

Prueba de Caja Blanca

*“Título proyecto sistema de automatización de mensajes
e ingreso de datos para fechas importantes”*

Integrantes:

**Alejandro De La Cruz
Santiago Nogales
Ian Escobar**

Fecha 2025-06-16

Prueba caja blanca de describa el requisito funcional

1. CÓDIGO FUENTE

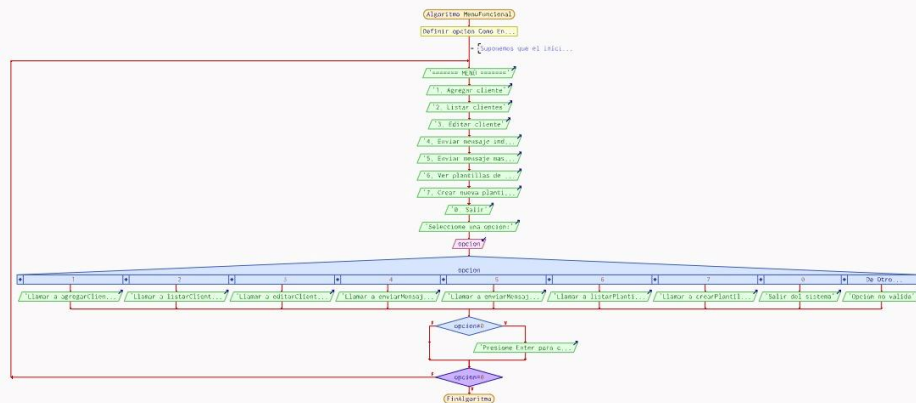
Pegar el trozo de código fuente que se requiere para el caso de prueba

programa final > src > main > java > com > example > Main.java

```
1
2 package com.example;
3
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6 import org.springframework.context.annotation.ComponentScan;
7
8 @SpringBootApplication
9 @ComponentScan(basePackages = {"com.example", "controllers", "services", "models"})
10 public class Main {
11     public static void main(String[] args) {
12         SpringApplication.run(Main.class, args);
13     }
14 }
15
```

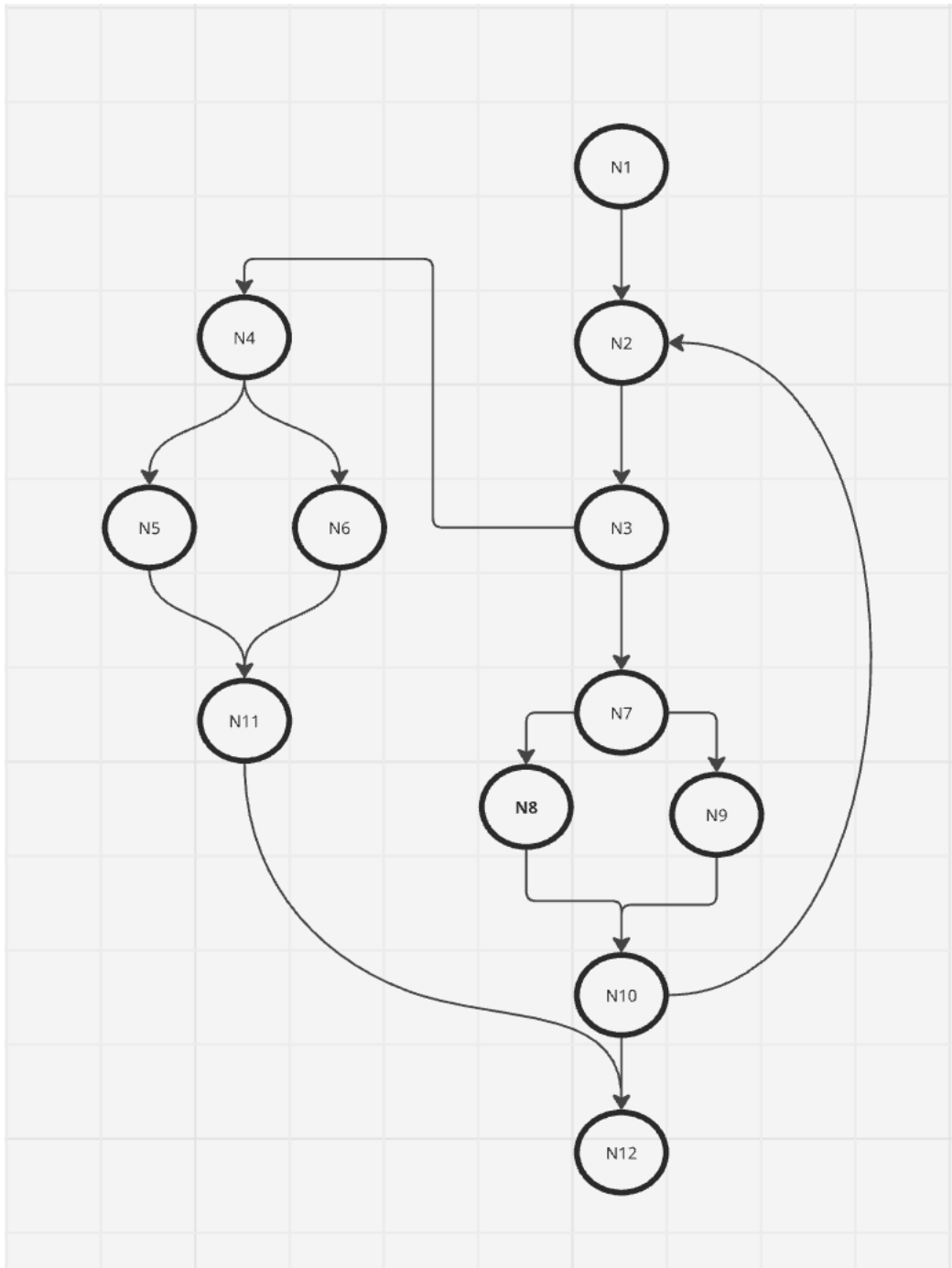
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4

Rutas Independientes:

R1: 1 → 2 → 6 (Login fallido).

R2: 1 → 2 → 3 → 4 → 5 → 6 (Login exitoso → Salir).

R3: 1 → 2 → 3 → 4 → 5 → 7 → 8 → 9/10/11/12 → 3... (Bucle de opciones).

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados (decisiones)} + 1$
- $V(G) = P + 1 = 3 + 1 = 4$
- $V(G) = A - N + 2$
- $V(G) = A - N + 2 = 14$

DONDE:

P: Número de nodos predichados

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

*“Título proyecto sistema de automatización de mensajes
e ingreso de datos para fechas importantes”*

Integrantes:

**Alejandro De La Cruz
Santiago Nogales
Ian Escobar**

Fecha 2025-06-16

Prueba caja blanca de describa el requisito funcional

1. CÓDIGO FUENTE

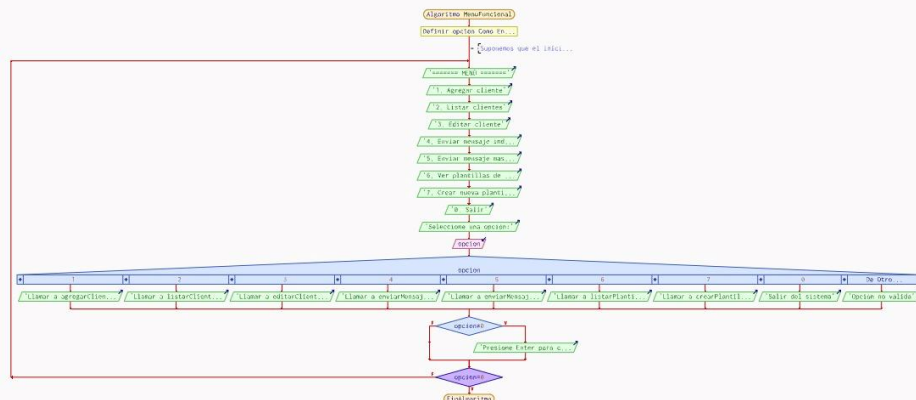
Pegar el trozo de código fuente que se requiere para el caso de prueba

programa final > src > main > java > com > example > Main.java

```
1
2 package com.example;
3
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6 import org.springframework.context.annotation.ComponentScan;
7
8 @SpringBootApplication
9 @ComponentScan(basePackages = {"com.example", "controllers", "services", "models"})
10 public class Main {
11     public static void main(String[] args) {
12         SpringApplication.run(Main.class, args);
13     }
14 }
15
```

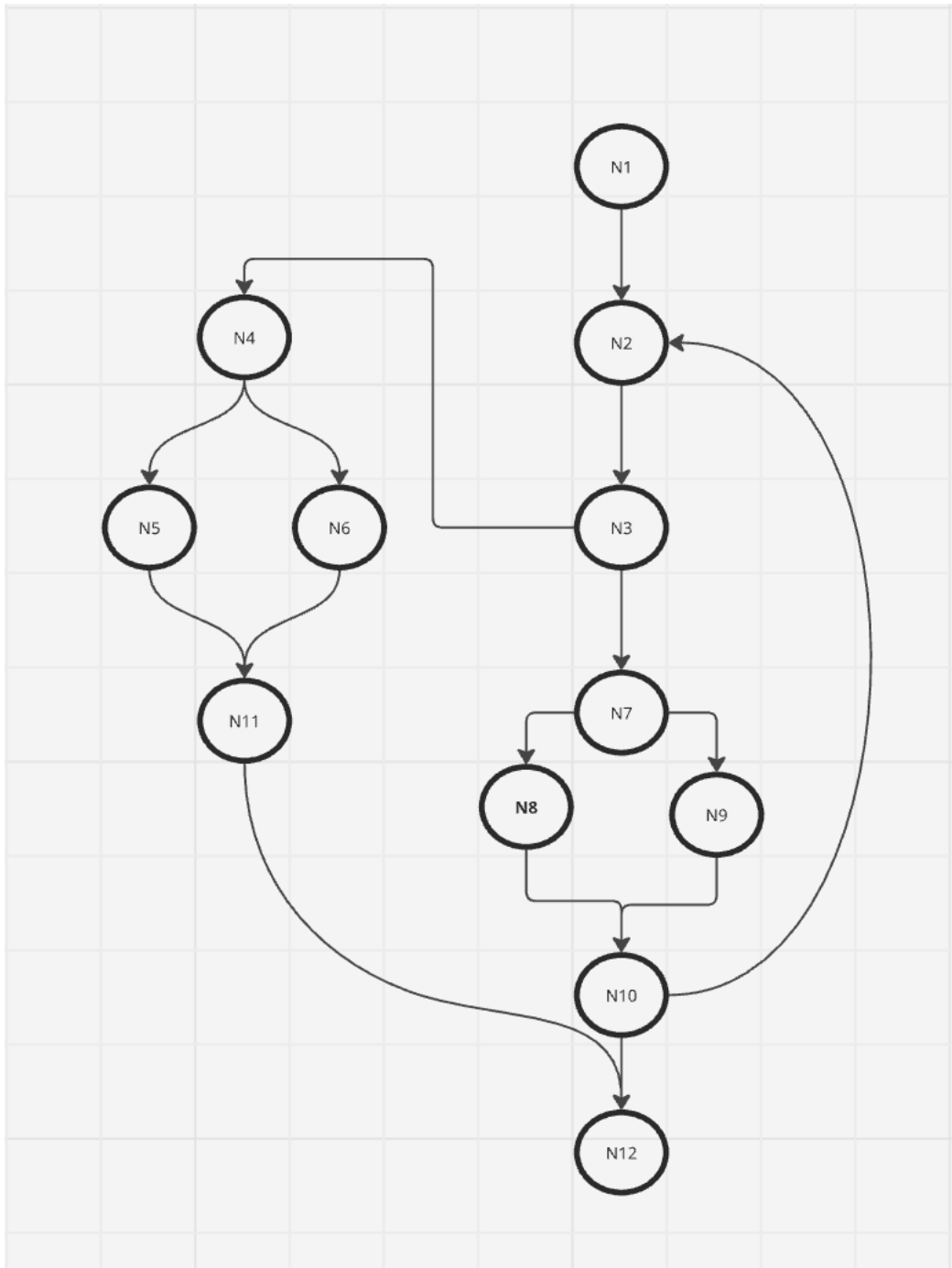
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

Determinar en base al GF del numeral 4

Rutas Independientes:

R1: $1 \rightarrow 2 \rightarrow 6$ (Login fallido).

R2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ (Login exitoso \rightarrow Salir).

R3: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9/10/11/12 \rightarrow 3...$ (Bucle de opciones).

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
- $V(G) = P + 1 = 3 + 1 = 4$
- $V(G) = A - N + 2$
- $V(G) = A - N + 2 = 14$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

*“Título proyecto sistema de automatización de mensajes
e ingreso de datos para fechas importantes”*

Integrantes:

**Alejandro De La Cruz
Santiago Nogales
Ian Escobar**

Fecha 2025-06-16

Prueba caja blanca de describa el requisito funcional

1. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

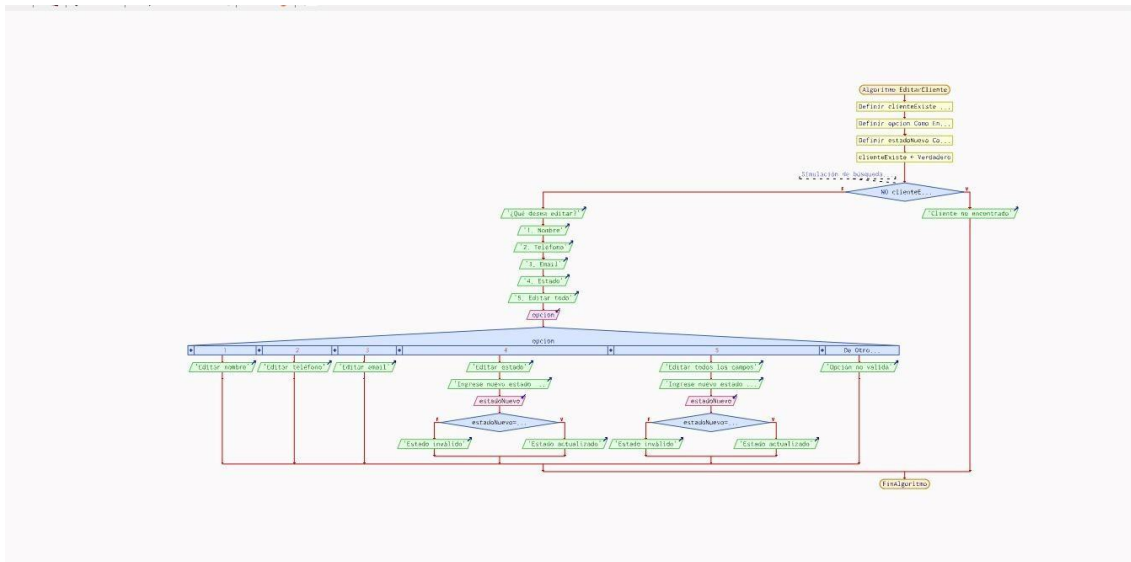
```
public void cargarClientesDesdeCSV() {
    try {
        File file = new File(CSV_FILE);
        if (!file.exists()) {
            crearArchivoCSVVacio();
            return;
        }

        try (CSVReader reader = new CSVReader(new FileReader(file))) {
            List<String[]> records = reader.readAll();
            // Skip header row if it exists
            for (int i = (records.size() > 0 && records.get(0)[0].equals("id")) ? 1 : 0; i < records.size(); i++) {
                String[] record = records.get(i);
                if (record.length >= 6 && !record[0].trim().isEmpty()) {
                    Cliente cliente = new Cliente();
                    cliente.setId(Integer.parseInt(record[0]));
                    cliente.setNombre(record[1]);
                    cliente.setCedula(record[2]);
                    cliente.setTelefono(record[3]);
                    cliente.setEmail(record[4]);
                    cliente.setEstado(record[5]);
                    clientes.add(cliente);
                    if (cliente.getId() >= proximoId.get()) {
                        proximoId.set(cliente.getId() + 1);
                    }
                }
            }
        }
    } catch (Exception e) {
        System.err.println("Error cargando clientes: " + e.getMessage());
    }
}

private void crearArchivoCSVVacio() {
    try (CSVWriter writer = new CSVWriter(new FileWriter(CSV_FILE))) {
        String[] header = {"ID", "Nombre", "Cedula", "Telefono", "Email", "Estado", "Fecha_Registro"};
        writer.writeNext(header);
    } catch (Exception e) {
        System.err.println("Error creating archivo CSV: " + e.getMessage());
    }
}
```

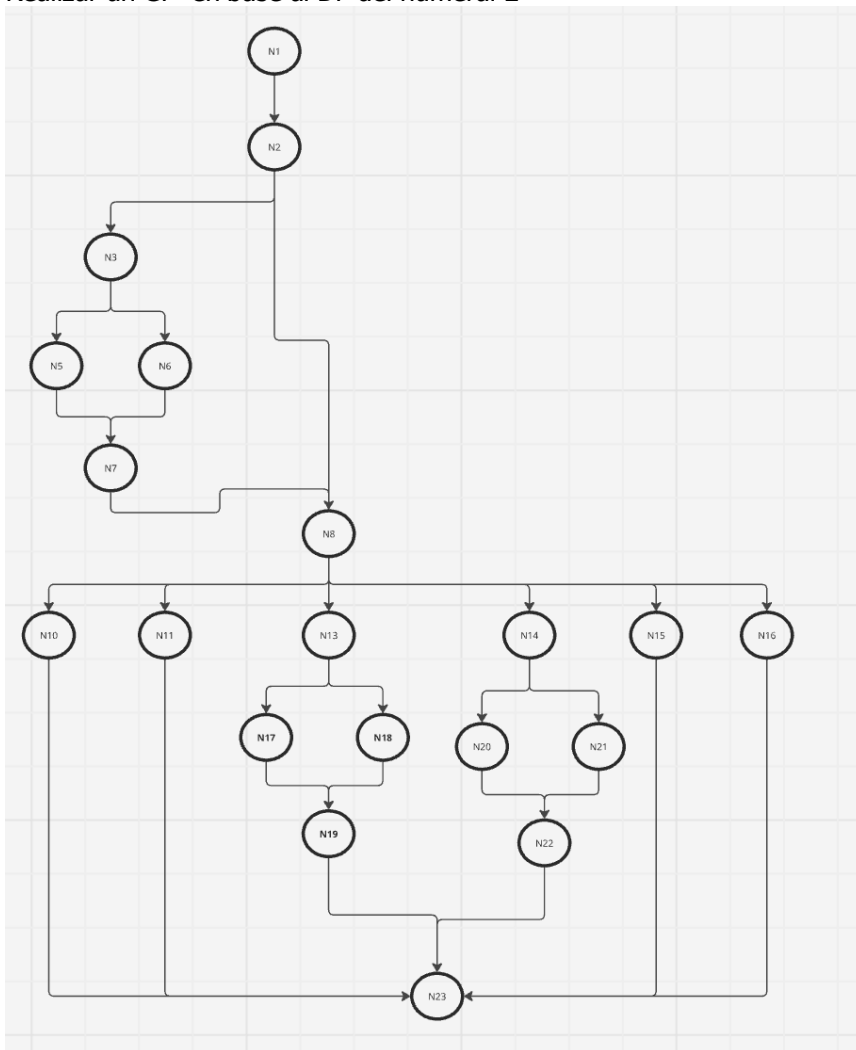
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

Determinar en base al GF del numeral 4

RUTAS

1. **R1:** N5 → N14
 - Camino: Seleccionar opción 0 (Salir) → Terminar programa.
2. **R2:** N5 → N15
 - Camino: Ingresar opción inválida → Mostrar error → Volver al menú.
3. **R3:** N5 → N6 → N16 → N17 → N5
 - Camino: Opción 1 (Agregar cliente) → Ejecutar función → Pausa → Volver al menú.
4. **R4:** N5 → N7 → N18 → N5
 - Camino: Opción 2 (Listar clientes) → Ejecutar → Volver al menú.
5. **R5:** N5 → N9 → N20 → N21 → N5
 - Camino: Opción 4 (Mensaje individual) → Enviar → Pausa → Volver al menú.

Se puede calcular de las siguientes formas:

$$A = 21$$

$$N = 18$$

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = P = 8 + 1 = 9$
- $V(G) = A - N + 2$
 $V(G) = 21 - 18 + 2 = 5.$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos