|  |
| --- |
|  |
| 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 2025-01-16** |

**Prueba caja blanca de**  **INICIO DE SESIÓN**

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;**

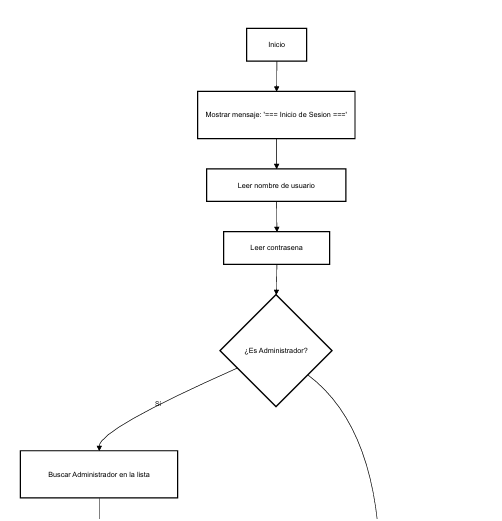
**}**

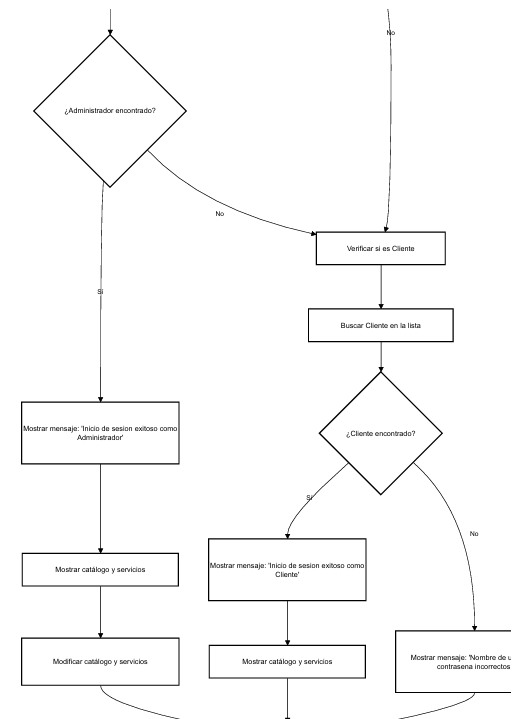
**}**

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

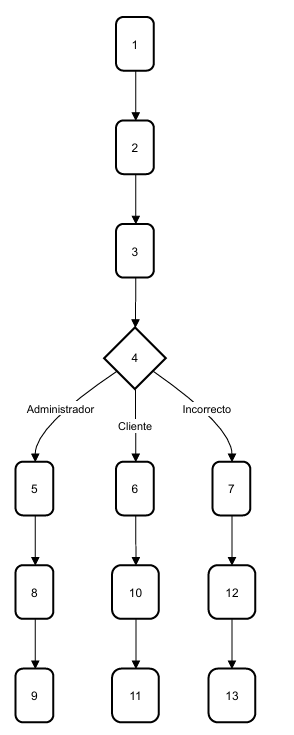
**}**

1. **DIAGRAMA DE FLUJO (DF)**





1. **GRAFO DE FLUJO (GF)**



1. **IDENTIFIACCIÒN DE LAS RUTAS (Camino basico)**

Determinar en base al GF del numeral 4

**RUTAS**

**1 → 2 → 3 → 4 → B1 → C1 → C2**

**Flujo como cliente:**

**1 → 2 → 3 → 4 → B2 → D1 → D2**

**1 → 2 → 3 → 4 → B3 → E1 → E2**

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G)=2+1

V(G)=3

* V(G) = 4– 5 + 2

V(G)= 3

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

**Prueba caja blanca de**  **VISUALIZACIÓN DE LISTA DE CATÁLOGO Y SERVICIOS**

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++) {**

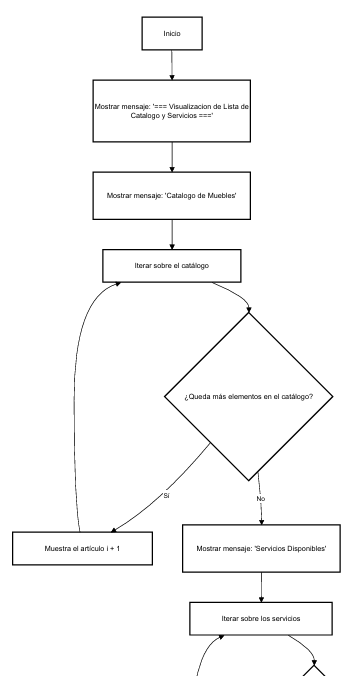
**cout << i + 1 << ". " << servicios[i] << "\n";**

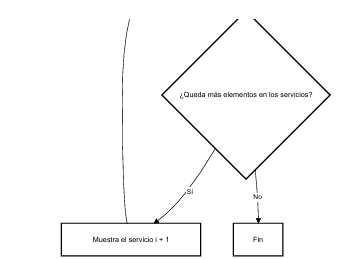
**}**

**cout << "\n";**

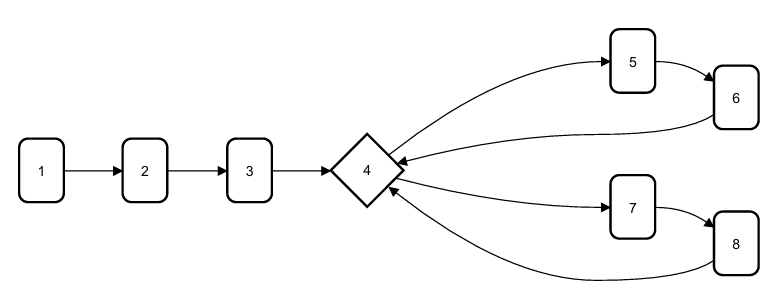
**}**

1. **DIAGRAMA DE FLUJO (DF)**





1. **GRAFO DE FLUJO (GF)**



1. **IDENTIFIACCIÒN DE LAS RUTAS (Camino basico)**

Determinar en base al GF del numeral 4

**RUTAS**

**1 → 2 → 3 → 4 → 5 → 6 → 4**

**1 → 2 → 3 → 4 → 5 → 6 → 4 → 7 → 8 → 4**

**1 → 2 → 3 → 4 → 5 → 6 → 4 → 5 → 6 → 4 → 7 → 8 → 4**

**1 → 2 → 3 → 4 → 5 → 6 → 4 → 7 → 8 → 4 → 7 → 8 → 4**

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = 2+1

V(G)=3

* V(G) = 3 – 2 + 2

V(G)= 3

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

**Prueba caja blanca de**  **PERMITIR LA MODIFICACIÓN DE LA LISTA DE CATÁLOGO Y SERVICIOS**

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 numero 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) {**

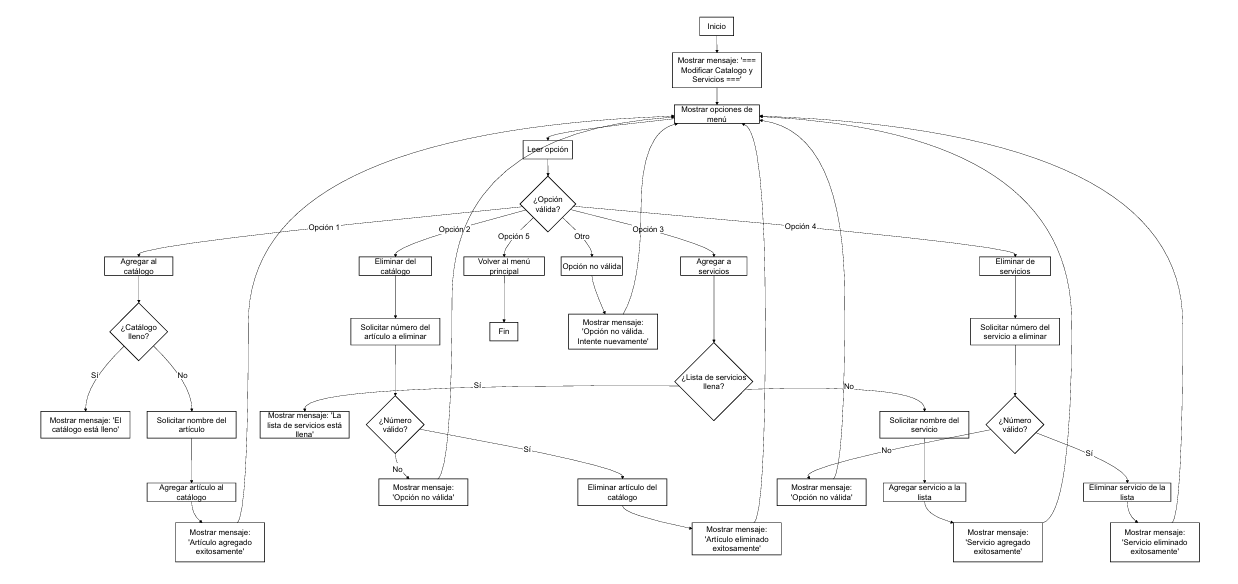
**cout << "Opcion no valida. Intente nuevamente.\n";**

**}**

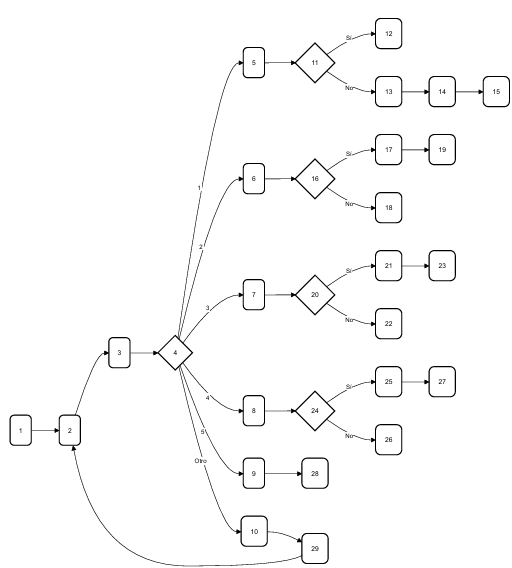
**} while (opcion != 5);**

**}**

1. **DIAGRAMA DE FLUJO (DF)**



1. **GRAFO DE FLUJO (GF)**



1. **IDENTIFIACCIÒN DE LAS RUTAS (Camino basico)**

Determinar en base al GF del numeral 4

**RUTAS**

**A1 → A2 → A3 → A4 → 5 → 11 → 12 → 13 → 14 → 15**

**A1 → A2 → A3 → A4 → 6 → 16 → 17 → 19**

**A1 → A2 → A3 → A4 → 7 → 20 → 21 → 23**

**A1 → A2 → A3 → A4 → 8 → 24 → 25 → 27**

**A1 → A2 → A3 → A4 → 9 → 28**

**A1 → A2 → A3 → A4 → 10 → 29 → A2**

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

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

V(G)=7

* V(G) = 6 – 15 + 2

V(G)= 7

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos