

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 1 |
| | | Fecha: 20/02/2025 |

Prueba de Caja Blanca

“Sistema de administración de pedidos comedor Sabor Manaba”

Integrantes:

Andrade Uriel

Matias Jean

Plua Thomas

Fecha: 2025-02-09

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 2 |
| | | Fecha: 20/02/2025 |

Historia de Revisión

| Fecha | Versión | Descripción | Autores |
|-----------------|---------|---|--|
| 28/Enero/2025 | 1 | Creación del documento de caja blanca. Versión inicial | Matias Jean, Andrade Uriel, Plua Thomas. |
| 11/Febrero/2025 | 2 | Actualización de los requisitos del primer sprint y se agregan los requisitos del segundo sprint. | Matias Jean, Andrade Uriel, Plua Thomas. |
| 20/Febrero/2025 | 3 | Implementación del control de versiones | Matias Jean, Andrade Uriel, Plua Thomas. |

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 3 |
| | | Fecha: 20/02/2025 |

INDICE

| | |
|-----------------------------------|----|
| Primer requisito..... | 5 |
| 1. Código fuente | 5 |
| 2. Algoritmo | 6 |
| 3. Diagrama de flujo..... | 6 |
| 4. Grafo de flujo | 7 |
| 5. Identificación de rutas | 7 |
| 5.1. Rutas: 7 | |
| 5.2. Complejidad ciclomática 7 | |
| Segundo requisito..... | 8 |
| 1. Código fuente | 8 |
| 2. Algoritmo | 8 |
| 3. Diagrama de flujo..... | 9 |
| 4. Grafo de flujo | 9 |
| 5. Identificación de rutas | 10 |
| 5.1. Rutas: 10 | |
| 5.2. Complejidad ciclomática 10 | |
| Tercer requisito | 11 |
| 1. Código fuente | 11 |
| 2. Algoritmo | 12 |
| 3. Diagrama de flujo..... | 14 |
| 4. Grafo de flujo | 15 |
| 5. Identificación de rutas | 16 |
| 5.1. Rutas: 16 | |
| 5.2. Complejidad ciclomática 16 | |
| Cuarto requisito..... | 17 |
| 1. Código fuente | 17 |
| 2. Algoritmo | 17 |
| 3. Diagrama de flujo..... | 17 |
| 4. Grafo de flujo | 18 |
| 5. Identificación de rutas | 18 |
| 5.1. Rutas: 18 | |
| 5.2. Complejidad ciclomática 18 | |
| Quinto requisito | 19 |
| 1. Código fuente | 19 |

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 4 |
| | | Fecha: 20/02/2025 |

| | | |
|-----------------------|-------------------------------|----|
| 2. | Algoritmo | 19 |
| 3. | Diagrama de flujo..... | 20 |
| 4. | Grafo de flujo | 20 |
| 5. | Identificación de rutas | 21 |
| 5.1. | Rutas: 21 | |
| 5.2. | Complejidad ciclomática 21 | |
| Sexto requisito | | 22 |
| 1. | Código fuente | 22 |
| 2. | Algoritmo | 22 |
| 3. | Diagrama de flujo..... | 23 |
| 4. | Grafo de flujo | 23 |
| 5. | Identificación de rutas | 24 |
| 5.1. | Rutas: 24 | |
| 5.2. | Complejidad ciclomática 24 | |

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 5 |
| | | Fecha: 20/02/2025 |

Primer requisito

| | |
|--|------------------------|
| Historia de Usuario | |
| Número: REQ001 | Usuario: Alex Gonzales |
| Nombre Historia: Menú principal del sistema | |
| Prioridad: alta | |
| Programador Responsable: Uriel Adrade | |
| Descripción: Mantener un control del menú mediante las opciones, Pedidos, Clientes y Reportes | |
| Validación: Que se activen botones del evento correcto (ingresar pedido, ingresar cliente, reporte) | |

1. Código fuente

```
if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
    if (this.Opcion.getText().equals("1")) { //entrada a Pedidos
        //ENTRADA AL MENU PEDIDOS
        JOptionPane.showMessageDialog(this, "Exito, ingresando al menu pedidos", "Opcion Valida", JOptionPane.INFORMATION_MESSAGE);
        getMenuPedidos().setVisible(true);
        this.setVisible(false);
        getOpcion().setText("");
    } else if (this.Opcion.getText().equals("2")) { //entrada a Clientes
        //ENTRADA AL MENU CLIENTES
        JOptionPane.showMessageDialog(this, "Exito, ingresando al menu clientes", "Opcion Valida", JOptionPane.INFORMATION_MESSAGE);
        getMenuClientes().setVisible(true);
        this.setVisible(false);
        getOpcion().setText("");
    } else if (this.Opcion.getText().equals("3")) { //entrada a Reportes
        //ENTRADA AL MENU REPORTES
        JOptionPane.showMessageDialog(this, "Exito, ingresando al menu reportes", "Opcion Valida", JOptionPane.INFORMATION_MESSAGE);
        getMenuReporte().setVisible(true);
        this.setVisible(false);
        getOpcion().setText("");
    } else if (this.Opcion.getText().equals("4")) {
        JOptionPane.showMessageDialog(this, "Saliendo del sistema", "Salir", JOptionPane.CLOSED_OPTION);
        System.exit(0); //salir del programa
    } else {
        //MENSAJE DE OPCION NO VALIDA DEL IF
        JOptionPane.showMessageDialog(this, "Opcion no valida, ingrese una opcion entre el 1 y 4 ", "Opcion Invalida", JOptionPane.WARNING_MESSAGE);
        getOpcion().setText("");
    }
}
```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 6 |
| | | Fecha: 20/02/2025 |

2. Algoritmo

```

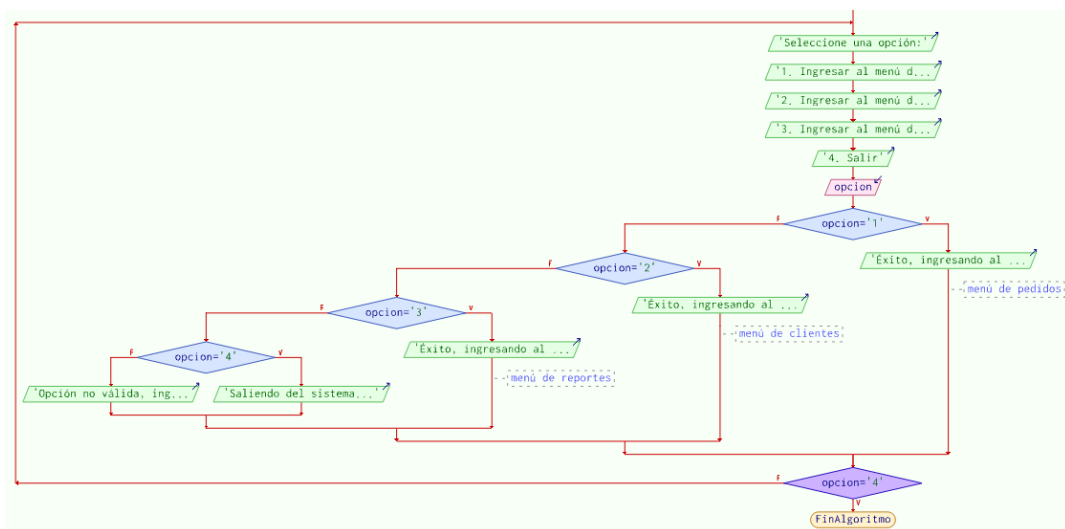
Proceso Menu_Principal
  Definir opcion Como Cadena

  Repetir
    Escribir "Seleccione una opción:"
    Escribir "1. Ingresar al menú de pedidos"
    Escribir "2. Ingresar al menú de clientes"
    Escribir "3. Ingresar al menú de reportes"
    Escribir "4. Salir"
    Leer opcion

    Si opcion = "1" Entonces
      Escribir "Éxito, ingresando al menú de pedidos"
      //menú de pedidos
    Sino
      Si opcion = "2" Entonces
        Escribir "Éxito, ingresando al menú de clientes"
        // menú de clientes
      Sino
        Si opcion = "3" Entonces
          Escribir "Éxito, ingresando al menú de reportes"
          //menú de reportes
        Sino
          Si opcion = "4" Entonces
            Escribir "Saliendo del sistema..."
          Sino
            Escribir "Opción no válida, ingrese una opción entre 1 y 4"
          FinSi
        FinSi
      FinSi
    FinSi
  Hasta Que opcion = "4"
FinProceso

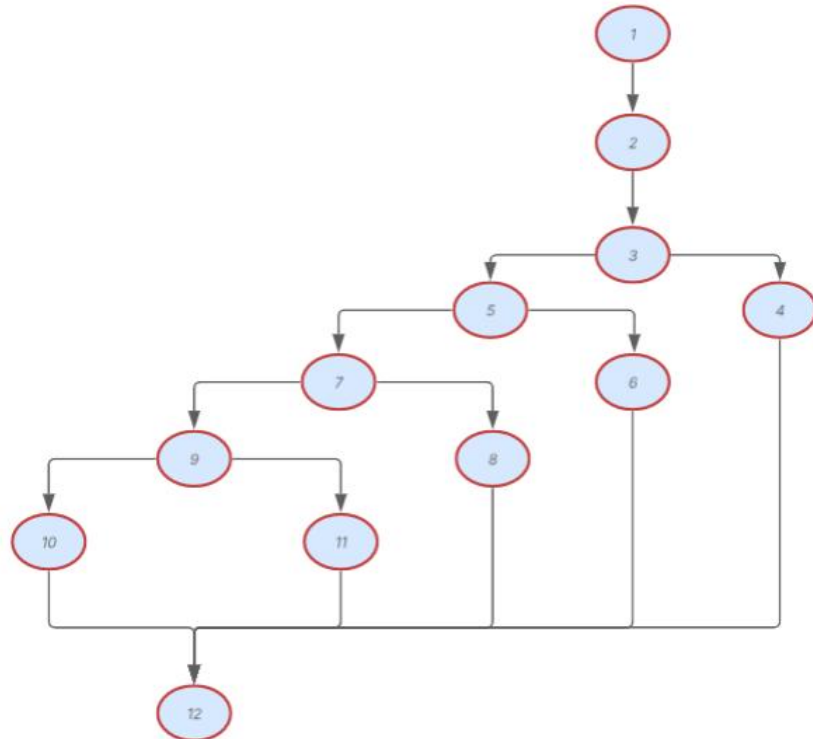
```

3. Diagrama de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 7 |
| | | Fecha: 20/02/2025 |

4. Grafo de flujo



5. Identificación de rutas

5.1. Rutas:

R1: 1,2,3,4,12

R2: 1,2,3,5,6,12

R3: 1,2,3,5,7,8,12

R4: 1,2,3,5,7,9,11,12

R5: 1,2,3,5,7,9,10,12

5.2. Complejidad ciclomática

$V(G) = \text{número de nodos predichados(decisiones)} + 1$

$V(G) = 4 + 1 = \mathbf{5}$

$V(G) = A - N + 2$

$V(G) = 15 - 12 + 2 = \mathbf{5}$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 8 |
| | | Fecha: 20/02/2025 |

Segundo requisito

| Historia de Usuario | |
|--|------------------------|
| Número: REQ002 | Usuario: Alex Gonzales |
| Nombre Historia: Ingreso de datos del pedido | |
| Prioridad: Alta | |
| Programador Responsable: Uriel Andrade | |
| Descripción: Se debe realizar el registro del pedido de algún cliente | |
| Validación: Si no se selecciona ningún platillo ofertado, si no se selecciona ningún método de pago. | |

1. Código fuente

```

public boolean validarDatos() {
    String pagoTipo = (String) this.getTipoPago().getSelectedItem();
    // Validación de que al menos un platillo ha sido seleccionado
    if (!this.platoUno.isSelected() && !this.platoDos.isSelected() && !this.platoTres.isSelected()) {
        mensaje("Por favor, debe seleccionar al menos un platillo.", "Advertencia");
        return false;
    }

    // Validación de tipo de pago
    System.out.println("Tipo de pago seleccionado: " + pagoTipo); // Depuración
    if (getOPCION_DEFECTO().equals(pagoTipo)) {
        mensaje("Por favor, seleccione un tipo de pago.", "Advertencia");
        return false;
    }

    // Mostrar mensaje de que todo es válido
    mensaje("Se han agregado con éxito los datos.", "Datos Agregados");

    // Mensaje de depuración
    System.out.println("Datos validados correctamente. Tipo de pago: " + pagoTipo);

    return true;
}

```

2. Algoritmo

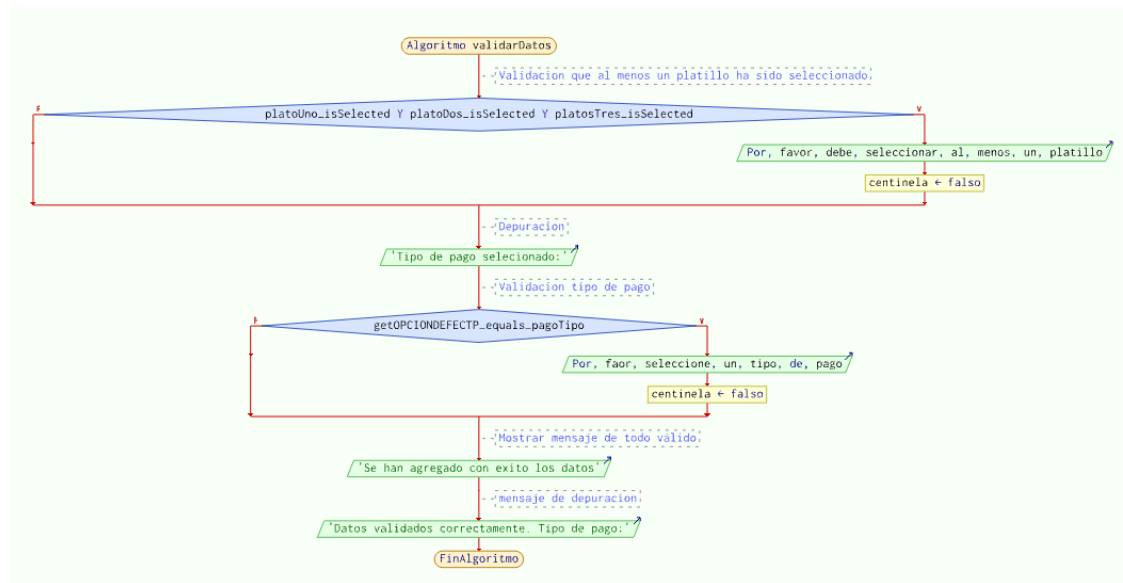
```

1 Algoritmo validarDatos
2     // Validacion que al menos un platillo ha sido seleccionado
3     Si platoUno_isSelected ^ platoDos_isSelected ^ platosTres_isSelected Entonces
4         Escribir Por favor, debe seleccionar al menos un platillo
5         centinela ← falso
6     FinSi
7     // Depuracion
8     Escribir "Tipo de pago seleccionado:"
9     // Validacion tipo de pago
10    Si getOPCIONDEFECTP_equals_pagoTipo Entonces
11        Escribir Por faor, seleccione un tipo de pago
12        centinela ← falso
13    FinSi
14    // Mostrar mensaje de todo válido
15    Escribir "Se han agregado con exito los datos"
16    // mensaje de depuracion
17    Escribir "Datos validados correctamente. Tipo de pago:"
18 FinAlgoritmo
19

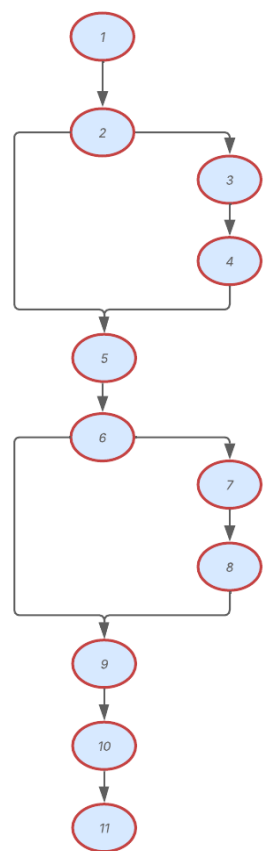
```


| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 9 |
| | | Fecha: 20/02/2025 |

3. Diagrama de flujo



4. Grafo de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 10 |
| | | Fecha: 20/02/2025 |

5. Identificación de rutas

5.1.Rutas:

R1: 1,2,3,4,5,6,7,8,9,10,11

R2: 1,2,5,6,7,8,9,10,11

R3: 1,2,3,4,5,6,9,10,11

5.2.Complejidad ciclomática

$V(G) = \text{número de nodos predichados(decisiones)} + 1$

$V(G) = 2 + 1 = \mathbf{3}$

$V(G) = A - N + 2$

$V(G) = 12 - 11 + 2 = \mathbf{3}$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 11 |
| | | Fecha: 20/02/2025 |

Tercer requisito

| | |
|---|------------------------|
| Historia de Usuario | |
| Número: REQ003 | Usuario: Alex Gonzales |
| Nombre Historia: Ingreso de datos del cliente | |
| Prioridad: Alta | |
| Programador Responsable: Uriel Andrade | |
| Descripción: Se debe realizar el registro del pedido de algún cliente | |
| Validación: Si no se selecciona ningún platillo ofertado, si no se selecciona ningún método de pago. | |

1. Código fuente

```
if (nombre.isEmpty()) {
    mostrarError("Por favor, complete el nombre del cliente.");
    return false;
}
if (!nombre.matches("^[\\p{L}\\s]+$")) {
    mostrarError("El nombre solo debe contener letras.");
    return false;
}

// VALIDACIÓN DEL APELLIDO: no vacío y solo letras
if (apellido.isEmpty()) {
    mostrarError("Por favor, complete el apellido del cliente.");
    return false;
}
if (!apellido.matches("^[\\p{L}\\s]+$")) {
    mostrarError("El apellido solo debe contener letras.");
    return false;
}

// VALIDACIÓN DEL TIPO DE IDENTIFICACIÓN
if (getOPCION_DEFEECTO().equals(tipoIdentificacion)) {
    mostrarError("Por favor, seleccione un tipo de identificación válido.");
    return false;
}

// VALIDACIÓN DE LA IDENTIFICACIÓN SEGÚN EL TIPO
if (tipoIdentificacion.equals("Cedula")) {
    // La cédula debe tener 10 dígitos y ser válida
    if (!identificacion.matches("\\d{10}")) {
        mostrarError("La cédula debe tener 10 dígitos.");
        return false;
    }
    if (!validarCedula(identificacion)) {
        mostrarError("La cédula no es válida.");
        return false;
    }
} else if (tipoIdentificacion.equals("RUC")) {
    // El RUC debe tener 13 dígitos
    if (!identificacion.matches("\\d{13}")) {
```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 12 |
| | | Fecha: 20/02/2025 |

```

        mostrarError("El RUC debe tener 13 dígitos.");
        return false;
    }

    // Para RUC de persona natural, los primeros 10 dígitos deben ser una cédula válida y los últimos
    String cedulaPart = identificacion.substring(0, 10);
    String sufixo = identificacion.substring(10);
    if (!validarCedula(cedulaPart) || !sufijo.equals("001")) {
        mostrarError("El RUC no es válido.");
        return false;
    }
} else if (tipoIdentificacion.equals("Pasaporte")) {
    // Por ejemplo, se puede exigir que el pasaporte sea alfanumérico y tenga entre 6 y 9 caracteres
    if (!identificacion.matches("[A-Za-z0-9]{6,9}")) {
        mostrarError("El pasaporte debe ser alfanumérico y tener entre 6 y 9 caracteres.");
        return false;
    }
}

// VALIDACIÓN DE LA DIRECCIÓN: no vacía
if (direccion.isEmpty()) {
    mostrarError("Por favor, complete la dirección del cliente.");
    return false;
}

// VALIDACIÓN DEL TELÉFONO: no vacío y debe ser un número válido de Ecuador (ej. 09xxxxxxx, 10 dígitos)
if (telefono.isEmpty()) {
    mostrarError("Por favor, complete el teléfono del cliente.");
    return false;
}
if (!telefono.matches("^09\\d{8}$")) {
    mostrarError("Por favor, ingrese un teléfono válido de Ecuador (debe comenzar con '09' y tener 10 dígitos).");
    return false;
}

// VALIDACIÓN DEL CORREO: no vacío y formato válido
if (correo.isEmpty()) {
    mostrarError("Por favor, complete el correo del cliente.");
    return false;
}
if (!correo.matches("^[\\w.-]+@[\\w.-]+\\.([a-zA-Z]{2,})$")) {
    mostrarError("Por favor, ingrese un correo válido.");
    return false;
}
}

//</editor-fold>

return true;

```

2. Algoritmo

```

1  Algoritmo validar_Campos
2      // validacion de nombre
3      Si nombre_isEmpty = V Entonces
4          Escribir 'Por favor, complete el nombre del cliente.'
5          centinela ← Falso
6      FinSi
7      Si nombre_matches ≠ V Entonces
8          Escribir 'El nombre solo debe contener letras.'
9      FinSi
10
11     // validacion de apellido
12     Si apellido_isEmpty = V Entonces
13         Escribir 'Por favor, complete el apellido del cliente.'
14         centinela ← Falso
15     FinSi
16     Si apellido_matches ≠ V Entonces
17         Escribir 'El nombre solo debe contener letras.'
18     FinSi
19
20     // validacion tipo de identificacion (combo_box)
21     Si tipoidentificación_DEFECTO = V Entonces
22         Escribir 'Por favor, seleccione un tipo de identificación válido.'
23         centinela ← Falso
24     FinSi
25
26     //validacionn de los tipos de identificacion
27     Si tipoIdentificacion = "Cedula" Entonces
28         // Validación de que la cédula tenga 10 dígitos
29         Si cedula_matches_digitos = F Entonces
30             Escribir 'La cédula debe tener 10 dígitos.'
31             centinela ← Falso
32         FinSi
33

```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 13 |
| | | Fecha: 20/02/2025 |

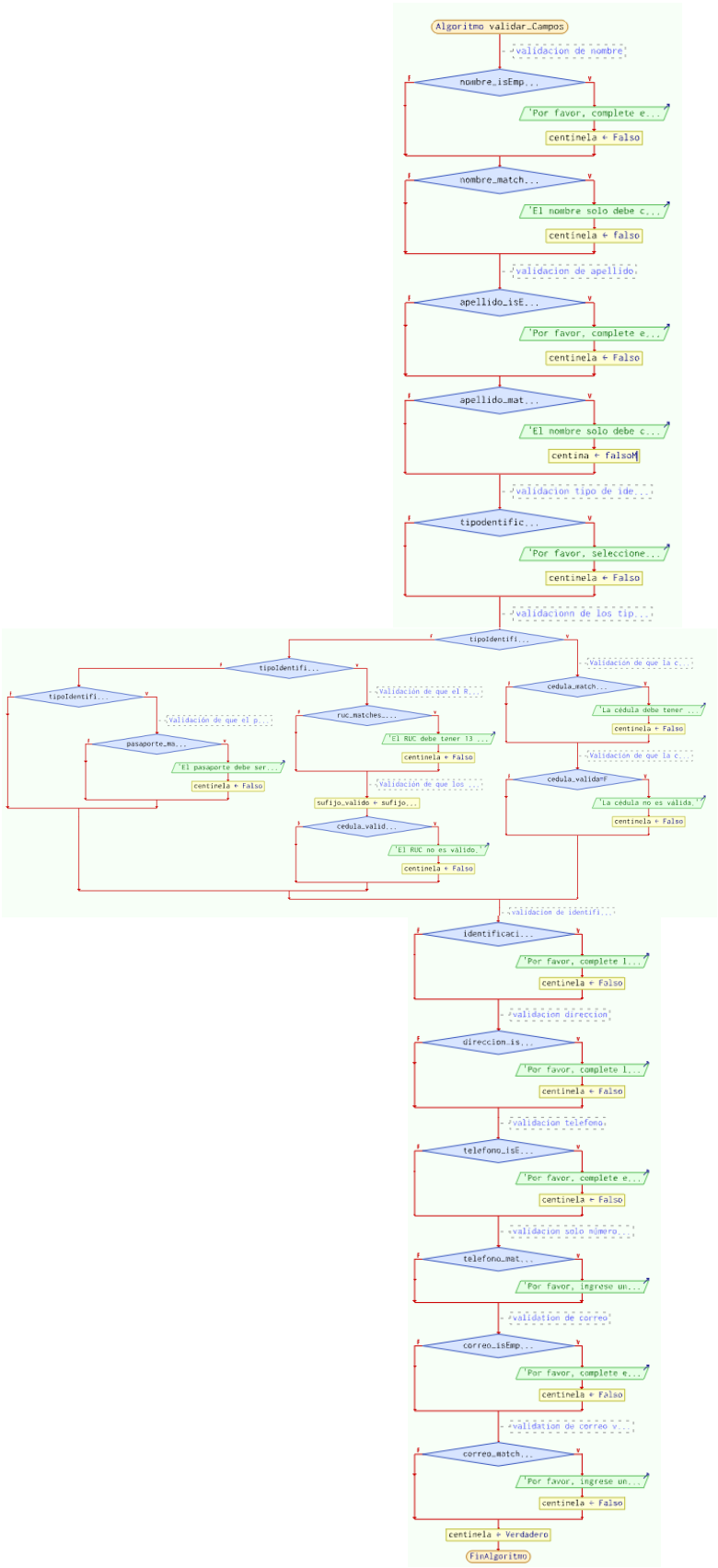
```

34      // Validación de que la cédula sea válida
35      Si cedula_valida = F Entonces
36          Escribir 'La cédula no es válida.'
37          centinela ← Falso
38      FinSi
39      Sino Si tipoIdentificacion = "RUC" Entonces
40          // Validación de que el RUC tenga 13 dígitos
41
42          Si ruc_matches_digitos = F Entonces
43              Escribir 'El RUC debe tener 13 dígitos.'
44              centinela ← Falso
45          FinSi
46
47          // Validación de que los primeros 10 dígitos sean una cédula válida y los últimos 3 sean "001"
48
49          sufijo_valido ← sufijo = "001"
50          Si cedula_valida = F O sufijo_valido = F Entonces
51              Escribir 'El RUC no es válido.'
52              centinela ← Falso
53          FinSi
54
55          Sino Si tipoIdentificacion = "Pasaporte" Entonces
56              // Validación de que el pasaporte sea alfanumérico y tenga entre 6 y 9 caracteres
57              Si pasaporte_matches_formato = F Entonces
58                  Escribir 'El pasaporte debe ser alfanumérico y tener entre 6 y 9 caracteres.'
59                  centinela ← Falso
60              FinSi
61          finsi
62      FinSi
63  FinSi
64
65  // validacion de identificacion
66  Si identificacion_isEmpty = V Entonces
67      Escribir 'Por favor, complete la identificación del cliente.'
68      centinela ← Falso
69  FinSi
70
71  // validacion direccion
72  Si direccion_isEmpty = V Entonces
73      Escribir 'Por favor, complete la dirección del cliente.'
74      centinela ← Falso
75  FinSi
76
77
78  // validacion telefono
79  Si telefono_isEmpty = V Entonces
80      Escribir 'Por favor, complete el telefono del cliente.'
81      centinela ← Falso
82  FinSi
83
84  // validacion solo números telefono y 10 digitos
85  Si telefono_matches ≠ V Entonces
86      Escribir 'Por favor, ingrese un teléfono válido de Ecuador (debe comenzar con 09 y tener 10 dígitos).'
87  FinSi
88
89  // validation de correo
90  Si correo_isEmpty = V Entonces
91      Escribir 'Por favor, complete el correo del cliente.'
92      centinela ← Falso
93  FinSi
94
95  // validation de correo valido
96  Si correo_matches = V Entonces
97      Escribir 'Por favor, ingrese un correo válido.'
98      centinela ← Falso
99  FinSi
100
101  centinela ← Verdadero
102  FinAlgoritmo

```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 14 |
| | | Fecha: 20/02/2025 |

3. Diagrama de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 16 |
| | | Fecha: 20/02/2025 |

5. Identificación de rutas

5.1.Rutas:

R1: 1,2,3,4,51
R2: 1,2,5,6,7,51
R3: 1,2,5,8,9,10,51
R4: 1,2,5,8,11,12,13,51
R5: 1,2,5,8,11,14,15,16,51
R6: 1,2,5,8,11,14,17,20,21,22,51
R7: 1,2,5,8,11,14,17,20,23,24,25,51
R8: 1,2,5,8,11,14,17,18,19,35,...,51
R9: 1,2,5,8,11,14,17,18,26,27,28,51
R10: 1,2,5,8,11,14,17,20,23,35,36,37,51
R11: 1,2,5,8,11,14,17,18,26,29,35,...,51
R12: 1,2,5,8,11,14,17,18,19,32,35,...,51
R13: 1,2,5,8,11,14,17,18,19,32,33,34,51
R14: 1,2,5,8,11,14,17,18,26,29,30,31,51
R15: 1,2,5,8,11,14,17,20,23,35,38,39,40,51
R16: 1,2,5,8,11,14,17,20,23,35,38,41,42,43,51
R17: 1,2,5,8,11,14,17,20,23,35,38,41,44,47,50,51
R18: 1,2,5,8,11,14,17,20,23,35,38,41,44,45,46,51
R19: 1,2,5,8,11,14,17,20,23,35,38,41,44,47,48,49,51

5.2.Complejidad ciclomática

$V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 18 + 1 = \mathbf{19}$

$V(G) = A - N + 2$
 $V(G) = 68 - 51 + 2 = \mathbf{19}$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 17 |
| | | Fecha: 20/02/2025 |

Cuarto requisito

| Historia de Usuario | |
|---|------------------------|
| Número: REQ004 | Usuario: Alex Gonzales |
| Nombre Historia: Emisión de reportes | |
| Prioridad: Alta | |
| Programador Responsable: Uriel Andrade | |
| Descripción: Se deben mostrar los datos ya registrados | |
| Validación: Si no se han ingresado los datos de pedido y cliente con anterioridad. | |

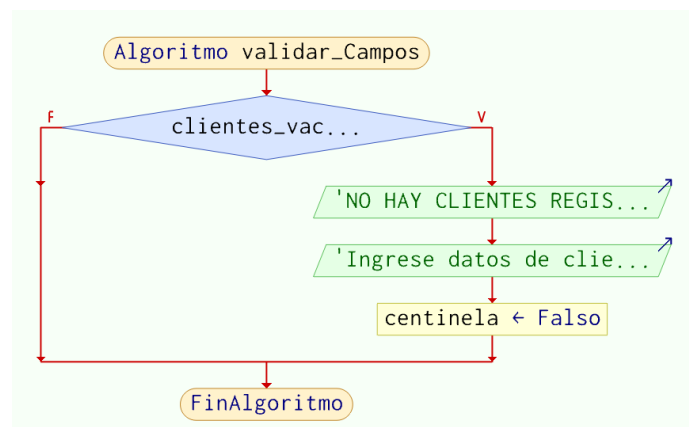
1. Código fuente

```
if (clientes.isEmpty()) {
    getJTextArea().setText("No hay clientes registrados.".toUpperCase());
    JOptionPane.showMessageDialog(this, "Ingrese datos de cliente y pedido.", "Advertencia", JOptionPane.ERROR_MESSAGE);
    return;
}
```

2. Algoritmo

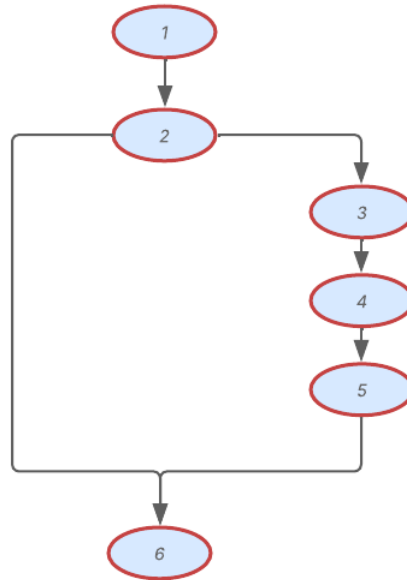
```
1  Algoritmo validar_Campos
2  Si clientes_vacio = V Entonces
3      escribir "NO HAY CLIENTES REGISTRADOS."
4      escribir 'Ingrese datos de cliente y pedido'
5      centinela ← Falso
6  FinSi
7  FinAlgoritmo
```

3. Diagrama de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 18 |
| | | Fecha: 20/02/2025 |

4. Grafo de flujo



5. Identificación de rutas

5.1.Rutas:

R1: 1,2,6

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

5.2.Complejidad ciclomática

$V(G) = \text{número de nodos predicados(decisiones)} + 1$

$V(G) = 1 + 1 = \mathbf{2}$

$V(G) = A - N + 2$

$V(G) = 6 - 6 + 2 = \mathbf{2}$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 19 |
| | | Fecha: 20/02/2025 |

Quinto requisito

| | |
|--|-------------------------------|
| Historia de Usuario | |
| Número: REQ005 | Usuario: Alex Gonzales |
| Nombre Historia: Eliminación de registro | |
| Prioridad: Alta | |
| Programador Responsable: Jean Matias | |
| Descripción: Se debe eliminar los reportes ya ingresados | |
| Validación: Confirmación de reporte eliminado | |

1. Código fuente

```
private void eliminarPedido() {
    int selectedRow = tablaPedidos.getSelectedRow();
    if (selectedRow != -1) {
        listaPedidos.remove(selectedRow); // Eliminar el pedido de la lista
        cargarPedidos(); // Recargar los pedidos en la tabla

        // Reflejar los cambios en MenuPedidos
        menuPedidos.actualizarDatosPedidos();

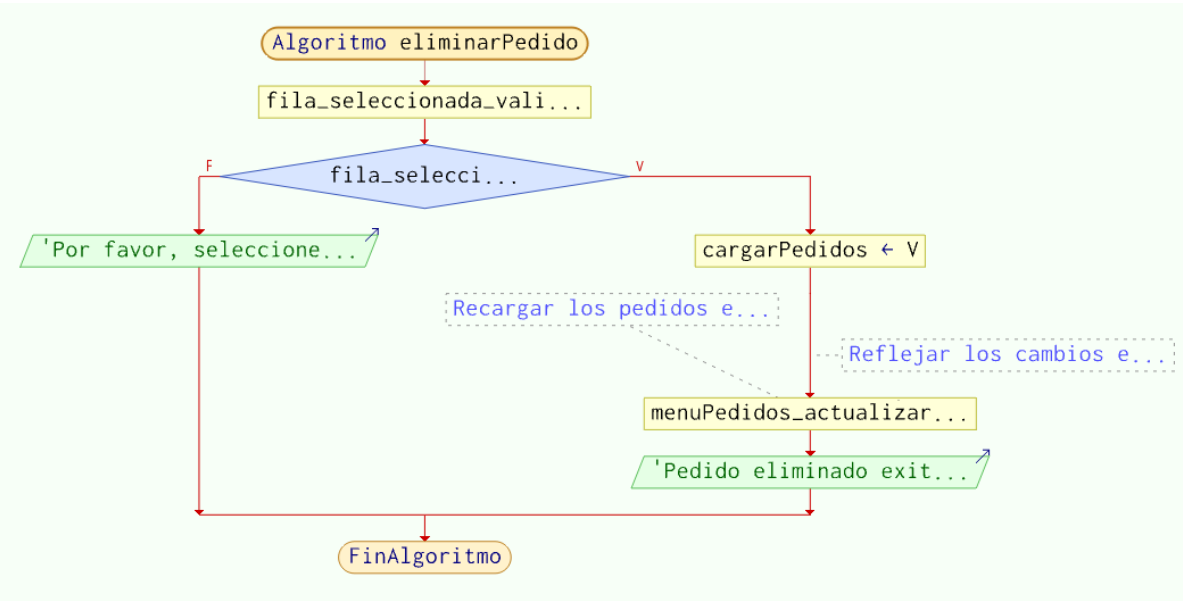
        JOptionPane.showMessageDialog(this, "Pedido eliminado exitosamente.", "Información", JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(this, "Por favor, seleccione un pedido para eliminar.", "Advertencia", JOptionPane.WARNING_MESSAGE);
    }
}
```

2. Algoritmo

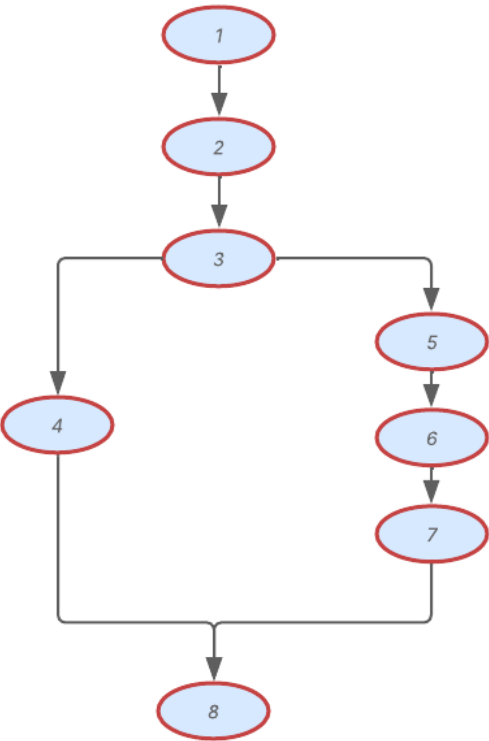
```
1  Algoritmo eliminarPedido
2  fila_seleccionada_valida ← selectedRow ≠ -1
3
4  Si fila_seleccionada_valida = V Entonces
5      cargarPedidos = V // Recargar los pedidos en la tabla
6
7      // Reflejar los cambios en MenuPedidos
8      menuPedidos_actualizarDatosPedidos = V
9
10     Escribir 'Pedido eliminado exitosamente.'
11 Sino
12     escribir 'Por favor, seleccione un pedido para eliminar'
13 FinSi
14 FinAlgoritmo
```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 20 |
| | | Fecha: 20/02/2025 |

3. Diagrama de flujo



4. Grafo de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 21 |
| | | Fecha: 20/02/2025 |

5. Identificación de rutas

5.1.Rutas:

R1: 1,2,3,4,8

R2: 1,2,3,5,6,7,8

5.2.Complejidad ciclomática

$V(G) = \text{número de nodos predichados (decisiones)} + 1$

$V(G) = 1 + 1 = \mathbf{2}$

$V(G) = A - N + 2$

$V(G) = 8 - 8 + 2 = \mathbf{2}$

DONDE:

P: Número de nodos predichados

A: Número de aristas

N: Número de nodos

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 22 |
| | | Fecha: 20/02/2025 |

Sexto requisito

| | |
|---|-------------------------------|
| Historia de Usuario | |
| Número: REQ006 | Usuario: Alex Gonzales |
| Nombre Historia: Modificación de registro | |
| Prioridad: Alta | |
| Programador Responsable: Thomas Plua | |
| Descripción: Se deben mostrar los datos ya registrados | |
| Validación: Se visualiza en la tabla que los datos han sido modificados | |

1. Código fuente

```
if (opcion == JOptionPane.YES_OPTION) {
    // Borrar datos de la tabla de pedidos asignándole un modelo vacío
    PedidoTableModel emptyPedidoModel = new PedidoTableModel(new ArrayList<>(), platos);
    tablaPedidos.setModel(emptyPedidoModel);

    // Borrar datos de la tabla de clientes asignándole un modelo vacío
    ClienteTableModel emptyClienteModel = new ClienteTableModel(new ArrayList<>());
    tablaClientes.setModel(emptyClienteModel);

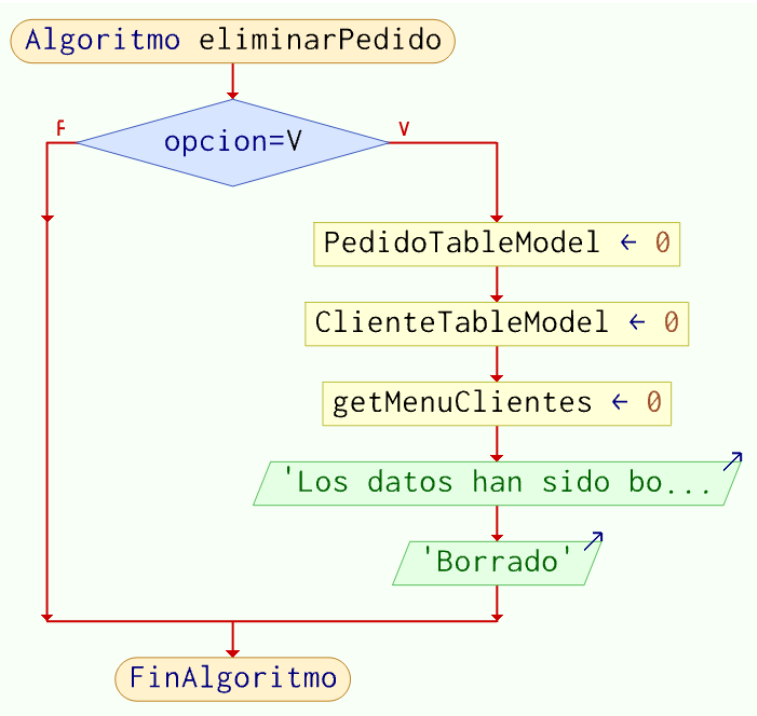
    // Si además quieres borrar los datos de las listas subyacentes, puedes hacerlo:
    getMenuClientes().getClientes().clear();
    // Y para los pedidos, si tienes una lista, también:
    listaPedidos.clear();
    JOptionPane.showMessageDialog(this,
        "Los datos han sido borrados.",
        "Borrado",
        JOptionPane.INFORMATION_MESSAGE);
}
```

2. Algoritmo

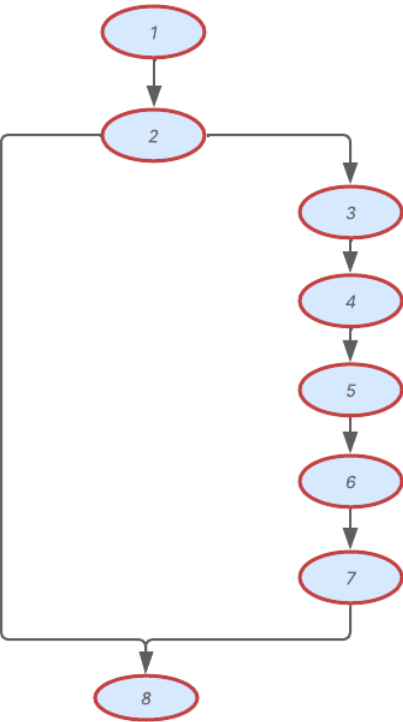
```
1  Algoritmo eliminarPedido
2      Si opcion = V Entonces
3          PedidoTableModel = 0
4          ClienteTableModel = 0
5          getMenuClientes = 0
6          escribir 'Los datos han sido borrados.'
7          escribir 'Borrado'
8      FinSi
9  FinAlgoritmo
```

| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 23 |
| | | Fecha: 20/02/2025 |

3. Diagrama de flujo



4. Grafo de flujo



| | | |
|---|-----------------------|-------------------|
| Sistema de administración de pedidos comedor “Sabor Manaba” | Prueba de caja blanca | Versión: 2 |
| | | Página: 24 |
| | | Fecha: 20/02/2025 |

5. Identificación de rutas

5.1.Rutas:

R1: 1,2,8

R2: 1,2,3,4,5,6,7,8

5.2.Complejidad ciclomática

$V(G) = \text{número de nodos predichados(decisiones)} + 1$

$V(G) = 1 + 1 = \mathbf{2}$

$V(G) = A - N + 2$

$V(G) = 8 - 8 + 2 = \mathbf{2}$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos