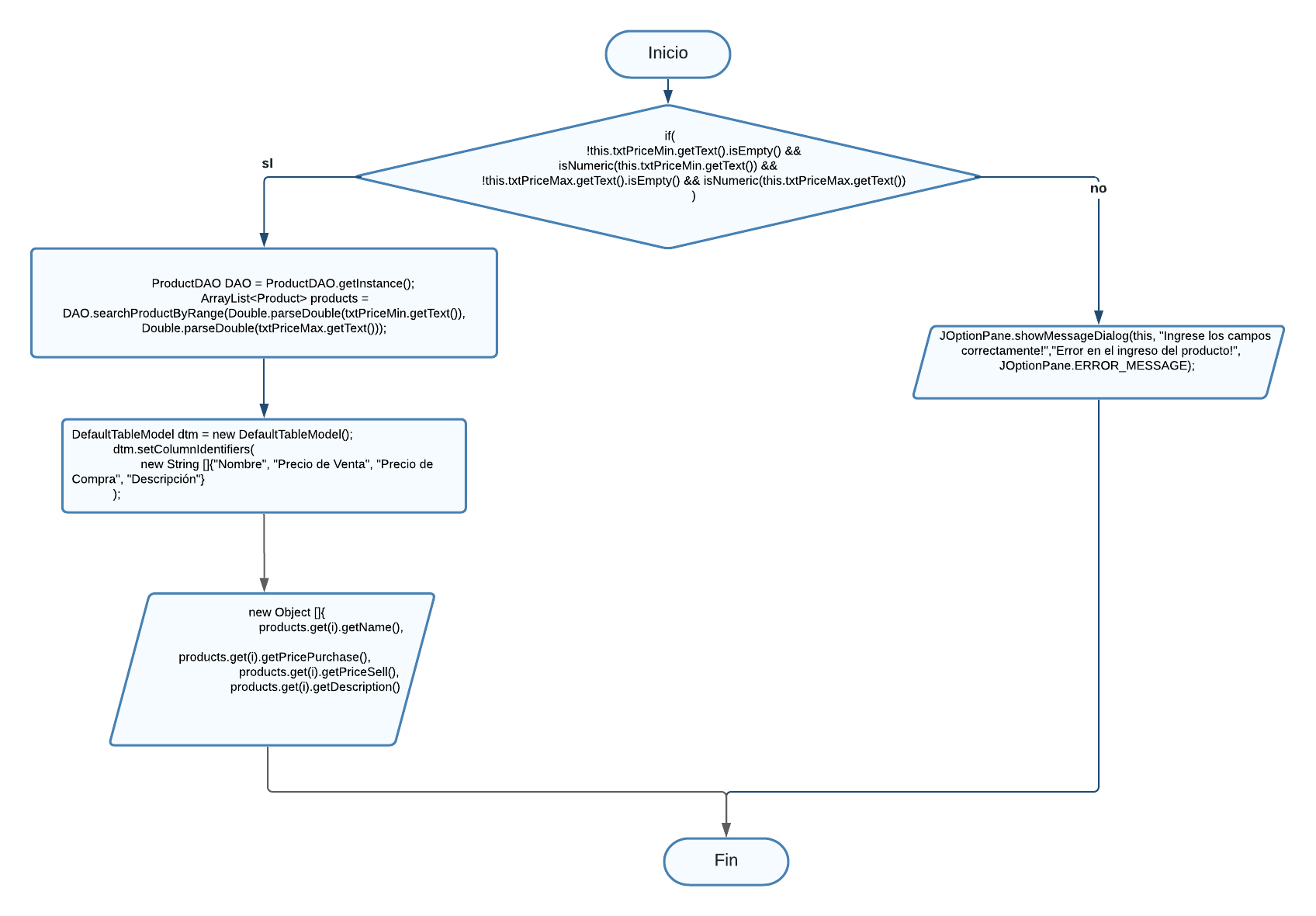
V(G)= 7 – 7 + 2 = 2

**Buscar Producto por rango de precio**

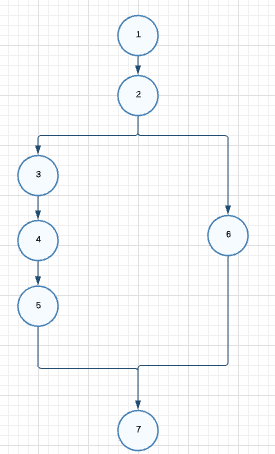
**CÓDIGO FUENTE**

| private void loadTable() {  if(  !this.txtPriceMin.getText().isEmpty() && isNumeric(this.txtPriceMin.getText()) &&  !this.txtPriceMax.getText().isEmpty() && isNumeric(this.txtPriceMax.getText())  ){    ProductDAO DAO = ProductDAO.getInstance();  ArrayList<Product> products = DAO.searchProductByRange(Double.parseDouble(txtPriceMin.getText()), Double.parseDouble(txtPriceMax.getText()));  DefaultTableModel dtm = new DefaultTableModel();  dtm.setColumnIdentifiers(  new String []{"Nombre", "Precio de Venta", "Precio de Compra", "Descripción"}  );  for (int i = 0; i < products.size(); i++) {  dtm.addRow(  new Object []{  products.get(i).getName(),  products.get(i).getPricePurchase(),  products.get(i).getPriceSell(),  products.get(i).getDescription()  }  );  }  this.table.setModel(dtm);  } else {  JOptionPane.showMessageDialog(this, "Ingrese los campos correctamente!","Error en el ingreso del producto!", JOptionPane.ERROR\_MESSAGE);  } |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 4, 5, 7

**R2:** 1, 2, 6, 7

**COMPLEJIDAD CICLOMÁTICA**

V(G)= 7 – 7 + 2 = 2

**VENTA**

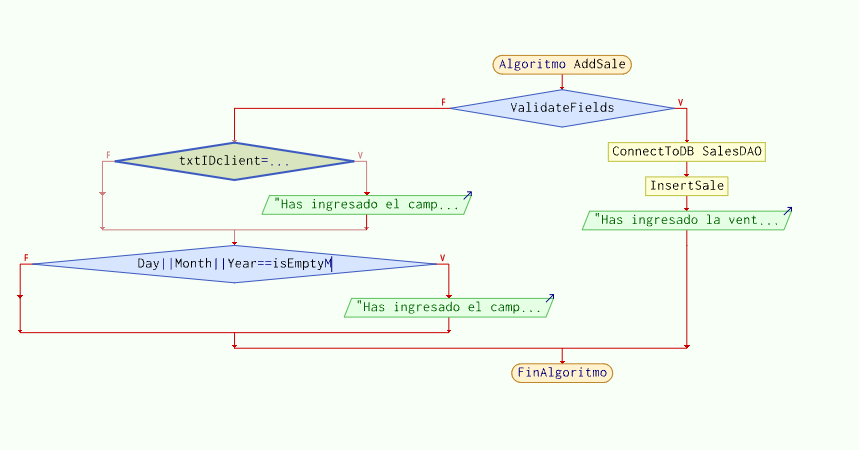
**Agregar Venta**

**CÓDIGO FUENTE**

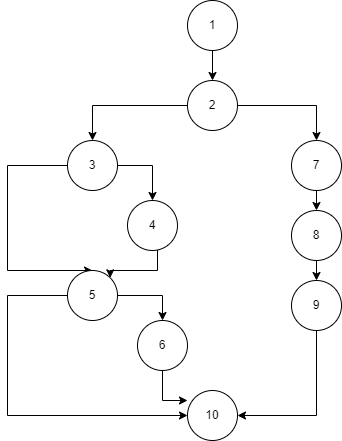
|  |
| --- |

|  |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 7, 8, 9, 10

**R2:** 1, 2, 3, 4, 5, 6, 10

**R3:** 1, 2, 3, 5, 10

**R4:** 1, 2, 3, 5, 6, 10

**R5:** 1, 2, 3, 4, 5, 10

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)=3+1=4

* V(G) = A – N + 2

V(G)= 12– 10 + 2 =4

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

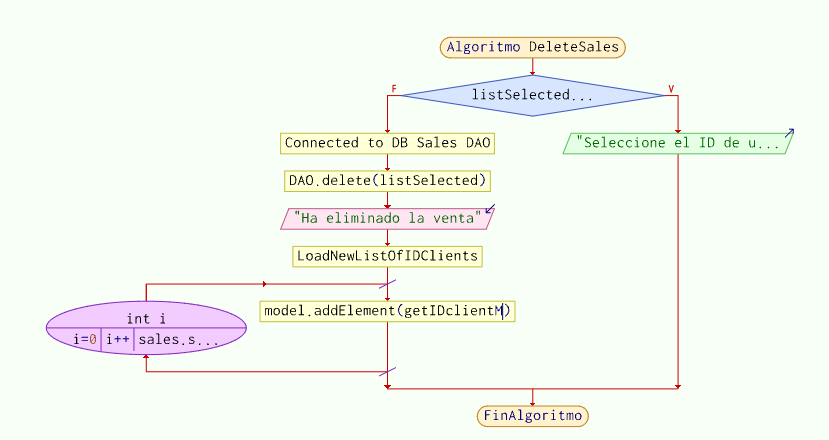
**Eliminar Venta**

**CÓDIGO FUENTE**

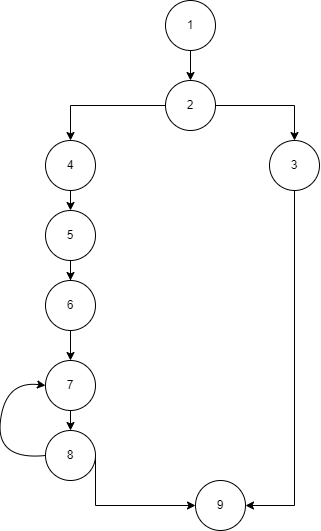
|  |
| --- |

|  |
| --- |

**DIAGRAMA DE FLUJO**

****

**GRAFO**

****

**RUTAS**

**R1:** 1, 2, 3, 9

**R2:** 1, 2, 3, 4,5,6,( 7, 8)^n, 9

**COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)=2+1=3

* V(G) = A – N + 2

V(G)= 10– 9 + 2 = 3

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos