

# Prueba de Caja Blanca

---

*“Sistema de inventario de frutos secos”*

**Integrantes:**

– Alvear

Alexander

– Campoverde

Anthony

– Velecela Mateo

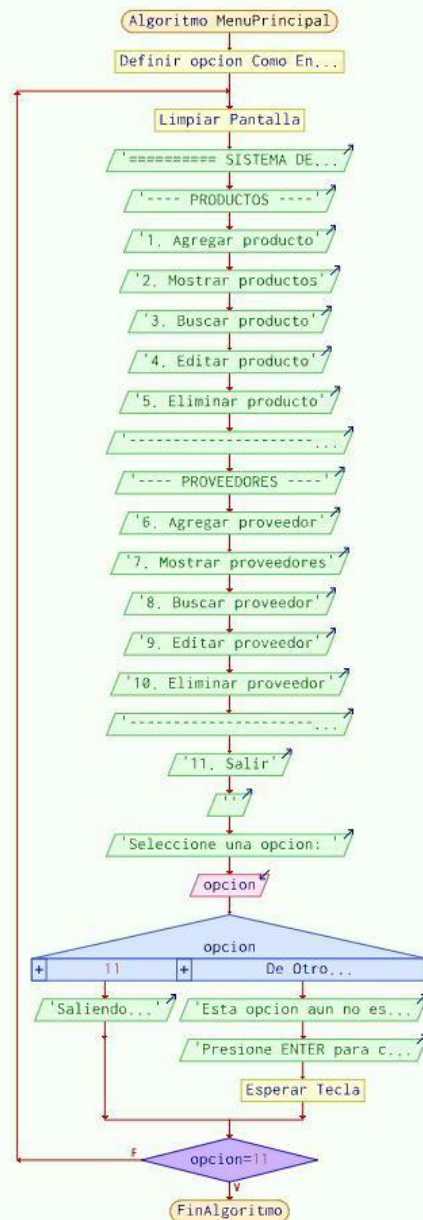
**Fecha: 2025/07/23**

**Prueba caja blanca de Requisito N° 1:** El sistema deberá iniciar con un menú principal con opciones separadas y visualmente claras para productos y proveedores

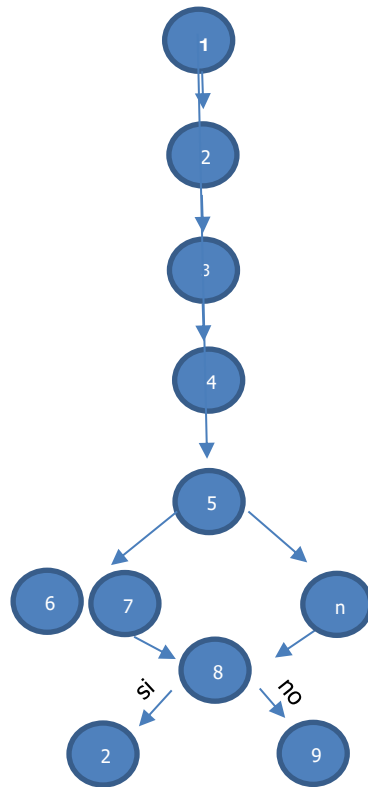
## 1. CÓDIGO FUENTE

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #ifdef _WIN32
5      #define CLEAR "cls"
6  #else
7      #define CLEAR "clear"
8  #endif
9
10 void imprimirMenu();
11 void pausa();
12 void limpiarBuffer();
13
14 void limpiarBuffer() {
15     int c;
16     while ((c = getchar()) != '\n' && c != EOF);
17 }
18
19 void pausa() {
20     printf("\nPresione ENTER para continuar...");
21     limpiarBuffer();
22 }
23
24 void imprimirMenu() {
25     printf("\n===== SISTEMA DE INVENTARIO Y PROVEEDORES =====\n");
26     printf("---- PRODUCTOS ----\n");
27     printf("1. Agregar producto\n");
28     printf("2. Mostrar productos\n");
29     printf("3. Buscar producto\n");
30     printf("4. Editar producto\n");
31     printf("5. Eliminar producto\n");
32     printf("-----\n");
33     printf("---- PROVEEDORES ----\n");
34     printf("6. Agregar proveedor\n");
35     printf("7. Mostrar proveedores\n");
36     printf("8. Buscar proveedor\n");
37     printf("9. Editar proveedor\n");
38     printf("10. Eliminar proveedor\n");
39     printf("-----\n");
40     printf("11. Salir\n");
41     printf("Seleccione una opcion: ");
42 }
```

## 2. DIAGRAMA DE FLUJO (DF) PSEINT



### 3. GRAFO DE FLUJO (GF)



### 4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

#### RUTAS

R1: (1 → 2 → 3 → 4 → 5 → 6 → 8 → 2)

R2: (1 → 2 → 3 → 4 → 5 → 7 → 8 → 2)

R3: (1 → 2 → 3 → 4 → 5 → n → 8 → 2)

R4: (1 → 2 → 3 → 4 → 5 → 6 → 8 → 9)

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

Método 1 (usando nodos de decisión):

$$V(G) = P + 1$$

- Nodos de decisión:
  - Evaluación del switch → múltiples caminos
  - Condición del while (opción != 11)  
→  $P = 2$   
 $V(G) = 2 + 1 = 3$

Método 2 (usando aristas y nodos):

$$V(G) = A - N + 2$$

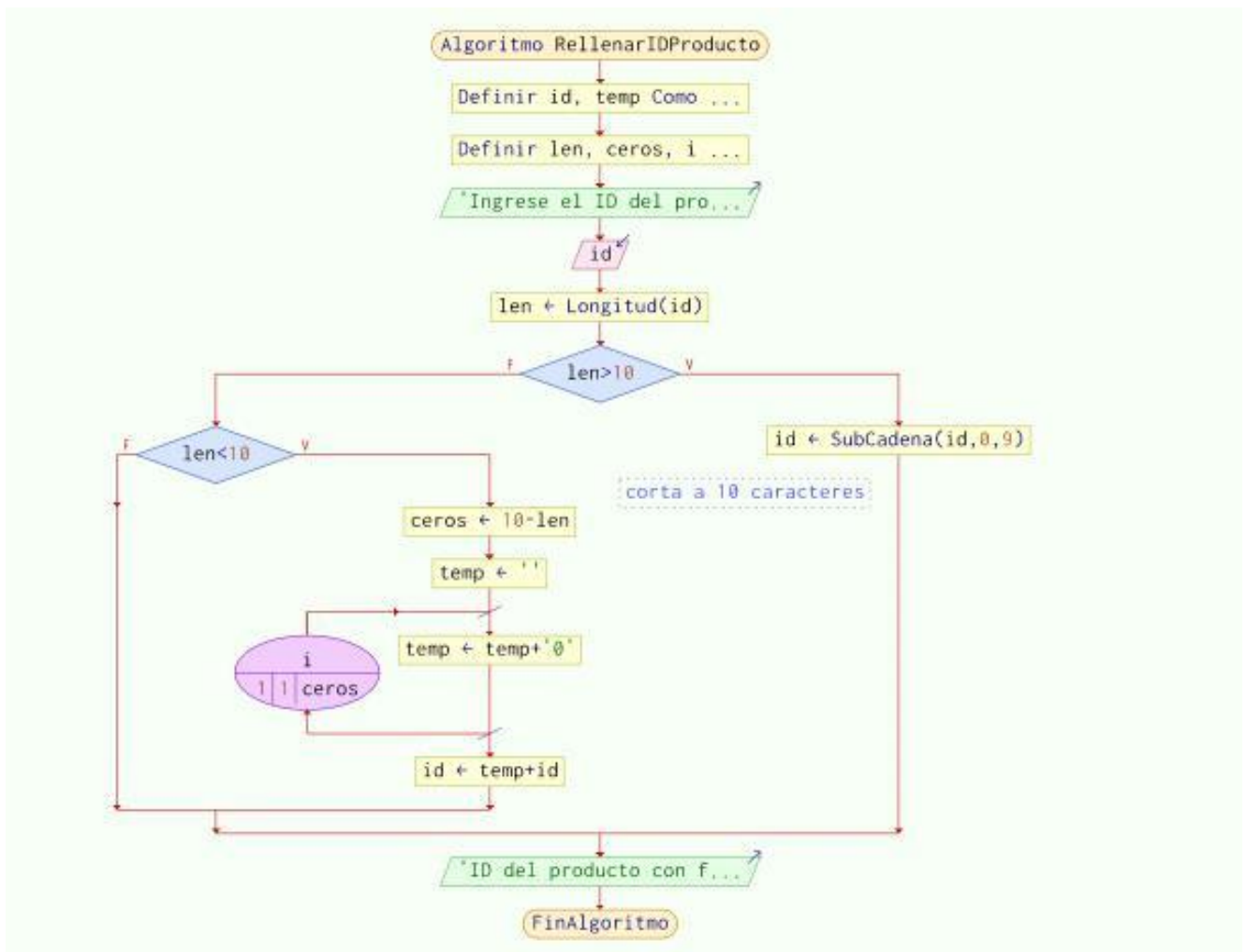
- A (aristas)  $\approx 9$
- N (nodos)  $\approx 8$   
 $V(G) = 9 - 8 + 2 = 3$

## Prueba caja blanca de Requisito N° 2: Agregar productos con ID de 10 dígitos

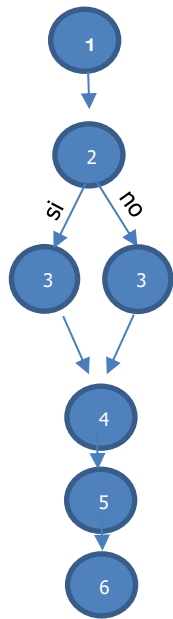
### 1. CÓDIGO FUENTE

```
29     printf("3. Buscar producto\n");
30     printf("4. Editar producto\n");
31     printf("5. Eliminar producto\n");
32     printf("-----\n");
33     printf("---- PROVEEDORES ----\n");
34     printf("6. Agregar proveedor\n");
35     printf("7. Mostrar proveedores\n");
36     printf("8. Buscar proveedor\n");
37     printf("9. Editar proveedor\n");
38     printf("10. Eliminar proveedor\n");
39     printf("-----\n");
40     printf("11. Salir\n");
41     printf("Seleccione una opcion: ");
42 }
43
44 int main() {
45     int opcion;
46     do {
47         system("CLEAR");
48         imprimirMenu();
49         if (scanf("%d", &opcion) != 1) {
50             printf("Entrada invalida.\n");
51             limpiarBuffer();
52             opcion = 0;
53             pausa();
54             continue;
55         }
56         limpiarBuffer();
57
58         switch (opcion) {
59             case 11:
60                 printf("Saliendo...\n");
61                 break;
62             default:
63                 printf("Esta opción aun no estas implementada.\n");
64                 pausa();
65                 break;
66         }
67     } while (opcion != 11);
68
69     return 0;
70 }
```

## 2. DIAGRAMA DE FLUJO (DF) PSEINT



### 3. GRAFO DE FLUJO (GF)



### 4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

#### RUTAS

- R1: flujo completo sin error (hasta guardar).
- R2: junto con error por límite de productos.
- R3: error por proveedor no existente.
- R4: otro error (entrada inválida).

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $P = 3$  (condición de límite, proveedor, validaciones múltiples)
- $V(G) = P + 1 = 4$
- También,  $A - N + 2 \approx 4$