

# Prueba de Caja Blanca

---

*“Implementación de un sistema e inventario para el local Chaskibots”*

**Integrantes:**

Cobeña Bravo Davis Ariel  
German Luna Jhon Alexis  
Ramos Vilca Paulo Alejandro

**Fecha Enero 2025**

## Prueba caja blanca de Inicio de sesion

### 1. CÓDIGO FUENTE

```
#include <stdio.h>
#include <string.h>

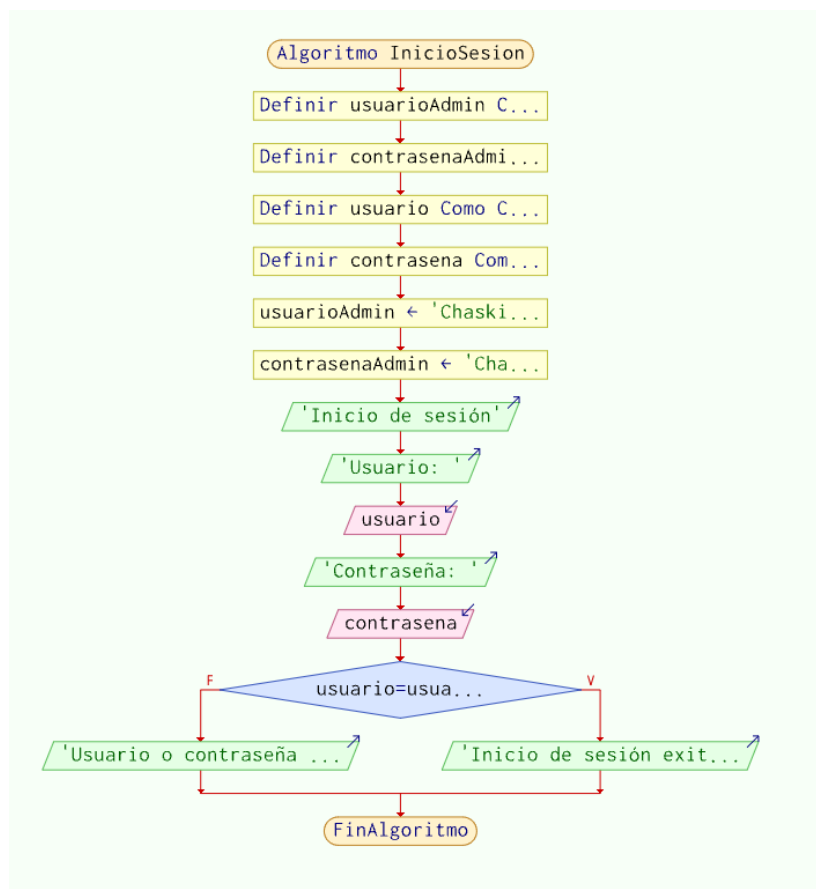
int main() {
    // Usuario y contraseña predefinidos
    char usuarioAdmin[] = "ChaskiBots1";
    char contraseñaAdmin[] = "ChaskiBots1";

    // Variables para ingresar datos
    char usuario[20];
    char contraseña[20];

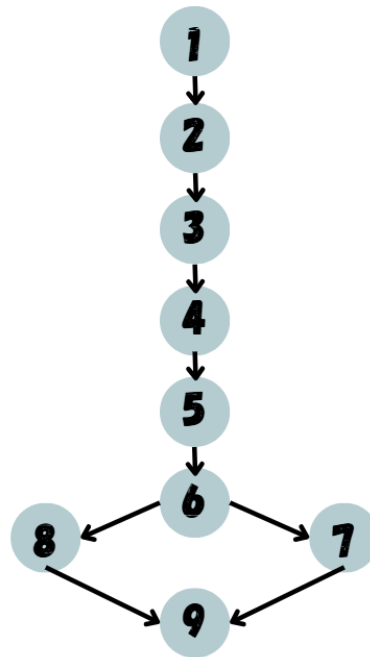
    printf("Inicio de sesión\n");
    printf("Usuario: ");
    scanf("%s", usuario);
    printf("Contraseña: ");
    scanf("%s", contraseña);

    // Verificar credenciales
    if (strcmp(usuario, usuarioAdmin) == 0 && strcmp(contraseña, contraseñaAdmin) == 0) {
        printf("Inicio de sesión exitoso.\n");
    } else {
        printf("Usuario o contraseña incorrectos.\n");
    }
}
```

### 2. DIAGRAMA DE FLUJO (DF)



### 3. GRAFO DE FLUJO (GF)



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

#### RUTAS

**R1:** 1, 2, 3, 4, 5, 6, 7, 9

**R2:** 1, 2, 3, 4, 5, 6, 8, 9

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$   
 $V(G) = 9 - 9 + 2 = 2$

DONDE:

**P:** 1

**A:** 9

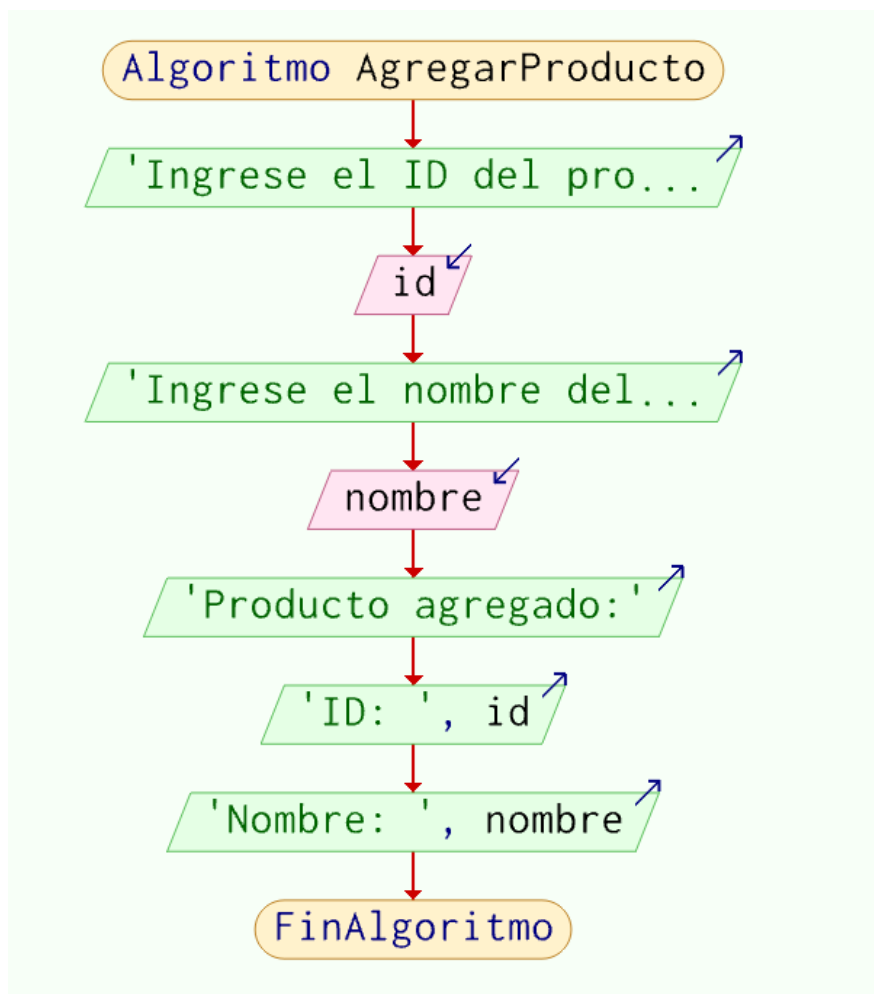
**N:** 9

## Prueba caja blanca de Añadir un producto

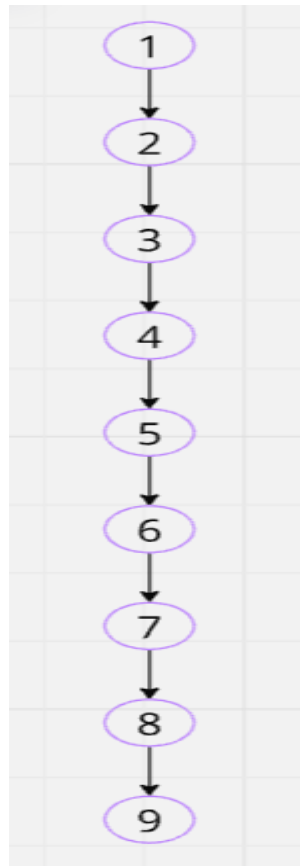
### 1. CÓDIGO FUENTE

```
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      int id;
5      char nombre[50];
6
7      // Pedir al usuario que ingrese el nombre y el ID del producto
8      printf("Ingrese el ID del producto: ");
9      scanf("%d", &id);
10
11     printf("Ingrese el nombre del producto: ");
12     scanf("%s", &nombre);
13     // Mostrar los datos ingresados
14     printf("\nProducto agregado:\n");
15     printf("ID: %d\n", id);
16     printf("Nombre: %s\n", nombre);
17     return 0;
18 }
```

### 2. DIAGRAMA DE FLUJO (DF)



### 3. Grafo de flujo (GF)



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

#### RUTAS

**R1:** 1,2,3,4,5,6,7,8,9

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos prediados(decisiones)} + 1$   
 $V(G) = 0 + 1 = 1$
- $V(G) = 8 - 9 + 2$   
 $V(G) = -1 + 2 = 1$

DONDE:

**P:** 0

**A:** 8

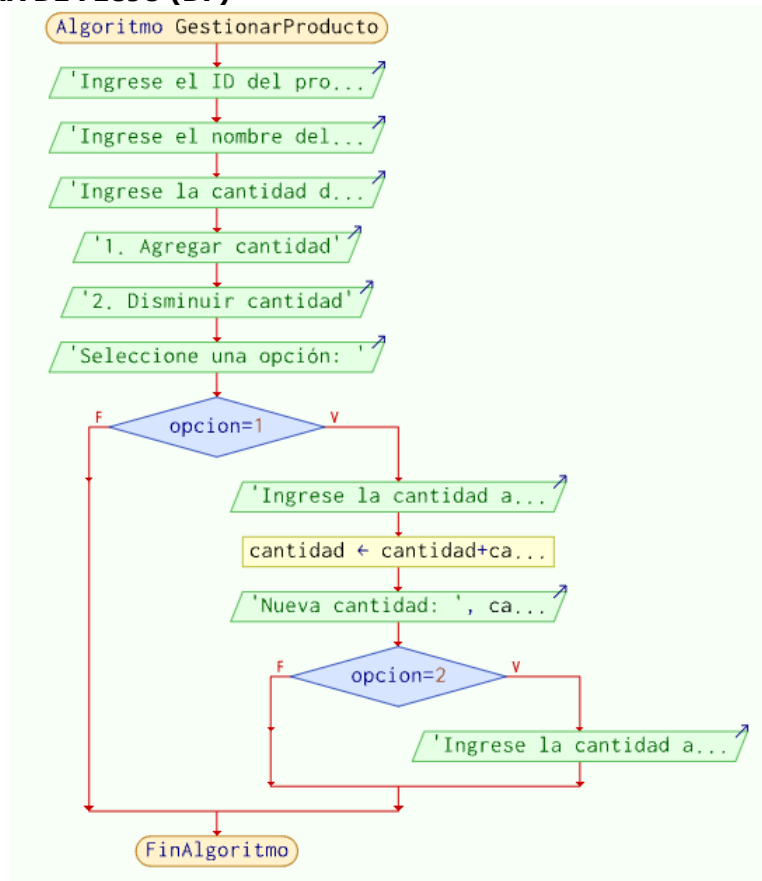
**N:** 9

## Prueba caja blanca de Modificar un producto

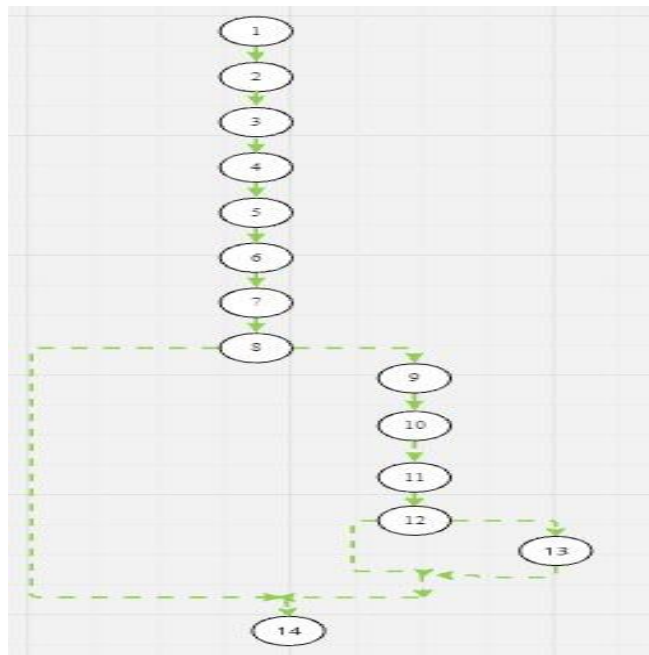
### 1. CÓDIGO FUENTE

```
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      int id;
5      char nombre[50];
6      int cantidad;
7      printf("Ingrese el ID del producto: ");
8      scanf("%d", &id);
9      printf("Ingrese el nombre del producto: ");
10     scanf("%s", &nombre);
11     printf("Ingrese la cantidad del producto: ");
12     scanf("%d", &cantidad);
13     int opcion;
14     do {
15         printf("\n¿Desea agregar o disminuir la cantidad?\n");
16         printf("1. Agregar cantidad\n");
17         printf("2. Disminuir cantidad\n");
18         printf("3. Salir\n");
19         printf("Seleccione una opción: ");
20         scanf("%d", &opcion);
21         if (opcion == 1) {
22             printf("Ingrese la cantidad a agregar: ");
23             scanf("%d", &cantidad);
24             cantidad += cantidad;
25             printf("Nueva cantidad: %d\n", cantidad);
26         } else if (opcion == 2) {
27             printf("Ingrese la cantidad a disminuir: ");
28             scanf("%d", &cantidad);
29             if (cantidad <= cantidad) {
30                 cantidad -= cantidad;
31                 printf("Nueva cantidad: %d\n", cantidad);
32             } else {
33                 printf("No puedes reducir la cantidad a un valor negativo.\n");
34             }
35         } else if (opcion == 3) {
36             printf("Saliendo...\n");
37         } else {
38             printf("Opción inválida. Intenta nuevamente.\n");
39         }
40     } while (opcion != 3);
41     return 0;
42 }
```

### 2. DIAGRAMA DE FLUJO (DF)



### 3. Grafo de flujo (GF)



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

#### RUTAS

**R1:** 1, 2, 3, 4, 5, 6, 7, 8, 14

**R2:** 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14

**R3:** 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 15 - 14 + 2 = 3$

DONDE:

**P:** 2

**A:** 15

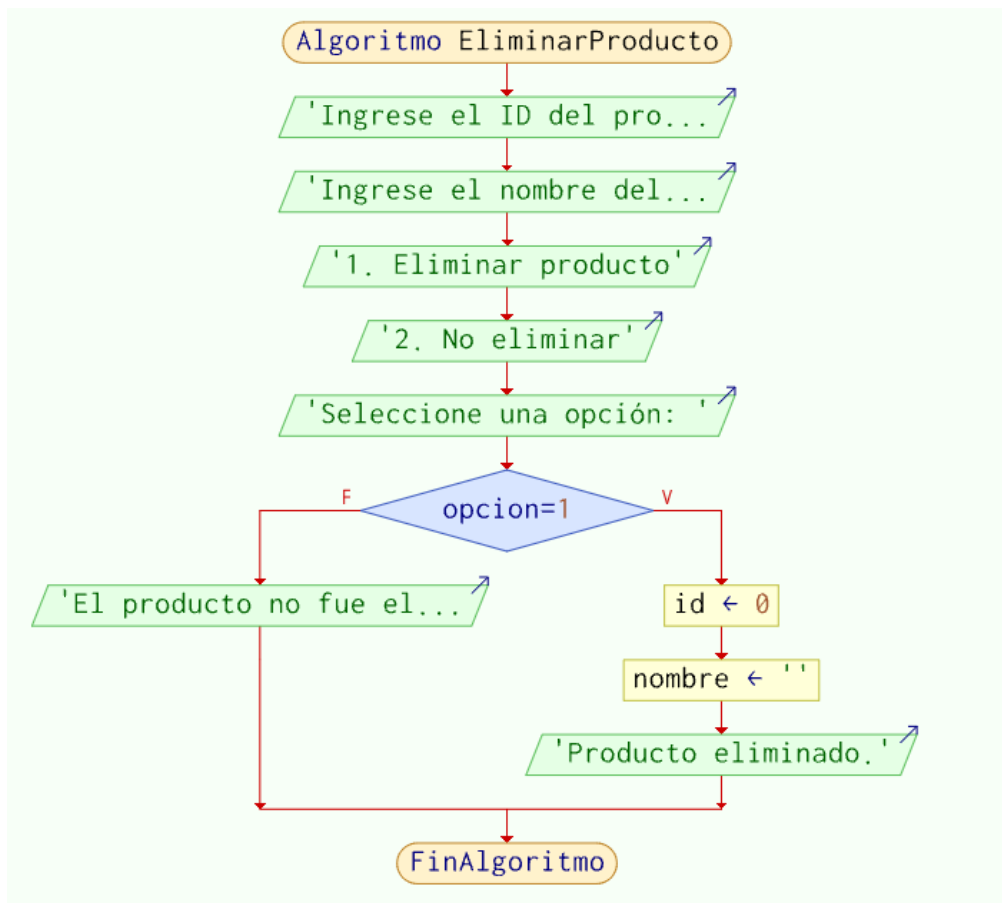
**N:** 14

## Prueba caja blanca de Eliminar un producto

### 1. CÓDIGO FUENTE

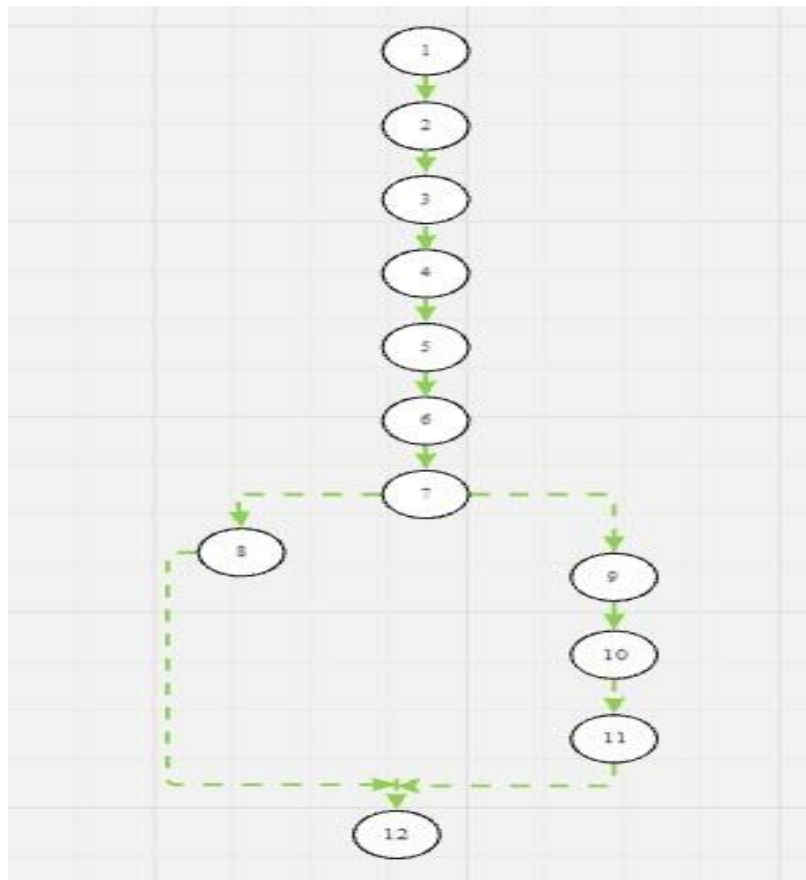
```
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      int id;
5      char nombre[50];
6      int opcion;
7      printf("Ingrese el ID del producto: ");
8      scanf("%d", &id);
9      printf("Ingrese el nombre del producto: ");
10     scanf ("%s", &nombre);
11     printf("1. Eliminar producto\n");
12     printf("2. No eliminar\n");
13     printf("Seleccione una opción: ");
14     scanf("%d", &opcion);
15     if (opcion == 1) {
16         // Eliminamos el producto (borramos los datos)
17         id = 0;
18         strcpy(nombre, "");
19         printf("\nProducto eliminado.\n");
20     } else {
21         printf("\nEl producto no fue eliminado.\n");
22     }
23     return 0;
24 }
```

### 2. DIAGRAMA DE FLUJO (DF)





### 3. Grafo de flujo (GF)



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

#### RUTAS

**R1:** 1, 2, 3, 4, 5, 6, 7, 8, 12

**R2:** 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$   
 $V(G) = 12 - 12 + 2 = 2$

DONDE:

**P:** 1

**A:** 12

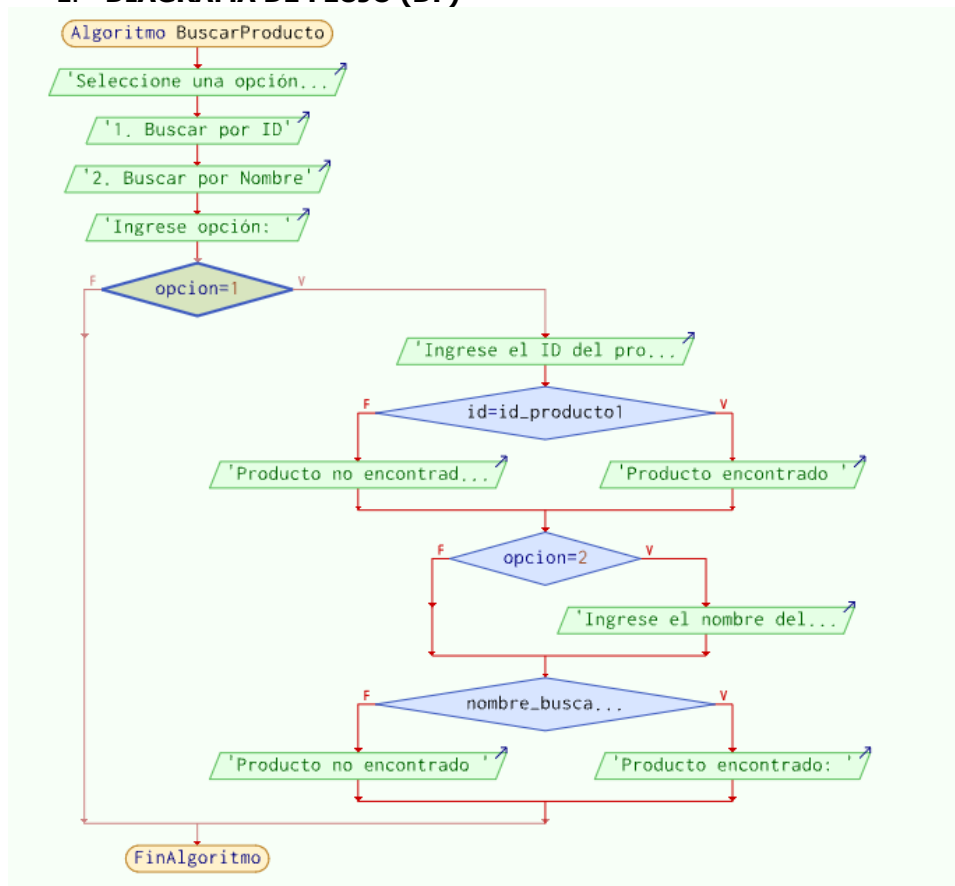
**N:** 12

## Prueba caja blanca de Buscar un producto

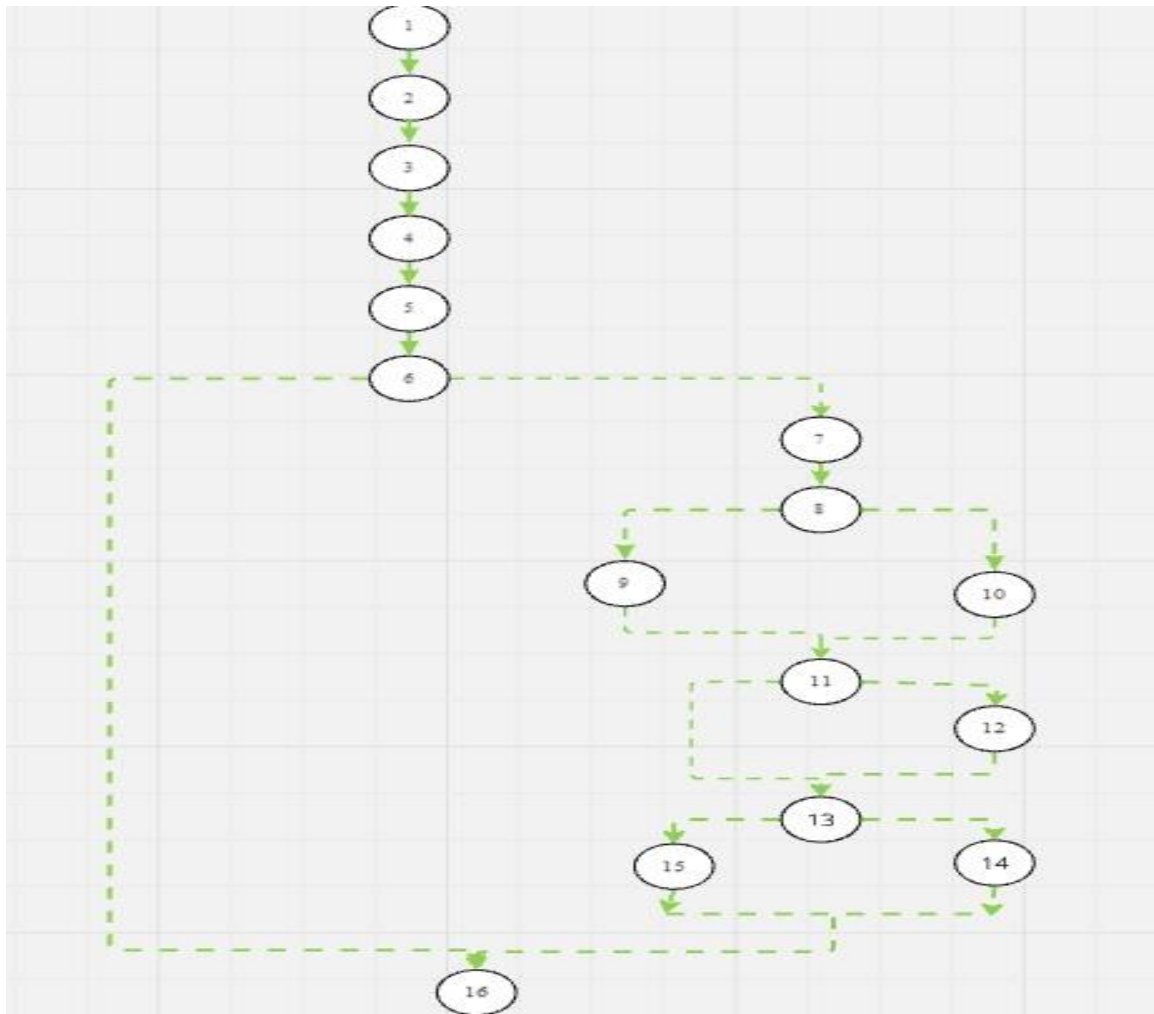
### 1. CÓDIGO FUENTE

```
1  #include <stdio.h>
2  #include <string.h>
3  int main() {
4      int id;
5      char nombre[50];
6      int opcion;
7      printf("Ingrese el ID del producto: ");
8      scanf("%d", &id);
9      printf("Ingrese el nombre del producto: ");
10     scanf ("%s", &nombre);
11     printf ("1. Eliminar producto\n");
12     printf("2. No eliminar\n");
13     printf("Seleccione una opción: ");
14     scanf("%d", &opcion);
15     if (opcion == 1) {
16         printf("\nProducto eliminado.\n");
17     } else {
18         printf("\nEl producto no fue eliminado.\n");
19     }
20     return 0;
21 }
```

### 2. DIAGRAMA DE FLUJO (DF)



### 3. Grafo de flujo (GF)



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

#### RUTAS

**R1:** 1, 2, 3, 4, 5, 6, 16

**R2:** 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13, 15, 16

**R3:** 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16

**R4:** 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 4 + 1 = 5$
- $V(G) = N - A + 2$   
 $V(G) = 19 - 16 + 2 = 5$

DONDE:

**P:** 4

**A:** 16

**N:** 19