

Prueba de Caja Blanca

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



Integrantes:

Lizandro Rivera

Bernardo Suárez

Mateo Unda

Fecha 2025-02-20

ITEM	USUARIO	STATUS	 
REQ001	Sr. Mateo Bedoya	Terminado	
TIEMPO	PRIORIDAD	PROG. RESP	
7	Alta	LIZANDRO RIVERA	
QUE	Guardar los datos de los clientes de la empresa mediante la opción de Registro de Usuario.		PARA QUE Los clientes pueden tener acceso directo a los productos y puedan guardar sus compras en el mismo programa.
COMO	Se va a crear un archivo de texto(txt), en el que se podrán guardar a todos los usuarios que se hayan registrado en la página de Muebles M&L		
NOMBRE HISTORIA		REGISTRO	
PRUEBA	Vamos a registrar a un usuario en el programa, posteriormente vamos a salir del mismo para verificar que se haya guardado adecuadamente el usuario, volviendo a ingresar los datos que ya estaban		COMENTARIOS
			-

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});

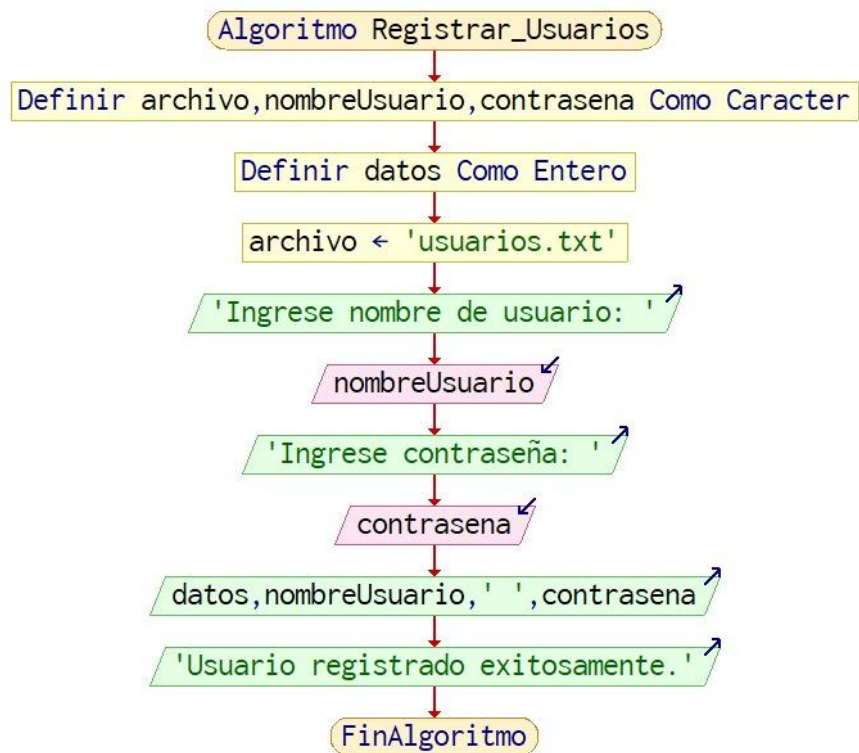
// Guardar los usuarios actualizados en el archivo

guardarUsuarios(archivo, usuarios);

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

}

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

Determinar en base al GF del numeral 4



RUTAS

1,2

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

ITEM REQ002	USUARIO Sr. Mateo Bedoya	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP LIZANDRO RIVERA	
QUE	Identificar al usuario que use la aplicación entre dos tipos establecidos: ADMINISTRADOR o		PARA QUE
	Cada tipo de usuario tendrá acceso a distintas funciones.		COMO
	Generar un sistema que verifique si el usuario es un responsable de la empresa o un cliente, mediante un nombre de usuario y una contraseña.		
NOMBRE HISTORIA	TIPO DE USUARIO		
PRUEBA	Establecer un nombre de usuario y contraseña para cada tipo, e intentar hacer uso de las funciones desarrolladas.		COMENTARIOS
			-

1. CÓDIGO FUENTE

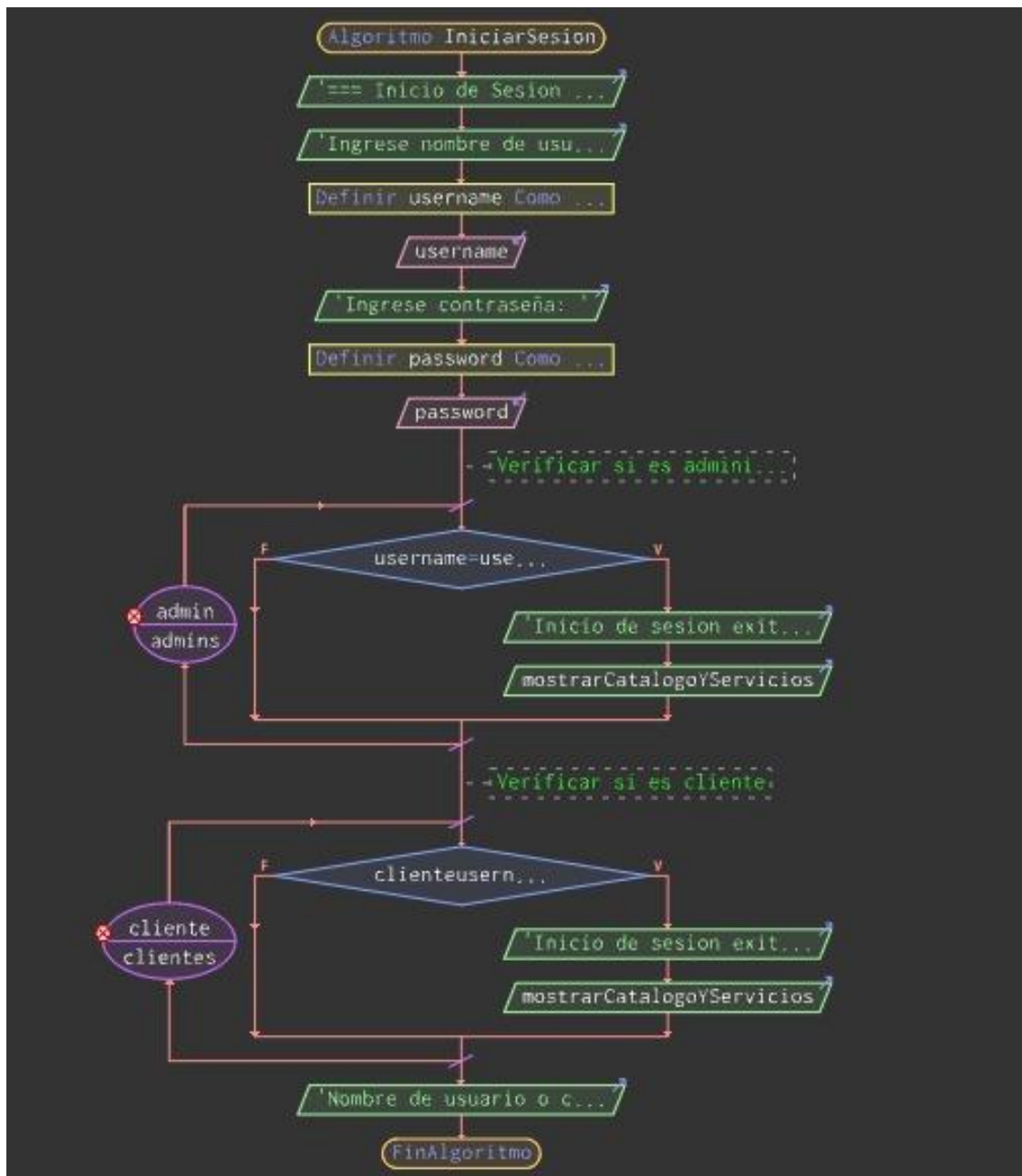
```
void iniciarSesion() {
    cout << "=== Inicio de Sesion ===\n";
    cout << "Ingrese nombre de usuario: ";
    string username;
    cin >> username;
    cout << "Ingrese contraseña: ";
    string password;
    cin >> password;

    // 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;
        }
    }

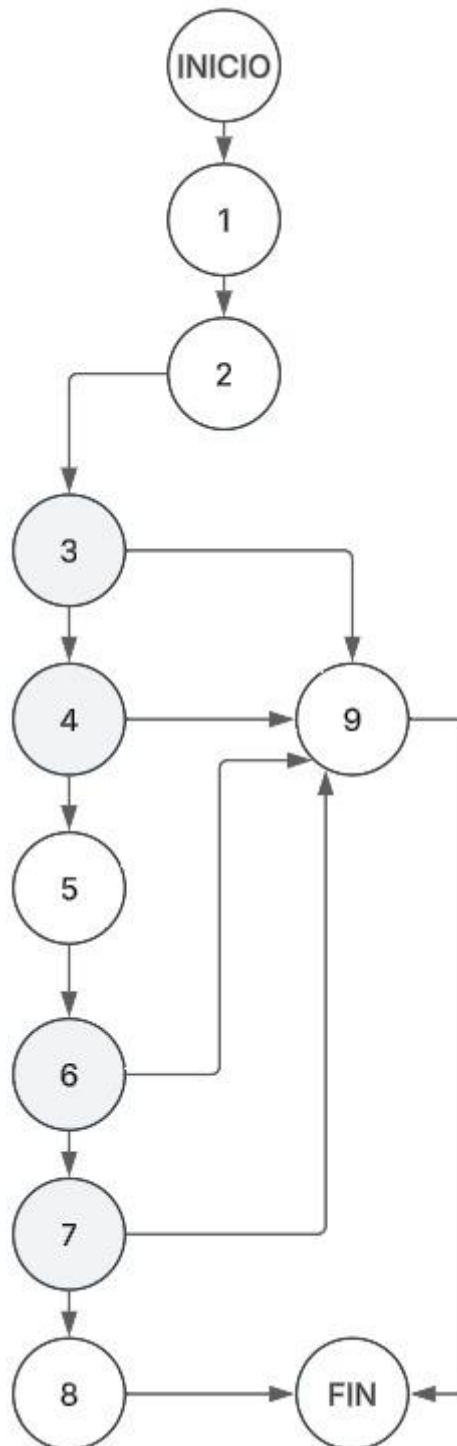
    // Verificar si es cliente
    for (const auto& cliente : clientes) {
        if (cliente.username == username && cliente.password == password) {
            cout << "Inicio de sesion exitoso como Cliente.\n";
            mostrarCatalogoYServicios();
            return;
        }
    }

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

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

Determinar en base al GF del numeral 4

RUTAS

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:

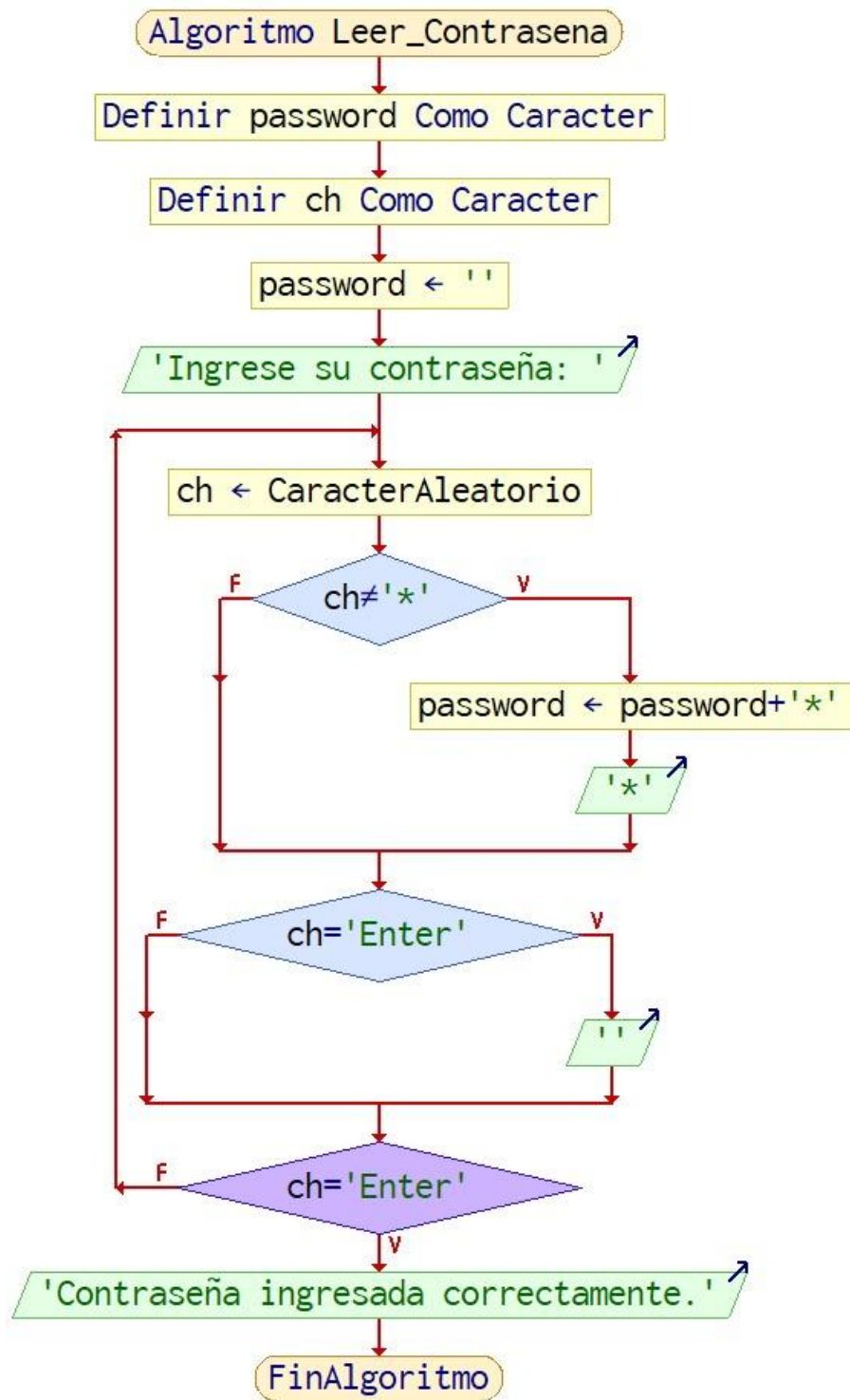
- $V(G) = 4+1$
 $V(G)=5$
- $V(G) = 14-11+2$
 $V(G)= 5$

ITEM REQ003	USUARIO Sr. Mateo Bedoya	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP MATEO UNDA	
QUE	Ocultar la contraseña mediante asteriscos que no posibiliten la visualización externa.		PARA QUE
	Para seguridad visual de los usuarios, evitando la inasión de privacidad de los mismos.		COMO
	Codificar un sistema que oculte los dígitos de la contraseña con ayuda de asteriscos, demostrando cuando la contraseña sea incorrecta.		
NOMBRE HISTORIA	SEGURIDAD DE USUARIO		
PRUEBA	Se registra el usuario y la contraseña, para proceder al inicio de sesión en el que se podrá visualizar el usuario, pero al ingresar la contraseña se visualizarán asteriscos.		COMENTARIOS
			-

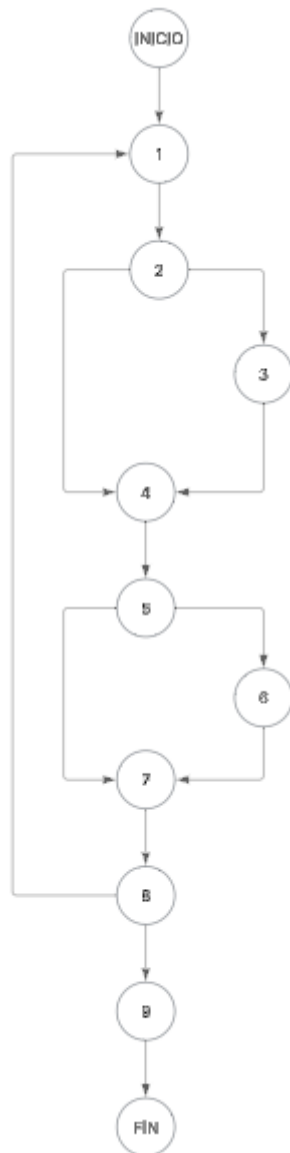
1. CÓDIGO FUENTE

```
// Leer contraseña con asteriscos
string leerPassword() {
    string password;
    char ch;
    while ((ch = _getch()) != '\r') { // '\r' es Enter
        if (ch == '\b' && !password.empty()) {
            password.pop_back();
            cout << "\b \b";
        } else if (ch != '\b') {
            password += ch;
            cout << '*';
        }
    }
    cout << endl;
    return password;
}
```

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, 5, 6, 7, 8, 9

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



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

1, 2, 4, 5, 7, 8

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = 3+1$
 $V(G)=4$
- $V(G) = 13-11+2$
 $V(G)= 4$

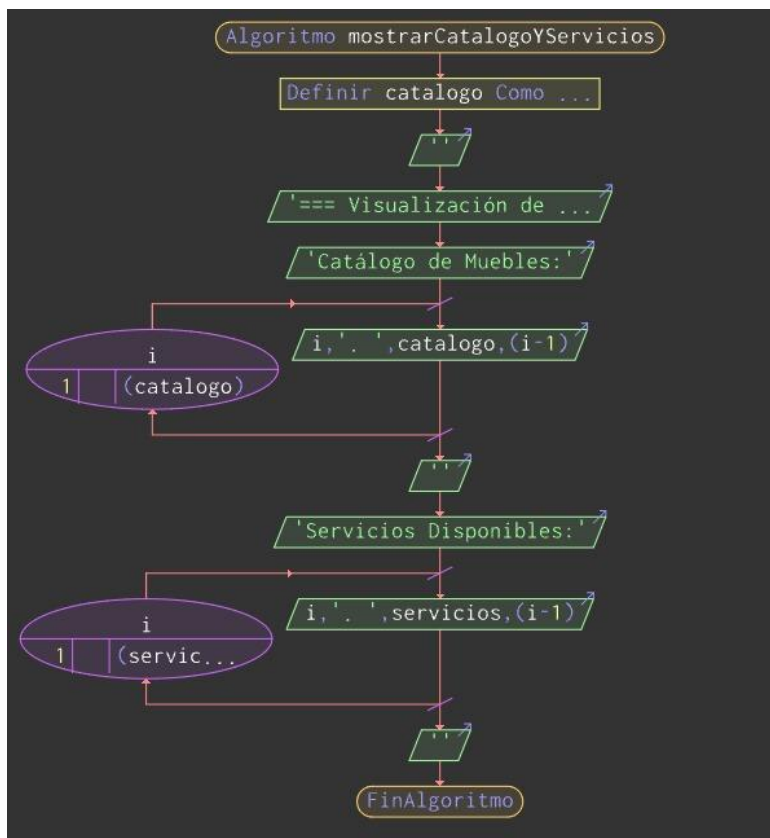
ITEM REQ004	USUARIO Sr. Mateo Bedoya	STATUS Terminado	 
TIEMPO 7	PRIORIDAD Alta	PROG. RESP. BERNARDO SUÁREZ	
QUE	Tener acceso a una lista de productos y servicios (items), que detallen su nombre y su valor.		
PARA QUE	En la aplicación poder generar un apartado exclusivamente para tener un catalogo para poder comprar		
COMO	Creación de una lista con los items disponibles y sus detalles, nombre, características, valor, etc.		
NOMBRE HISTORIA		CATÁLOGO	
PRUEBA	Al iniciar sesión se desplegará la lista de catálogo de productos y servicios, de manera organizada y clara para verificar todos los detalles necesarios.		COMENTARIOS

1. CÓDIGO FUENTE

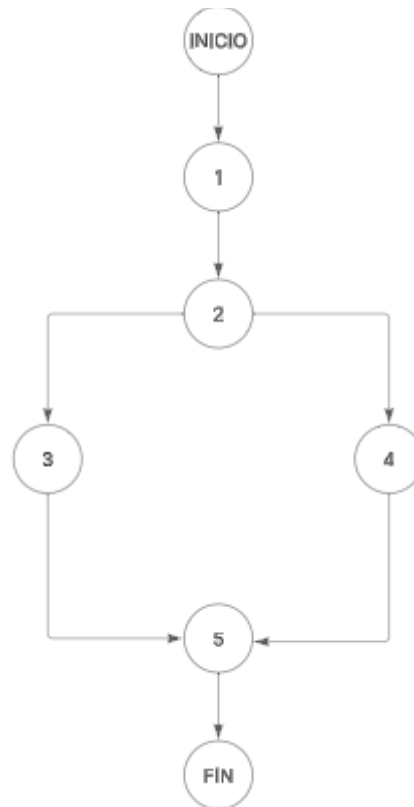
```
void mostrarCatalogoYServicios() {
    cout << "\n=== Visualizacion de Lista de Catalogo y Servicios ===\n";
    cout << "Catalogo de Muebles:\n";
    for (size_t i = 0; i < catalogo.size(); i++) {
        cout << i + 1 << ". " << catalogo[i] << "\n";
    }

    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 basico)

Determinar en base al GF del numeral 4

RUTAS



1, 2, 3, 5

1, 2, 4, 5

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

ITEM REQ005	USUARIO Sr. Mateo Bedoya	STATUS Terminado	 			
TIEMPO 7	PRIORIDAD Alta	PROG. RESP LIZANDRO RIVERA				
QUE	Con el logueo de un administrador permitir funciones de editar cualquier novedad en el		PARA QUE	En la aplicación al entrar como administrador se podrá tener una vista amplia del catálogo con opciones de edición en el mismo	COMO	En la aplicación al momento del inicio de sesión se presentará una vista diferente a la del cliente en la que se presentarán las opciones de eliminar o
NOMBRE HISTORIA		MODIFICACIÓN				
PRUEBA	Al iniciar sesión como administrador se verá la opción de ver el catálogo y de editar la lista del catálogo de productos y los servicios.			COMENTARIOS	-	

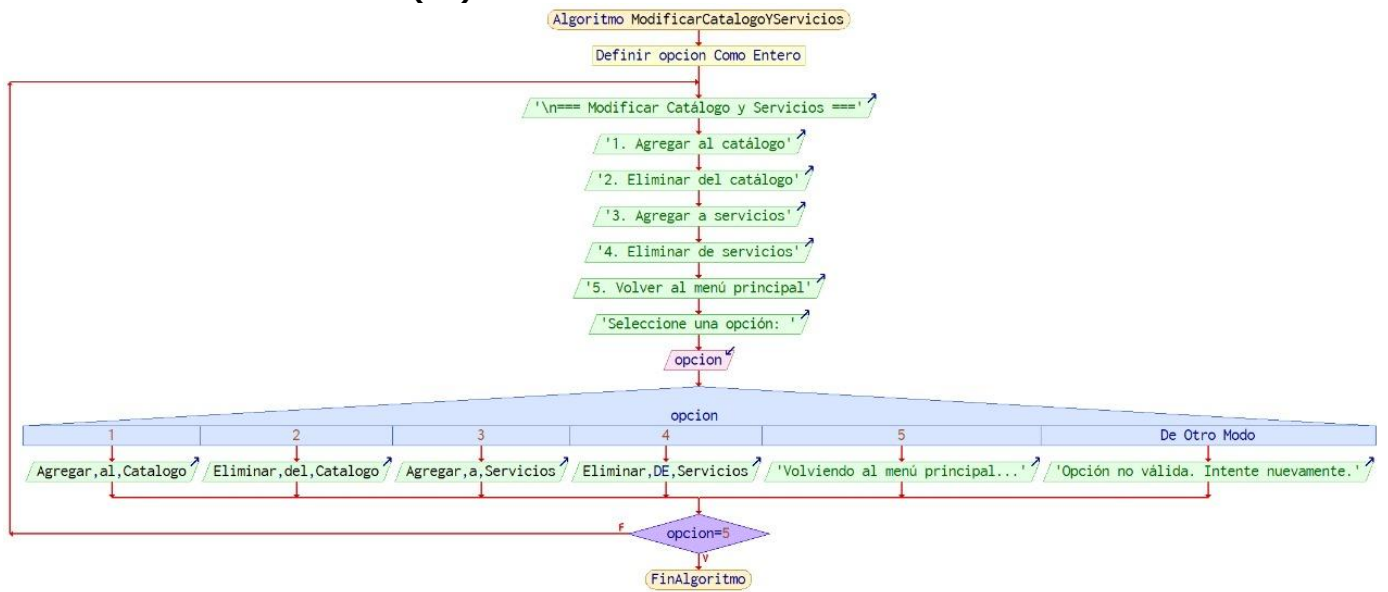
1. CÓDIGO FUENTE

```
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

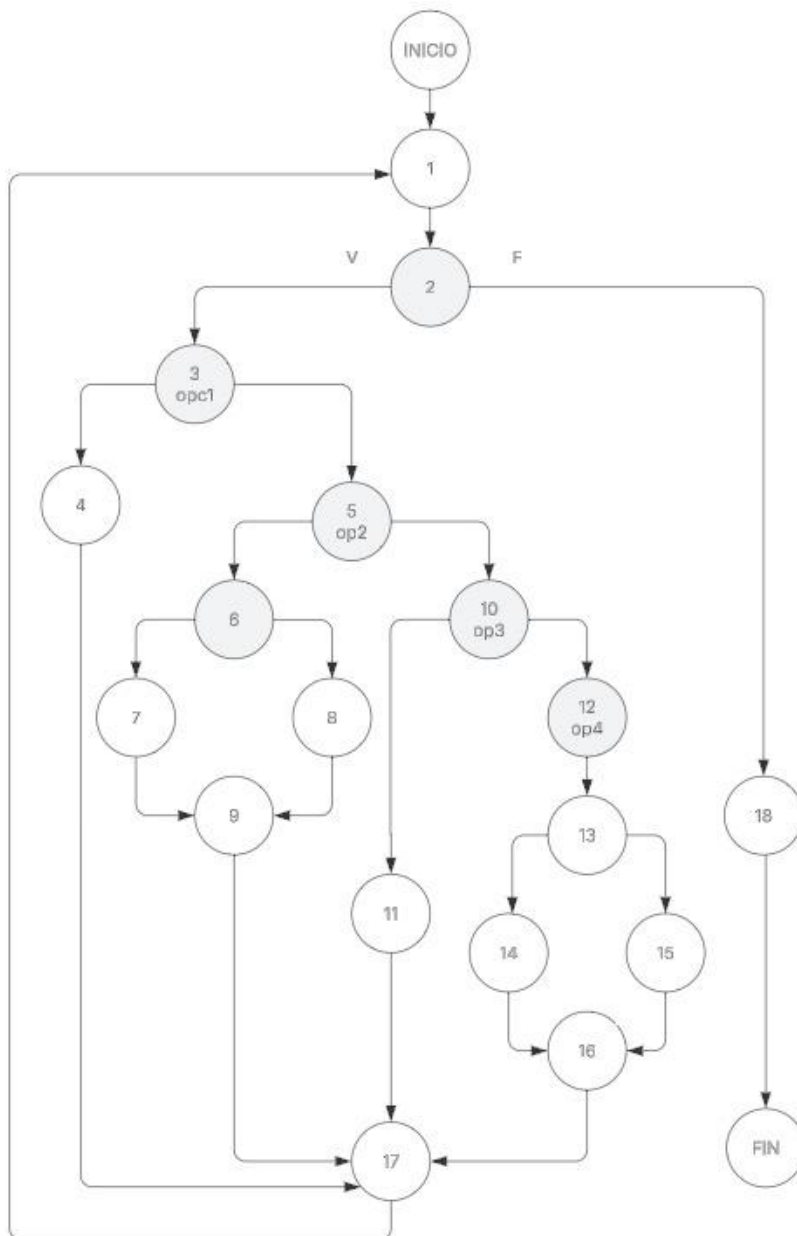
        if (opcion == 1) {
            if (catalogo.size() >= 10) {
                cout << "El catalogo esta lleno.\n";
                continue;
            }
            cout << "Ingrese el nombre del nuevo articulo: ";
            string articulo;
            getline(cin, articulo);
            catalogo.push_back(articulo);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

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