

Prueba de Caja Blanca

“Control de Inventario para Local de Ventas”

Integrantes:

Samuel Dávalos

Micaela Jácome

Daniel Palacios

Ezequiel Troya

Fecha 2025-01-09

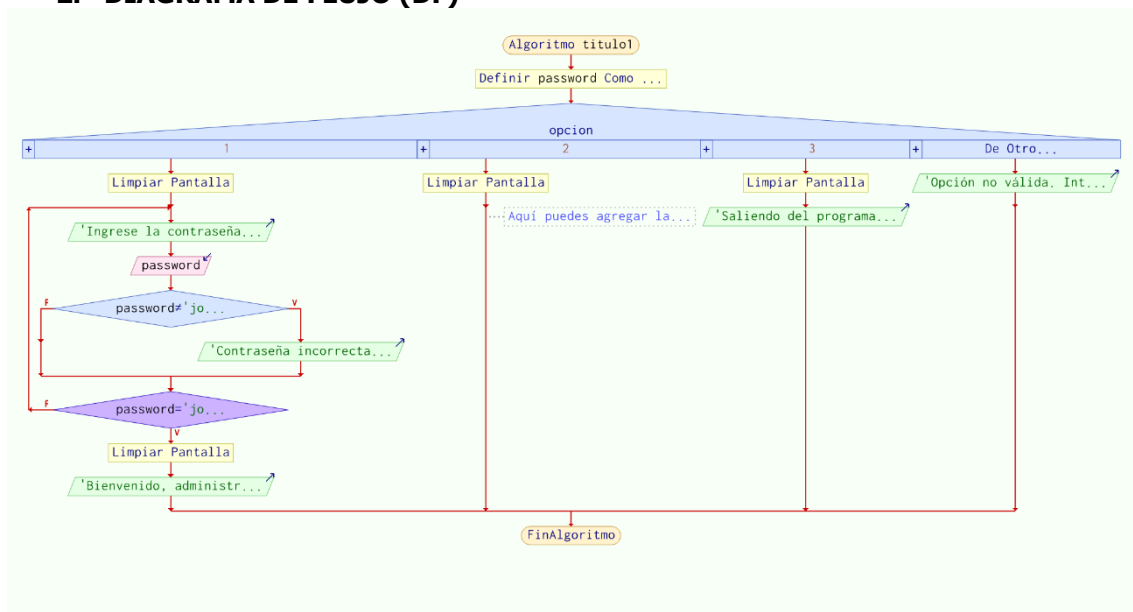
Prueba caja blanca

REQ 001 ACCESO AL SISTEMA

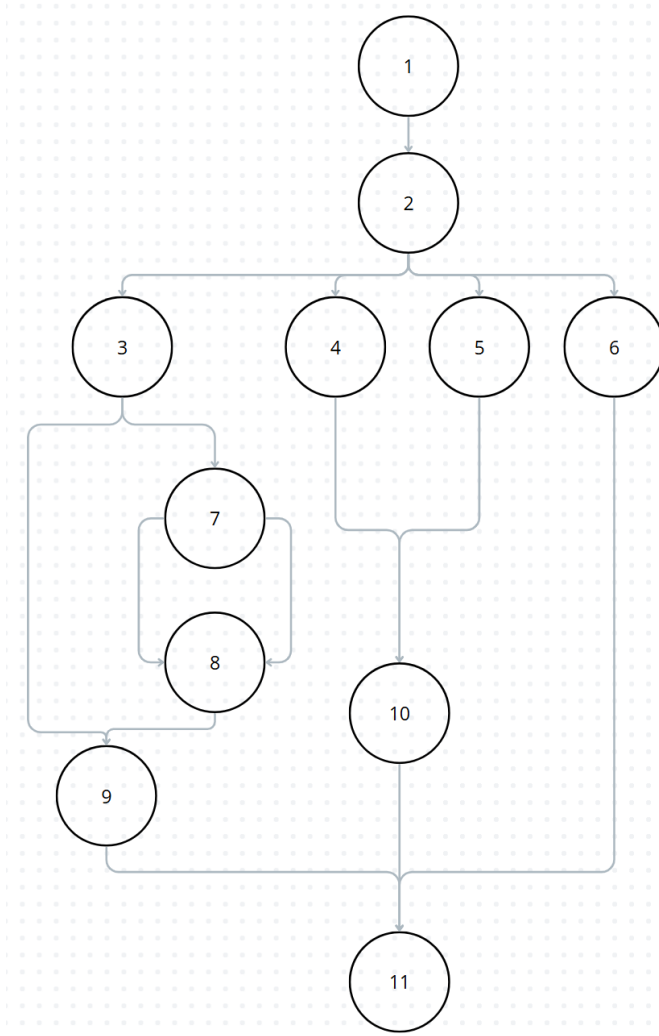
1. CÓDIGO FUENTE

```
void opcion_principal (int opcion) {  
    string password;  
    switch (opcion) {  
        case 1:  
            system("cls");  
            do {  
                cout << "Ingrese la contraseña de administrador: ";  
                cin >> password;  
                if (password != "joker2019") {  
                    cout << "Contraseña incorrecta. Intentalo de nuevo." << endl;  
                }  
            } while (password != "joker2019");  
            system("cls");  
            cout << "Bienvenido, administrador." << endl;  
  
            break;  
  
        case 2:  
            system("cls");  
  
            break;  
  
        case 3:  
            system("cls");  
            cout << "Saliendo del programa..." << endl;  
            break;  
  
        default:  
            cout << "Opcion no valida. Intente de nuevo." << endl;  
            break;  
    }  
}
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

RUTAS

R1: 1 → 2 → 5 → 10 → 11

R2: 1 → 2 → 6 → 11

R3: 1 → 2 → 3 → 7 → 8 → 9 → 11

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$
 $V(G) = 12 - 11 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

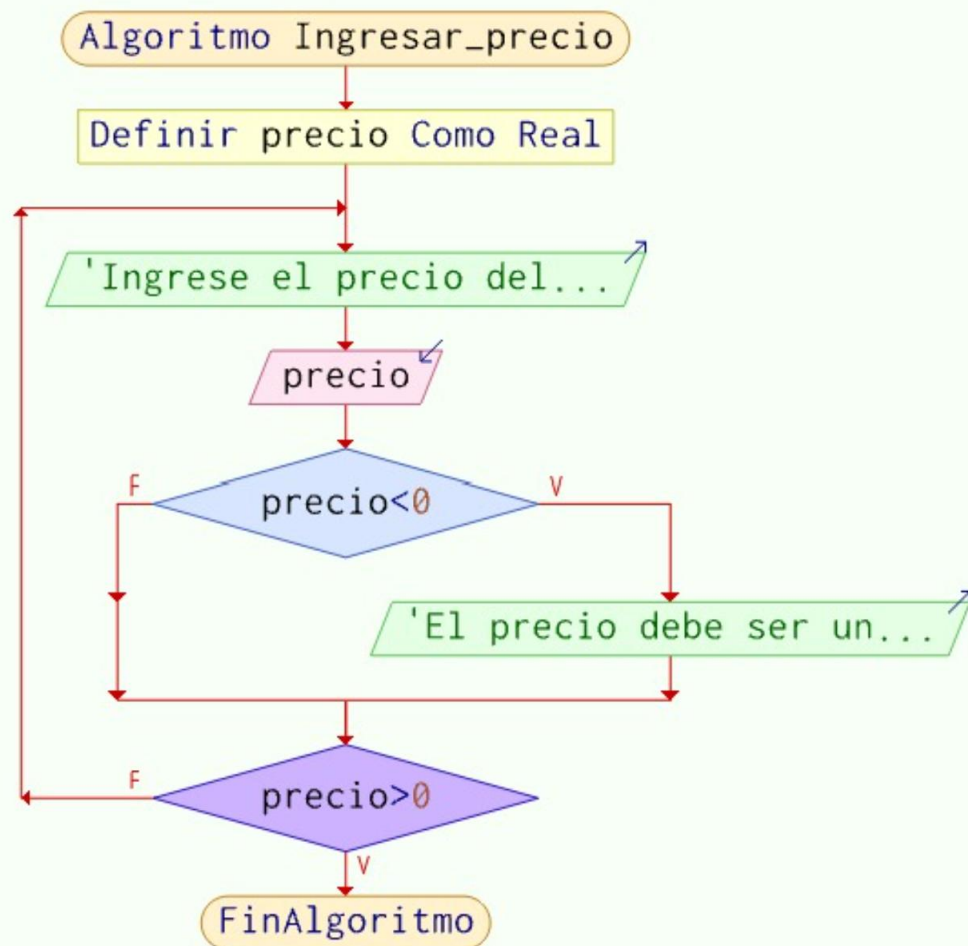
N: Número de nodos

Prueba caja blanca de REQ 002 CONTROL DE LOS PRODUCTOS CON UN SISTEMA CRUD

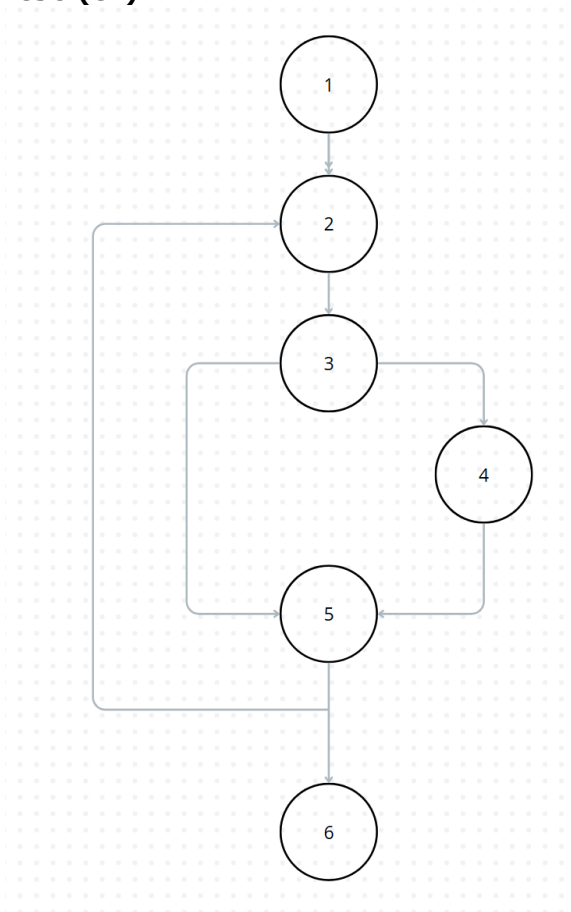
6. CÓDIGO FUENTE

```
// funcion para ingresar el precio del producto
float ingresarPrecio() {
    float precio;
    while (true) {
        cout << "Ingrese el precio del producto: ";
        cin >> precio;
        if (precio > 0) {
            break;
        } else {
            cout << "El precio debe ser un valor positivo. Intente de nuevo." << endl;
        }
    }
    return precio;
}
```

7. DIAGRAMA DE FLUJO (DF)



8. GRAFO DE FLUJO (GF)



9. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

RUTAS

R1: 1 → 2 → 3 → 5 → 6

R2: 1 → 2 → 3 → 4 → 5 → 6

R3: 1 → 2 → 3 → 4 → 5 → 2

10. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$
 $V(G) = 7 - 6 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

REQ 003 VENDER PRODUCTOS DEL SISTEMA

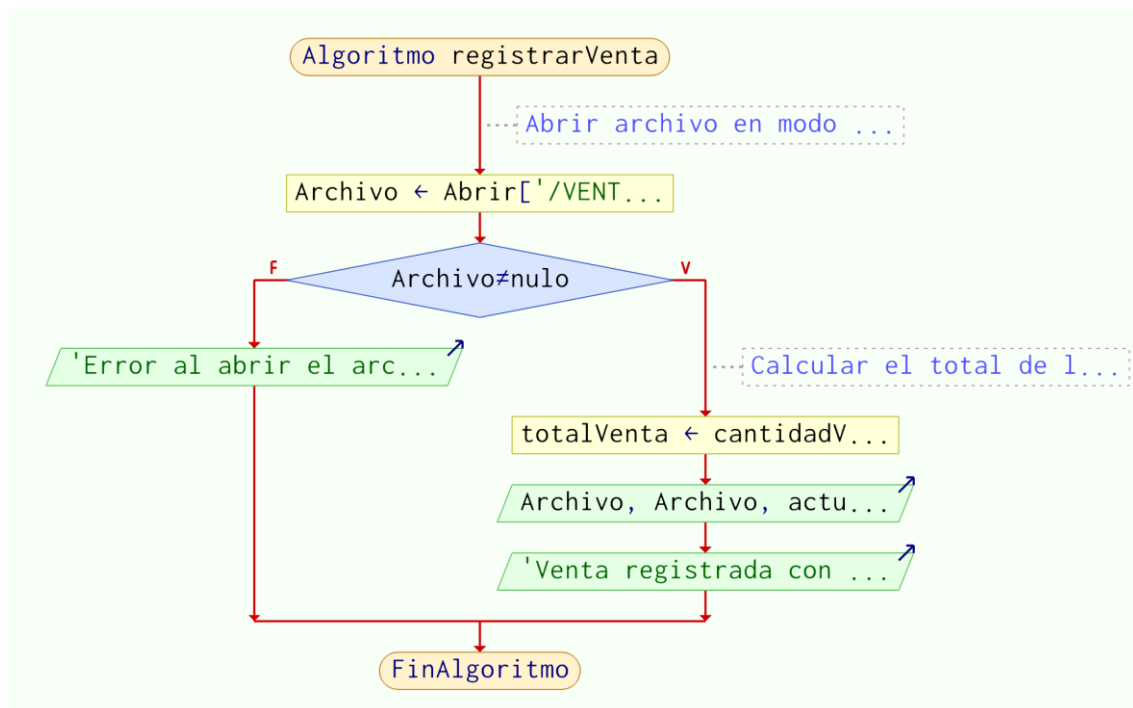
11. CÓDIGO FUENTE

```
// Registrar la venta en el archivo de ventas
ofstream archivoVentas("VENTAS.txt", ios::app); // Abrir archivo en modo agregar

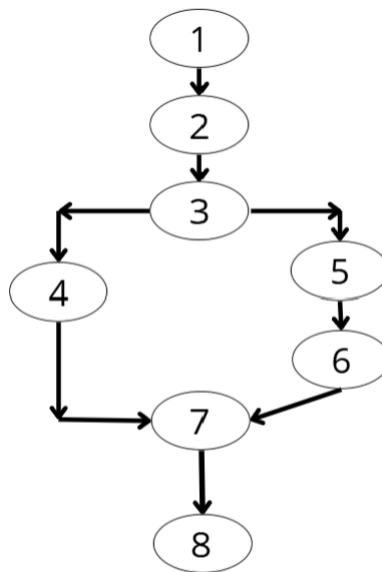
if (archivoVentas.is_open()) {
    // Calcular el total de la venta (cantidad * precio)
    float totalVenta = cantidadVendida * actual->precio;
    archivoVentas << actual->codigo << ", "
    << actual->nombreProducto << ", "
    << cantidadVendida << ", "
    << totalVenta << "\n";

    archivoVentas.close();
    cout << "Venta registrada con éxito. Stock actualizado." << endl;
} else {
    cout << "Error al abrir el archivo de ventas." << endl;
}
```

12. DIAGRAMA DE FLUJO (DF)



13. GRAFO DE FLUJO (GF)



14. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

RUTAS

R1: 1 → 2 → 3 → 4 → 7 → 8

R2: 1 → 2 → 3 → 5 → 6 → 7 → 8

15. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicaos(decisiones)} + 1$
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$
 $V(G) = 8 - 8 + 2 = 2$

DONDE:

P: Número de nodos predicaos

A: Número de aristas

N: Número de nodos

REQ 004 VISUALIZAR LOS PRODUCTOS

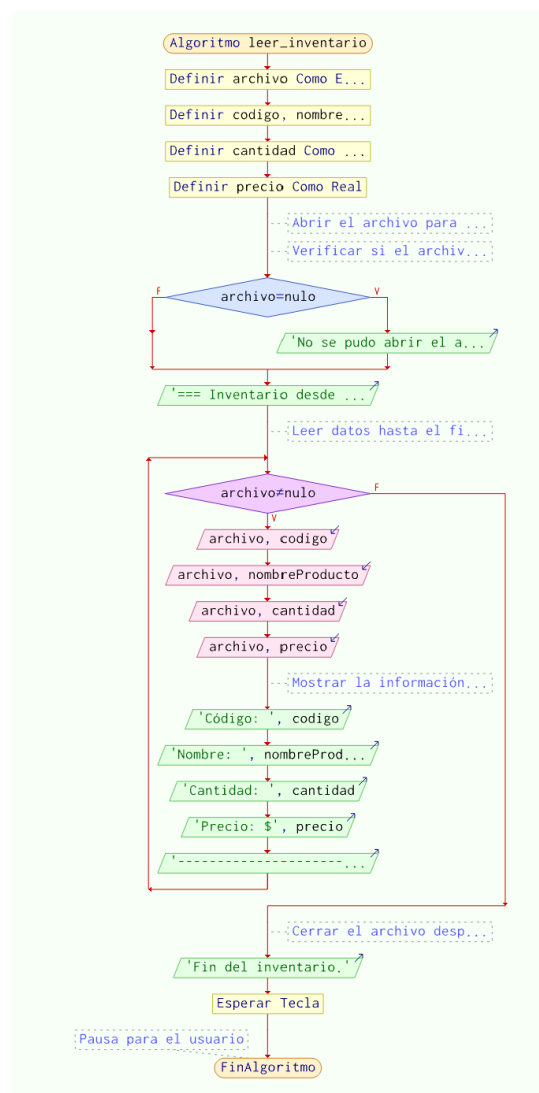
16. CÓDIGO FUENTE

```

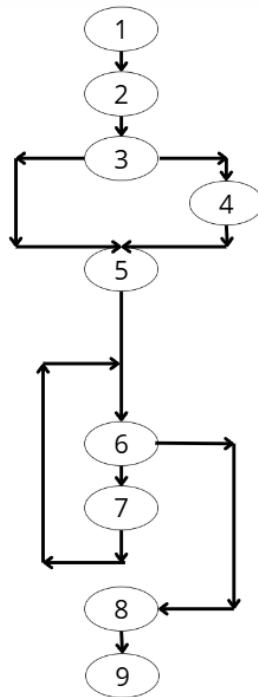
158 // Función para leer el inventario desde el archivo y mostrarlo
159 void leerInventario() {
160     ifstream archivo("INVENTARIO.txt");
161
162     if (!archivo.is_open()) {
163         cout << "No se pudo abrir el archivo INVENTARIO.txt.\n";
164         return;
165     }
166
167     string codigo, nombreProducto;
168     int cantidad;
169     float precio;
170
171     cout << "\n=== Inventario desde archivo ===\n";
172     while (getline(archivo, codigo, ',') && getline(archivo, nombreProducto, ',') && archivo >> cantidad && archivo.ignore() && archivo >> precio) {
173         archivo.ignore(); // Ignorar el salto de línea después de cada precio
174         cout << "Codigo: " << codigo << "\n";
175         cout << "Nombre: " << nombreProducto << "\n";
176         cout << "Cantidad: " << cantidad << "\n";
177         cout << "Precio: $" << precio << "\n";
178         cout << "-----\n";
179     }
180
181     archivo.close();
182     cout << "Fin del inventario.\n";
183     cin.ignore(); // Pausa para el usuario
184 }

```

17. DIAGRAMA DE FLUJO (DF)



18. GRAFO DE FLUJO (GF)



19. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1 → 2 → 3 → 4 → 5 → 6 → 8 → 9

R2: 1 → 2 → 3 → 5 → 6 → 8 → 9

R3: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 6 → 8 → 9

R3: 1 → 2 → 3 → 5 → 6 → 7 → 6 → 8 → 9

20. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 9 + 1 = 10$
- $V(G) = A - N + 2$
 $V(G) = 10 - 9 + 2 = 3$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

REQ 005 VISUALIZAR LAS VENTAS

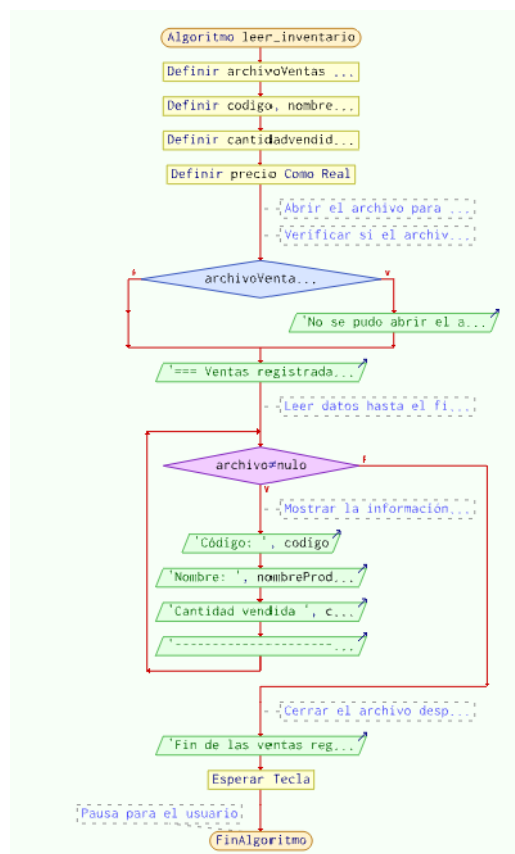
21. CÓDIGO FUENTE

```

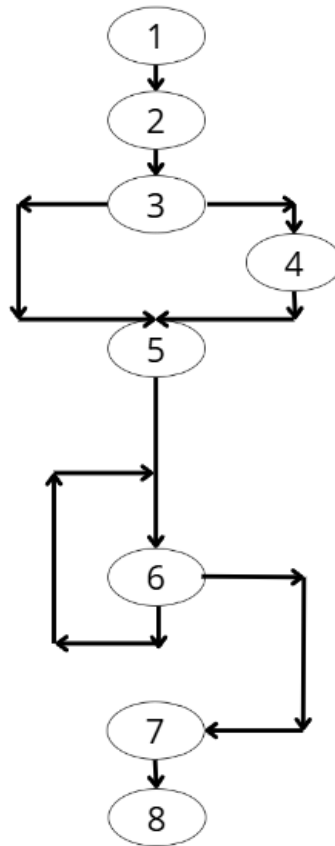
410
411 // Función para ver las ventas registradas
412 void verVentas() {
413     ifstream archivoVentas("VENTAS.txt");
414
415     if (!archivoVentas.is_open()) {
416         cout << "No se pudo abrir el archivo de ventas.\n";
417         return;
418     }
419
420     string linea;
421     string codigo, nombreProducto;
422     int cantidadVendida;
423
424     cout << "\n=== Ventas registradas ===\n";
425     while (getline(archivoVentas, linea)) {
426         stringstream ss(linea);
427         getline(ss, codigo, ',');
428         getline(ss, nombreProducto, ',');
429         ss >> cantidadVendida;
430         cout << "Código: " << codigo << "\n";
431         cout << "Nombre: " << nombreProducto << "\n";
432         cout << "Cantidad vendida: " << cantidadVendida << "\n";
433         cout << "-----\n";
434     }
435
436     archivoVentas.close();
437     cout << "Fin de las ventas registradas.\n";
438     cin.ignore(); // Pausa para el usuario
439 }

```

22. DIAGRAMA DE FLUJO (DF)



23. GRAFO DE FLUJO (GF)



24. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

- R1:** 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
- R2:** 1 → 2 → 3 → 5 → 6 → 7 → 8
- R3:** 1 → 2 → 3 → 4 → 5 → 6 → 6 → 7 → 8
- R3:** 1 → 2 → 3 → 5 → 6 → 6 → 7 → 8

25. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = 8 + 1 = 9$
- $V(G) = A - N + 2$
 $V(G) = 9 - 8 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

