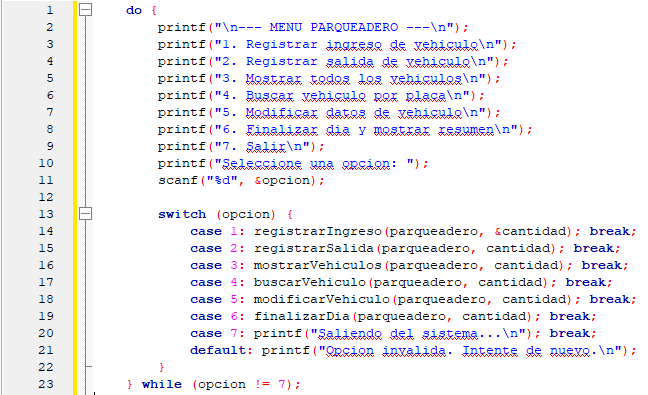
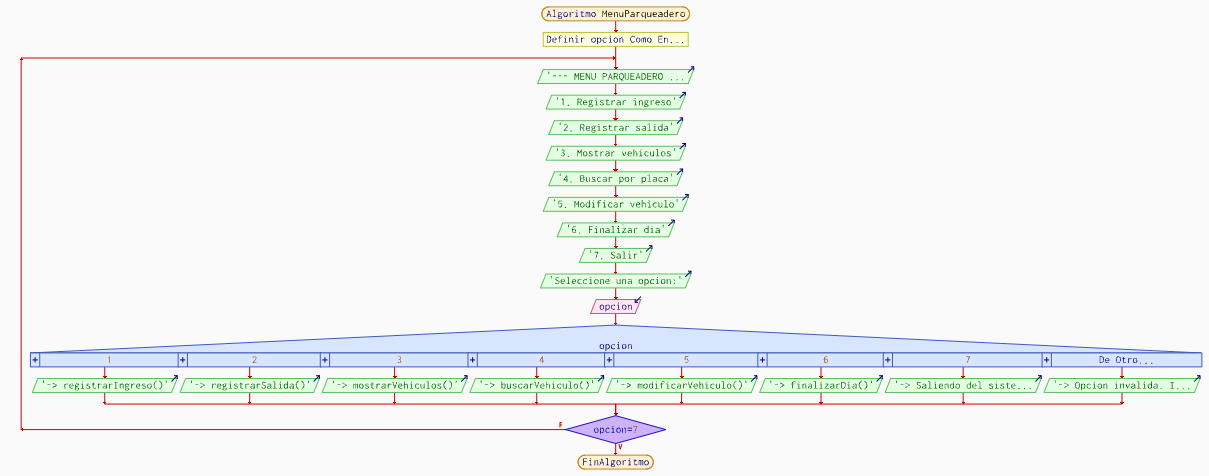
|  |
| --- |
| Prueba de Caja Blanca |
| ***“PARQUEADERO AUTOMATIZADO”*** |
| **Integrantes:**  Jordy Cevallos  Bryan Miguitama  Ana Ulloa  **Fecha: 2025/07/24** |

**Prueba caja blanca de Requisito N° 1**: El programa debe realizar el control general del sistema mediante un menú interactivo que permite acceder a cada una de las funciones disponibles.

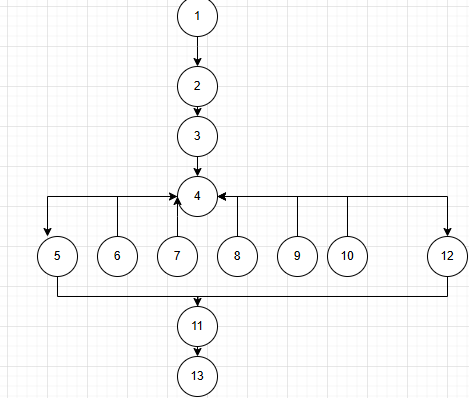
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Inicio → Mostrar menú → Leer opción = 1 → registrarIngreso() → Repetir

R2:  
Inicio → Mostrar menú → Leer opción = 2 → registrarSalida() → Repetir

R3:  
Inicio → Mostrar menú → Leer opción = 3 → mostrarVehiculos() → Repetir

R4:  
Inicio → Mostrar menú → Leer opción = 4 → buscarVehiculo() → Repetir

R5:  
Inicio → Mostrar menú → Leer opción = 5 → modificarVehiculo() → Repetir

R6:  
Inicio → Mostrar menú → Leer opción = 6 → finalizarDia() → Repetir

R7:  
Inicio → Mostrar menú → Leer opción = 7 → “Salir del sistema” → Fin

R8:  
Inicio → Mostrar menú → Leer opción inválida → Mostrar “Opción inválida” → Repetir

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

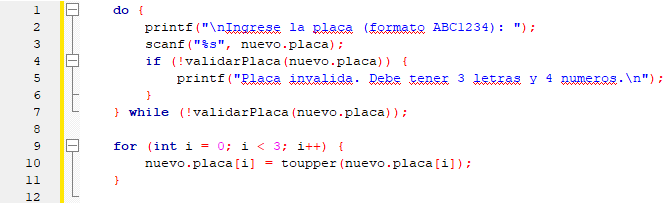
● V(G) = número de nodos predicados (decisiones) + 1  
V(G) = 2 + 1 = 3

● V(G) = A – N + 2  
V(G) = 12 – 11 + 2 = 3

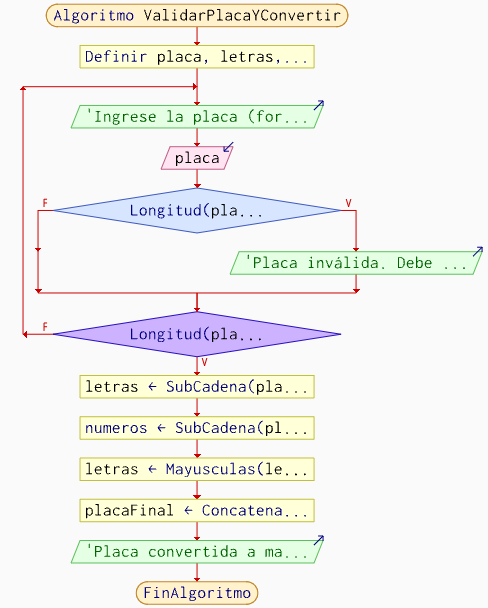
DONDE:  
P: Número de nodos predicado = 2  
A: Número de aristas = 12  
N: Número de nodos = 11

**Prueba caja blanca de Requisito N° 2**: El programa debe realizar la conversión de las placas ingresadas a mayúsculas.

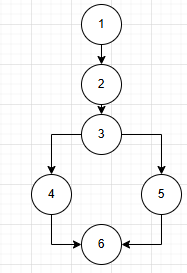
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Inicio → Ingresar placa válida → Convertir letras a mayúsculas → Fin

R2:  
Inicio → Ingresar placa inválida → Mostrar mensaje → Repetir hasta válida → Convertir letras a mayúsculas → Fin

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

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

V(G) = 1 + 1 = 2

● V(G) = A – N + 2

V(G) = 5 – 4 + 2 = 3

DONDE:

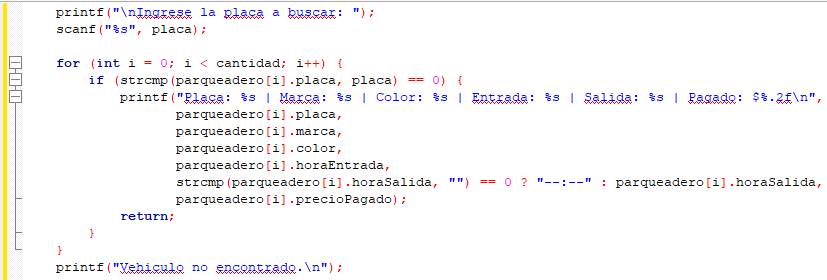
P: Número de nodos predicado = 1

A: Número de aristas = 5

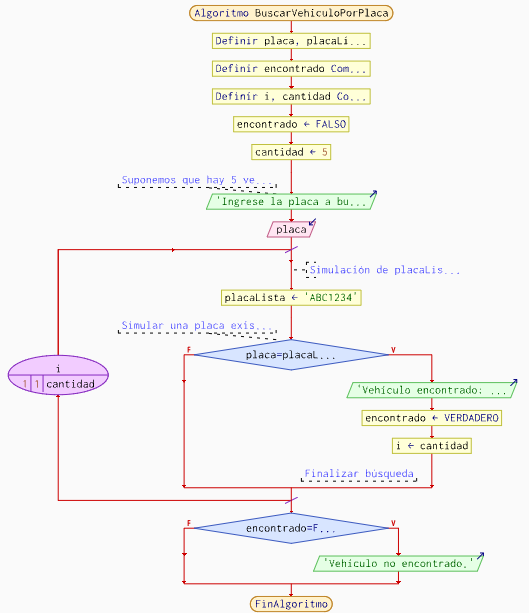
N: Número de nodos = 4

**Prueba caja blanca de Requisito N° 3**: El programa debe realizar la búsqueda de un vehículo en la lista mediante su placa para validar su existencia.

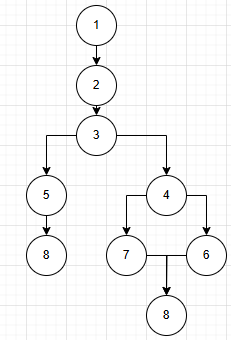
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Placa encontrada en el primer intento → Mostrar datos del vehículo

R2:  
Placa encontrada luego de varias iteraciones → Mostrar datos del vehículo

R3:  
Placa no encontrada → Mostrar mensaje "Vehiculo no encontrado"

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

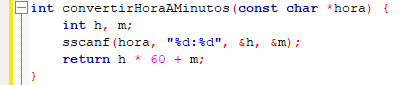
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 2 + 1 = 3

● V(G) = A – N + 2 V(G)= 9 – 8 + 2 = 3

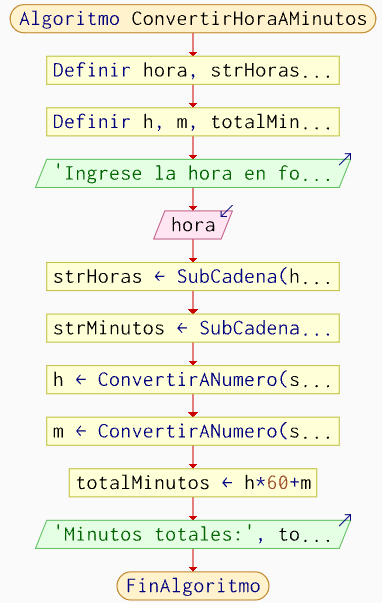
DONDE:  
P: Número de nodos predicado = 2  
A: Número de aristas = 9  
N: Número de nodos = 8

**Prueba caja blanca de Requisito N° 4**: El programa debe realizar la conversión de la hora en formato "HH:MM" a minutos para facilitar el cálculo de permanencia.

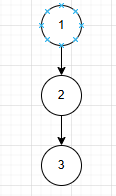
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Leer hora HH:MM → Convertir → Devolver total en minutos

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

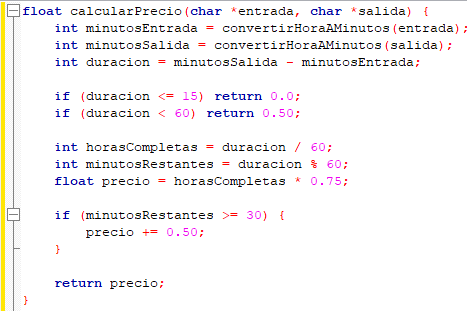
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 0 + 1 = 1

● V(G) = A – N + 2 V(G)= 2 – 1 + 2 = 3

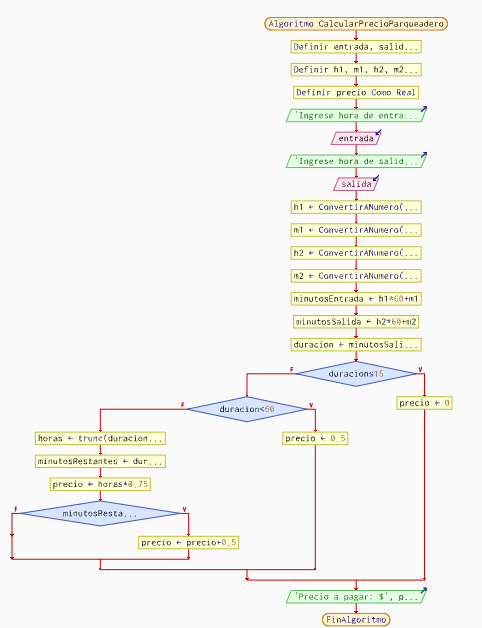
DONDE:  
P: Número de nodos predicado = 0  
A: Número de aristas = 2  
N: Número de nodos = 1

**Prueba caja blanca de Requisito N° 5**: El programa debe realizar el cálculo del precio a pagar según el tiempo de permanencia del vehículo en el parqueadero.

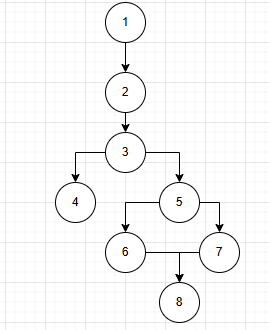
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Tiempo ≤ 15 → precio = 0.0

R2:  
Tiempo < 60 → precio = 0.50

R3:  
Tiempo ≥ 60 con minutosRestantes < 30 → precio = horas \* 0.75

R4:  
Tiempo ≥ 60 con minutosRestantes ≥ 30 → precio = horas \* 0.75 + 0.50

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

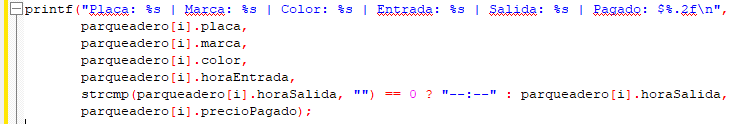
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 3 + 1 = 4

● V(G) = A – N + 2 V(G)= 10 – 8 + 2 = 4

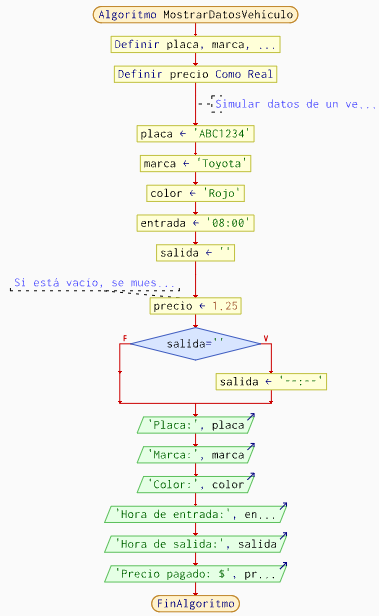
DONDE:  
P: Número de nodos predicado = 3  
A: Número de aristas = 10  
N: Número de nodos = 8

**Prueba caja blanca de Requisito N° 6**: El programa debe realizar la visualización de la información completa de un vehículo, incluyendo placa, hora de entrada, salida y valor pagado

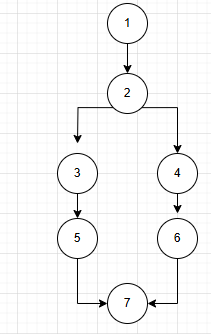
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
horaSalida tiene valor → mostrar datos completos

R2:  
horaSalida está vacío → mostrar "--:--" en vez de salida

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

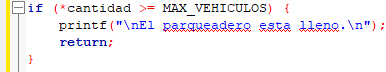
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 1 + 1 = 2

● V(G) = A – N + 2 V(G)= 4 – 3 + 2 = 3

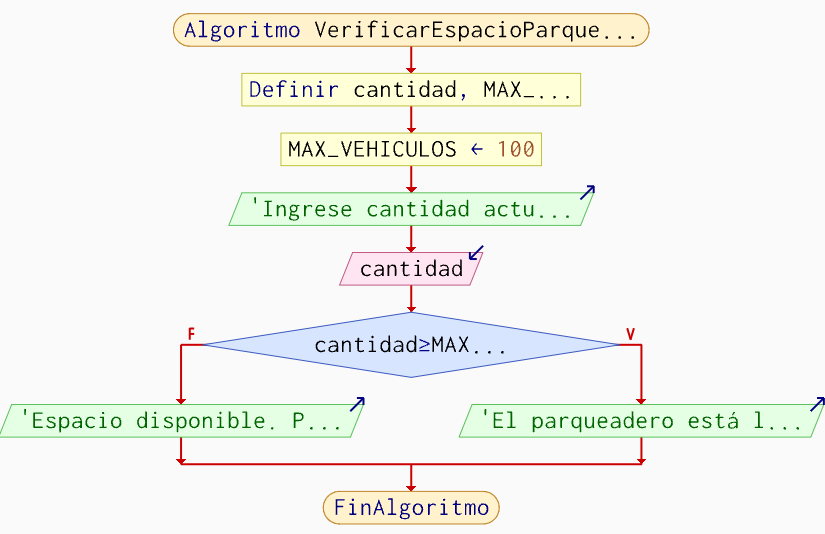
DONDE:  
P: Número de nodos predicado = 1  
A: Número de aristas = 4  
N: Número de nodos = 3

**Prueba caja blanca de Requisito N°7**: El programa debe realizar el registro de entrada de un vehículo validando que no exista un registro anterior sin salida.

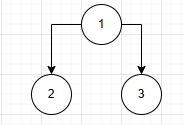
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Cantidad menor al máximo → permitir registro

R2:  
Cantidad igual o mayor al máximo → mostrar “parqueadero lleno” y salir

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

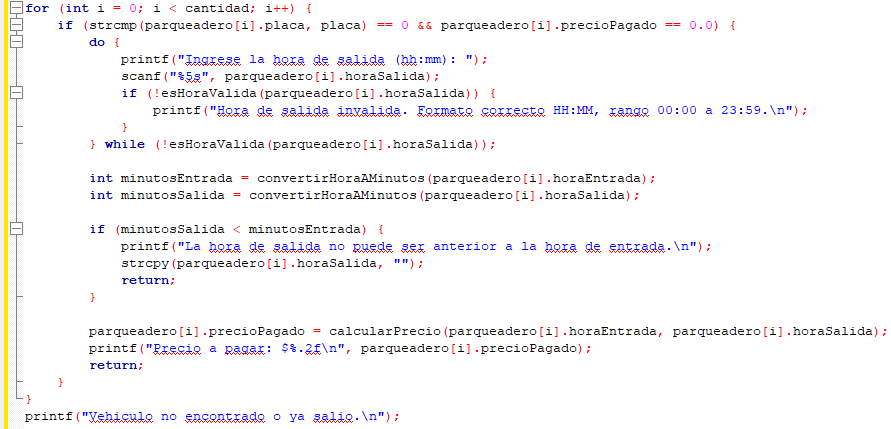
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 1 + 1 = 2

● V(G) = A – N + 2 V(G)= 3 – 2 + 2 = 3

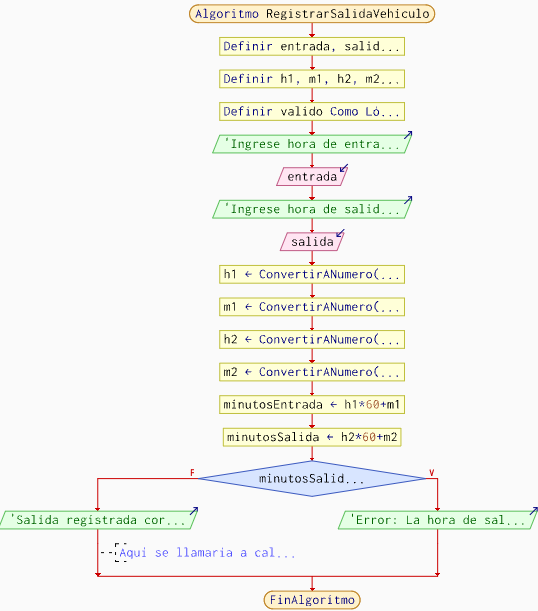
DONDE:  
P: Número de nodos predicado = 1  
A: Número de aristas = 3  
N: Número de nodos = 2

**Prueba caja blanca de Requisito N° 8**: El programa debe realizar el registro de salida de un vehículo y calcular automáticamente el valor a pagar

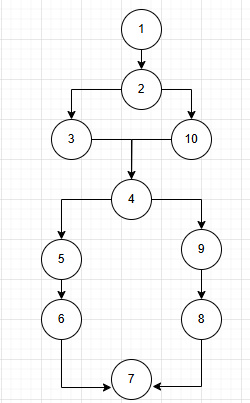
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Vehículo encontrado → hora válida → salida posterior → calcular precio → mostrar precio

R2:  
Vehículo encontrado → hora inválida → pedir de nuevo → luego calcular

R3:  
Vehículo encontrado → salida anterior a entrada → mensaje de error

R4:  
Vehículo no encontrado o ya salió → mostrar mensaje

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

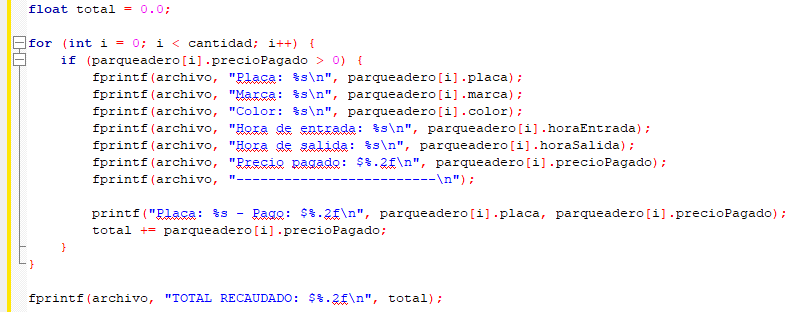
● V(G) = número de nodos predicados(decisiones)+1 V(G)= 4 + 1 = 5

● V(G) = A – N + 2 V(G)= 13 – 10 + 2 = 5

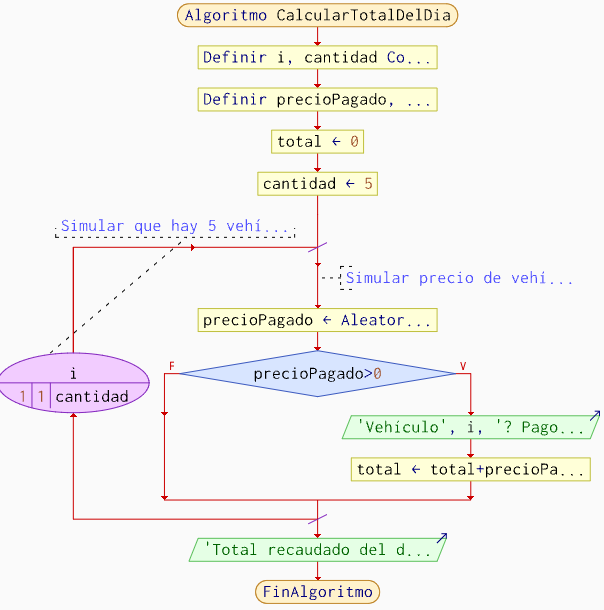
DONDE:  
P: Número de nodos predicado = 4  
A: Número de aristas = 13  
N: Número de nodos = 10

**Prueba caja blanca de Requisito N° 9**: El programa debe realizar el cálculo del total recaudado en el día, mostrando el detalle de los pagos realizados

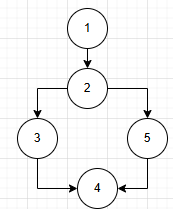
1. **CÓDIGO FUENTE**

****

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

****

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

****

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

R1:  
Vehículo con pago > 0 → imprimir datos → sumar total

R2:  
Vehículo sin pago → no imprimir → no sumar

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

● V(G) = número de nodos predicados(decisiones)+1 V(G)= 1 + 1 = 2

● V(G) = A – N + 2 V(G)= 6 – 5 + 2 = 3

DONDE:  
P: Número de nodos predicado = 1  
A: Número de aristas = 6  
N: Número de nodos = 5