

Prueba de Caja Blanca

*“Implementación de un sistema de catálogo para la
empresa Muebles M&L”*



Integrantes:

Lizandro Rivera

Bernardo Suárez

Mateo Unda

Fecha 2025-02-20

HISTORIA DE USUARIO (HU)			
ITEM REQ001	USUARIO Administrador y Cliente	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP LIZANDRO RIVERA	
QUE Implementar un sistema de registro de usuario.	PARA QUE Nuevos usuarios se registrarán para acceder fácilmente en el futuro.	COMO Crear un archivo .txt para guardar los usuarios registrados, ya sea como Administrador o como Cliente.	
NOMBRE HISTORIA	REGISTRO DE NUEVOS USUARIOS		
PRUEBA	Verificar la creación del archivo .txt y el registro correcto de las credenciales ingresadas.		COMENTARIOS NA

1. CÓDIGO FUENTE

// Función para guardar usuarios en un archivo

template<typename T>

void guardarUsuarios(const string& archivo, const vector<T>& usuarios) {

 ofstream file(archivo);

 for (const auto& user : usuarios) {

 file << user.username << " " << user.password << "\n";

 }

}

// Función para registrar un usuario y guardarlo en un archivo

void registrarUsuario(vector<Usuario>& usuarios, const string& archivo) {

 cout << "Ingrese nombre de usuario: ";

 string username;

 cin >> username;

 cout << "Ingrese contraseña: ";

 string password = leerPassword(); // Función que oculta la contraseña con asteriscos

 // Agregar usuario a la lista

 usuarios.push_back({username, password});

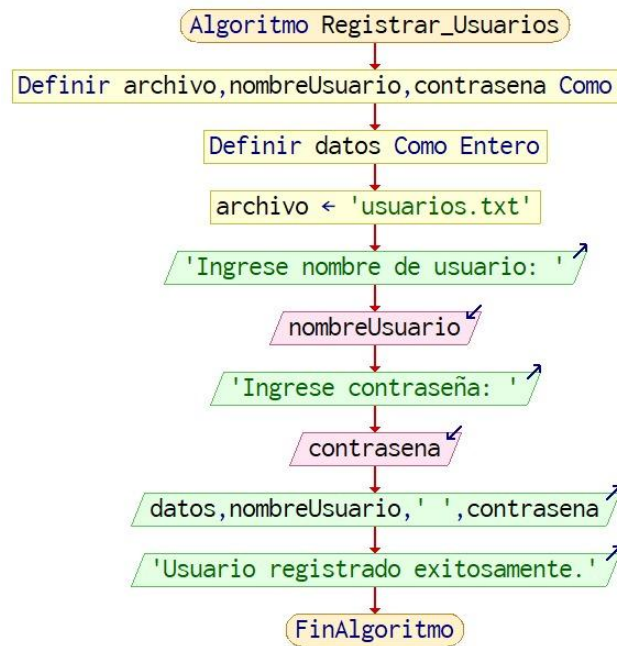
 // Guardar los usuarios actualizados en el archivo

 guardarUsuarios(archivo, usuarios);

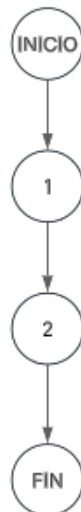
 cout << "Usuario registrado exitosamente.\n";

}

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)



RUTAS

1,2

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 0+1$
 $V(G) = 1$
- $V(G) = 3-4+2$
 $V(G) = 1$

HISTORIA DE USUARIO (HU)			
ITEM REQ002	USUARIO Administrador y Cliente	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP LIZANDRO RIVERA	
QUE	Implementar un sistema de inicio de sesión.	PARA QUE	Identificar al tipo de usuario que está usando el producto.
COMO	Crear función de inicio de sesión que verifique las credenciales en el archivo .txt creado.		
NOMBRE HISTORIA	INICIO DE SESIÓN		
PRUEBA	Iniciar sesión con una serie de credenciales designadas para validación, tanto correctas como incorrectas.		COMENTARIOS
e desplegará distintas opciones según el tipo de usuari			

1. CÓDIGO FUENTE

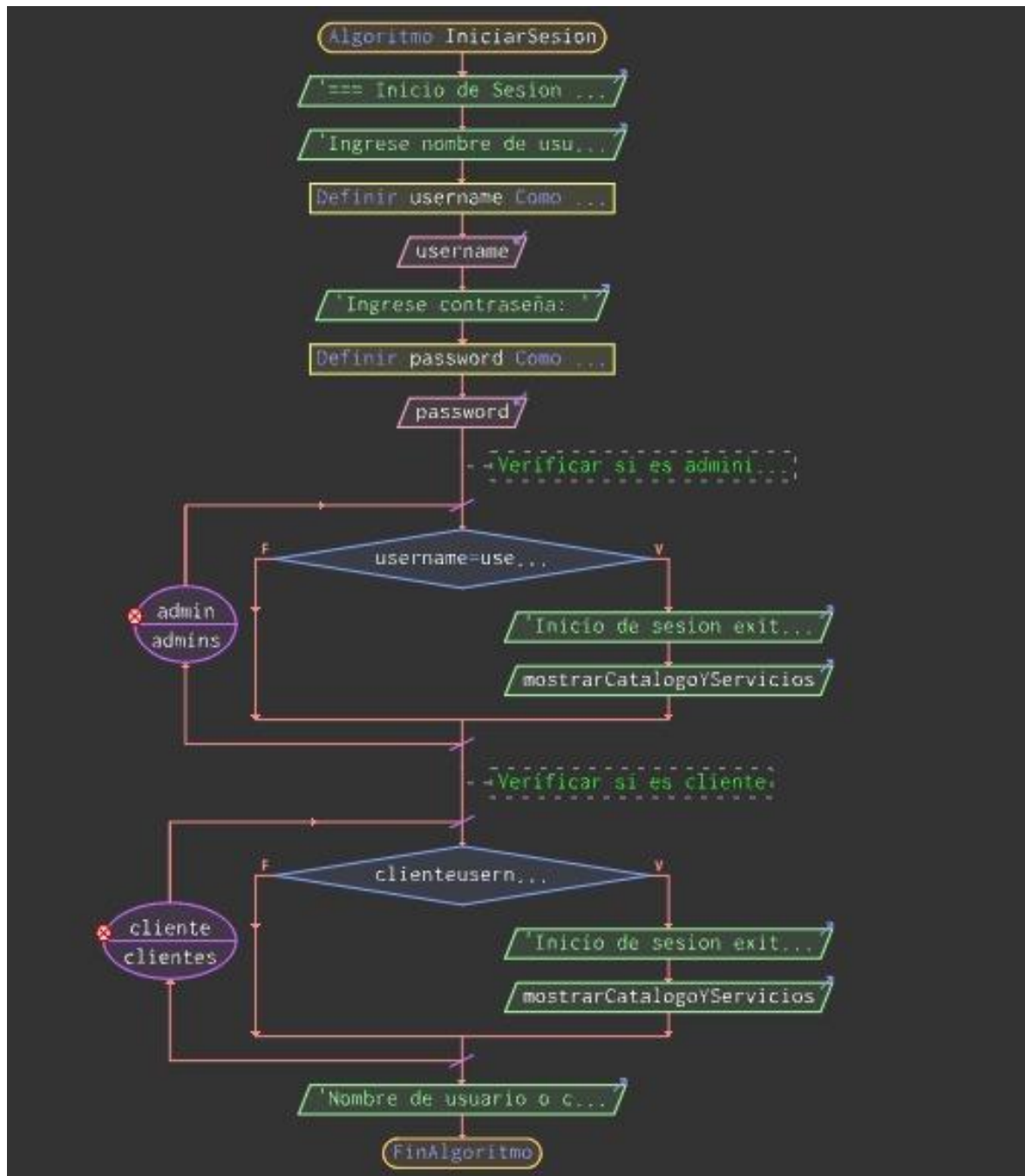
```
void iniciarSesion() {
    cout << "=== Inicio de Sesion ===\n";
    cout << "Ingrese nombre de usuario: ";
    string username;
    cin >> username;
    cout << "Ingrese contrasena: ";
    string password;
    cin >> password;

    // Verificar si es administrador
    for (const auto& admin : admins) {
        if (admin.username == username && admin.password == password) {
            cout << "Inicio de sesion exitoso como Administrador.\n";
            mostrarCatalogoYServicios();
            modificarCatalogoYServicios();
            return;
        }
    }

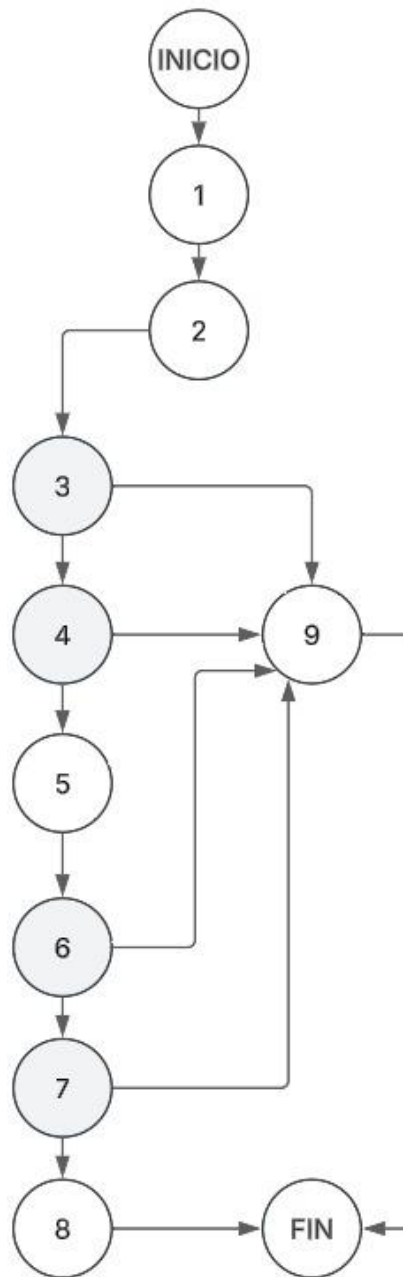
    // Verificar si es cliente
    for (const auto& cliente : clientes) {
        if (cliente.username == username && cliente.password == password) {
            cout << "Inicio de sesion exitoso como Cliente.\n";
            mostrarCatalogoYServicios();
            return;
        }
    }

    cout << "Nombre de usuario o contrasena incorrectos.\n";}
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)





4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

1, 2, 3, 9
1, 2, 3, 4, 9
1, 2, 3, 4, 5, 6, 9
1, 2, 3, 4, 5, 6, 7, 9
1, 2, 3, 4, 5, 6, 7, 8

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 4+1$
 $V(G)=5$
- $V(G) = 14-11+2$
 $V(G)= 5$

HISTORIA DE USUARIO (HU)			
ITEM REQ003	USUARIO Cliente	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP BERNARDO SUÁREZ	
QUE Implementar listado de productos y servicios.	PARA QUE Apreciar de forma organizada y unificada los productos y servicios ofrecidos.	COMO Creación una función que despliegue la lista de productos y servicios.	
NOMBRE HISTORIA		VER CATÁLOGO	
PRUEBA	Ejecutar la función y verificar que la salida sea correcta.		COMENTARIOS Cuidar legibilidad de la lista.

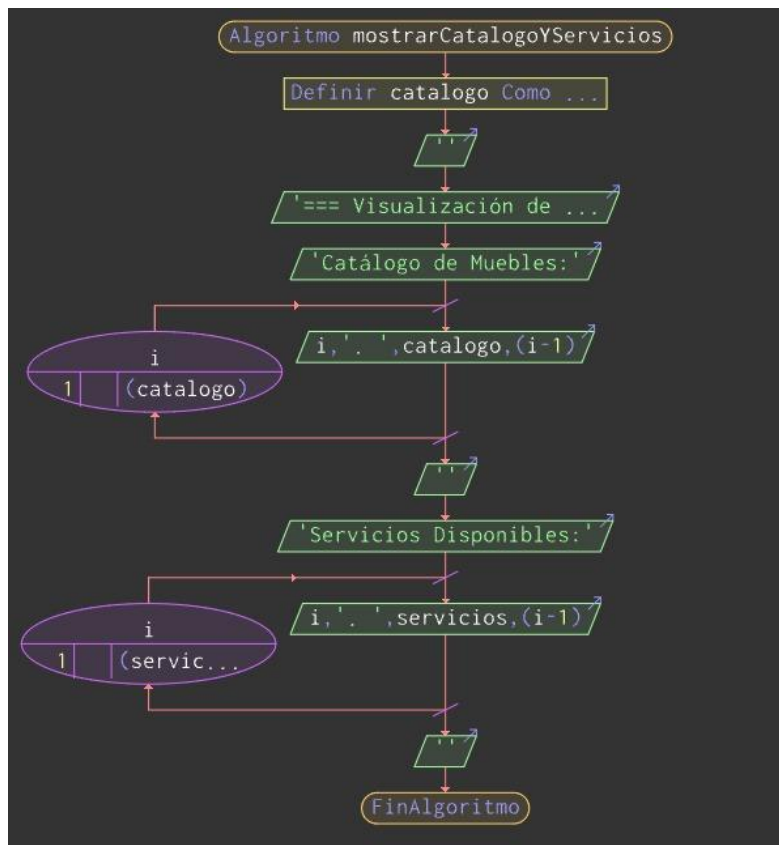
1. CÓDIGO FUENTE

```
void mostrarCatalogoYServicios() {
    cout << "\n=== Visualizacion de Lista de Catalogo y Servicios === \n";
    cout << "Catalogo de Muebles:\n";
    for (size_t i = 0; i < catalogo.size(); i++) {
        cout << i + 1 << ". " << catalogo[i] << "\n"; }

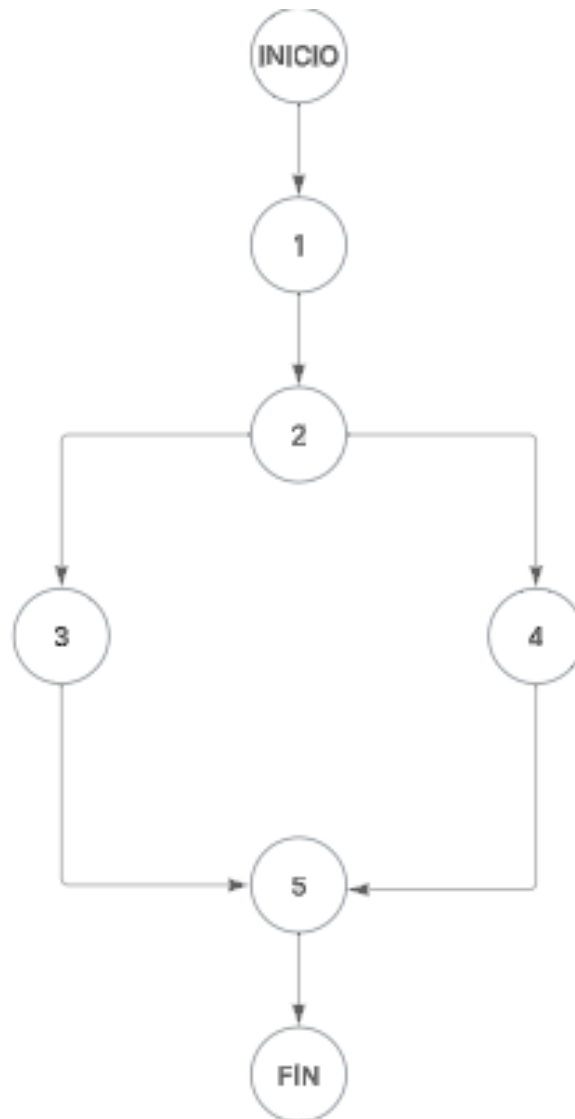
    cout << "\nServicios Disponibles:\n";
    for (size_t i = 0; i < servicios.size(); i++) {
        cout << i + 1 << ". " << servicios[i] << "\n"; }
    cout << "\n"; }

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

1, 2, 3, 5

1, 2, 4, 5

5. COMPLEJIDAD CICLOMÁTICA

- $V(G) = 1+1$
 $V(G)= 2$
- $V(G) = 7-7 + 2$
 $V(G)= 2$

HISTORIA DE USUARIO (HU)			
ITEM	USUARIO	STATUS	
REQ004	Administrador	Terminado	
TIEMPO	PRIORIDAD	PROG. RESP	
7	Alta	LIZANDRO RIVERA	
QUE	PARA QUE		COMO
Permitir la incorporación o eliminación de ítems.	Mantener el sistema actualizado ante posibles cambios en el catálogo.		El usuario Administrador tendrá una opción para modificar y actualizar el catálogo según se requiera.
NOMBRE HISTORIA	MODIFICAR CATÁLOGO		
PRUEBA	COMENTARIOS		
Realizar cambios de prueba	NA		

1. CÓDIGO FUENTE

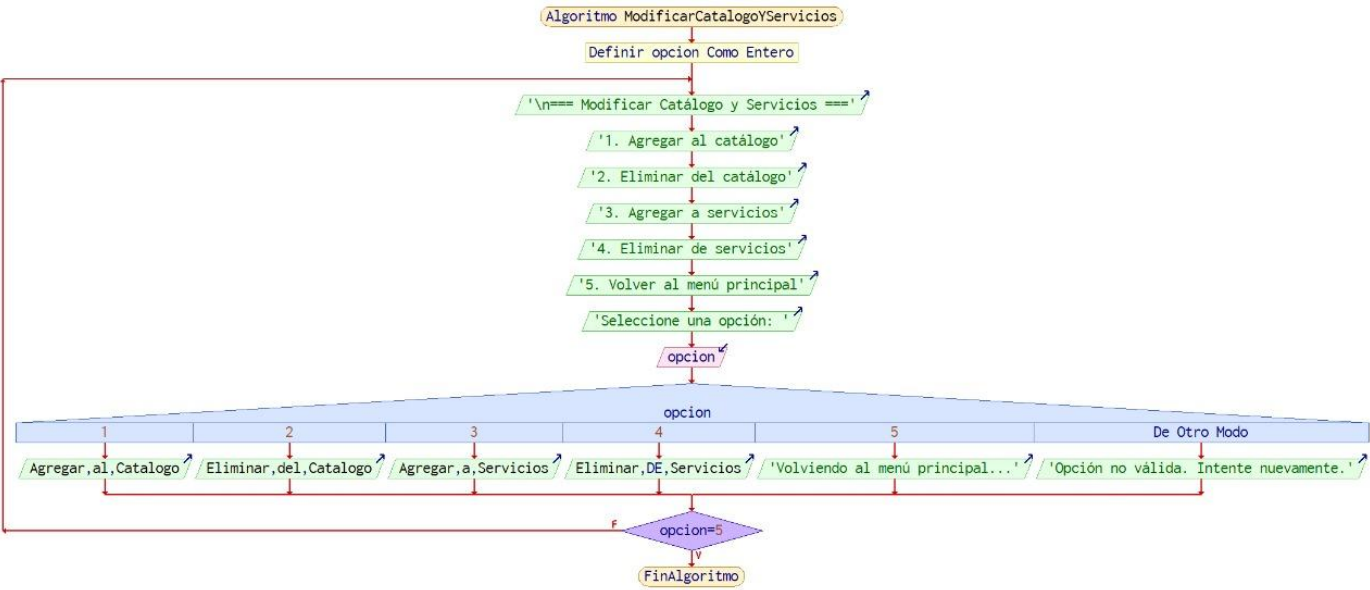
```

void modificarCatalogoYServicios() {
    int opcion;
    do {
        cout << "\n=== Modificar Catalogo y Servicios ===\n";
        cout << "1. Agregar al catalogo\n";
        cout << "2. Eliminar del catalogo\n";
        cout << "3. Agregar a servicios\n";
        cout << "4. Eliminar de servicios\n";
        cout << "5. Volver al menu principal\n";
        cout << "Seleccione una opcion: ";
        cin >> opcion;
        cin.ignore(); // Limpiar el buffer

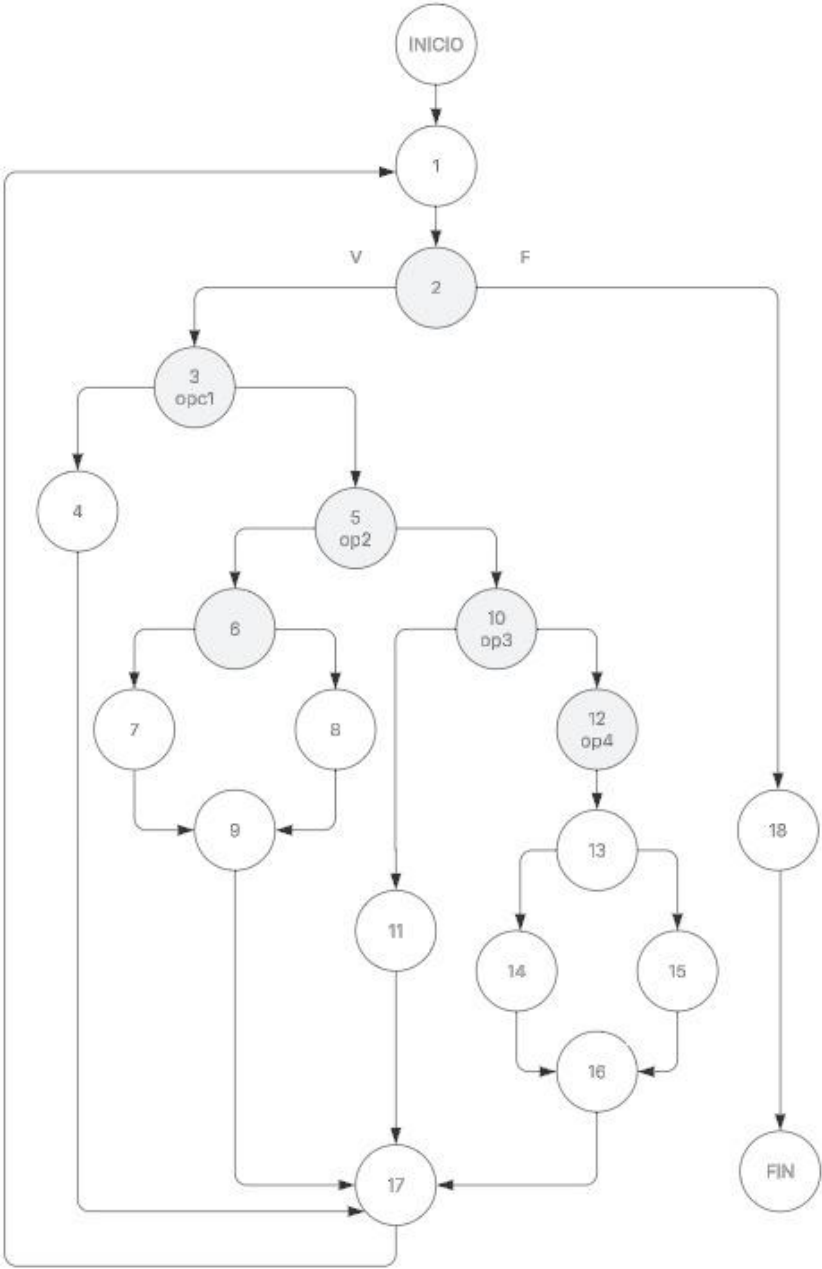
        if (opcion == 1) {
            if (catalogo.size() >= 10) {
                cout << "El catalogo esta lleno.\n";
                continue;
            }
            cout << "Ingrese el nombre del nuevo articulo: ";
            string articulo;
            getline(cin, articulo);
            catalogo.push_back(articulo);
            cout << "Articulo agregado exitosamente.\n";
        } else if (opcion == 2) {
            cout << "Ingrese el número del articulo a eliminar: ";
            int eliminar;
            cin >> eliminar;
            cin.ignore();
        }
    } while (opcion != 5);
}
  
```

```
        if (eliminar < 1 || eliminar > (int)catalogo.size()) {
            cout << "Opcion no valida.\n";
            continue;
        }
        catalogo.erase(catalogo.begin() + eliminar - 1);
        cout << "Articulo eliminado exitosamente.\n";
    } else if (opcion == 3) {
        if (servicios.size() >= 10) {
            cout << "La lista de servicios esta llena.\n";
            continue;
        }
        cout << "Ingrese el nombre del nuevo servicio: ";
        string servicio;
        getline(cin, servicio);
        servicios.push_back(servicio);
        cout << "Servicio agregado exitosamente.\n";
    } else if (opcion == 4) {
        cout << "Ingrese el numero del servicio a eliminar: ";
        int eliminar;
        cin >> eliminar;
        cin.ignore();
        if (eliminar < 1 || eliminar > (int)servicios.size()) {
            cout << "Opcion no valida.\n";
            continue;
        }
        servicios.erase(servicios.begin() + eliminar - 1);
        cout << "Servicio eliminado exitosamente.\n";
    } else if (opcion != 5) {
        cout << "Opcion no valida. Intente nuevamente.\n";
    }
} while (opcion != 5);
}
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4

RUTAS

- **1, 2, 3, 4, 17, 18**
- **1, 2, 3, 5, 6, 7, 9, 17, 18**
- **1, 2, 3, 5, 6, 8, 9, 17, 18**
- **1, 2, 3, 5, 10, 11, 17, 18**
- **1, 2, 3, 5, 10, 12, 13, 14, 16, 17, 18**
- **1, 2, 3, 5, 10, 12, 13, 15, 16, 17, 18**
- **1, 2, 18**

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 6 + 1 = 7$
- $V(G) = 25 - 20 + 2 = 7$