

# Prueba de Caja Blanca

---

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

**Integrantes:**  
**Davis Cobeña**  
**Jhon German**  
**Paulo Ramos**

**Fecha 2025 – 01 – 16**

## Prueba caja blanca de Inicio de sesion

### 1. CÓDIGO FUENTE

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

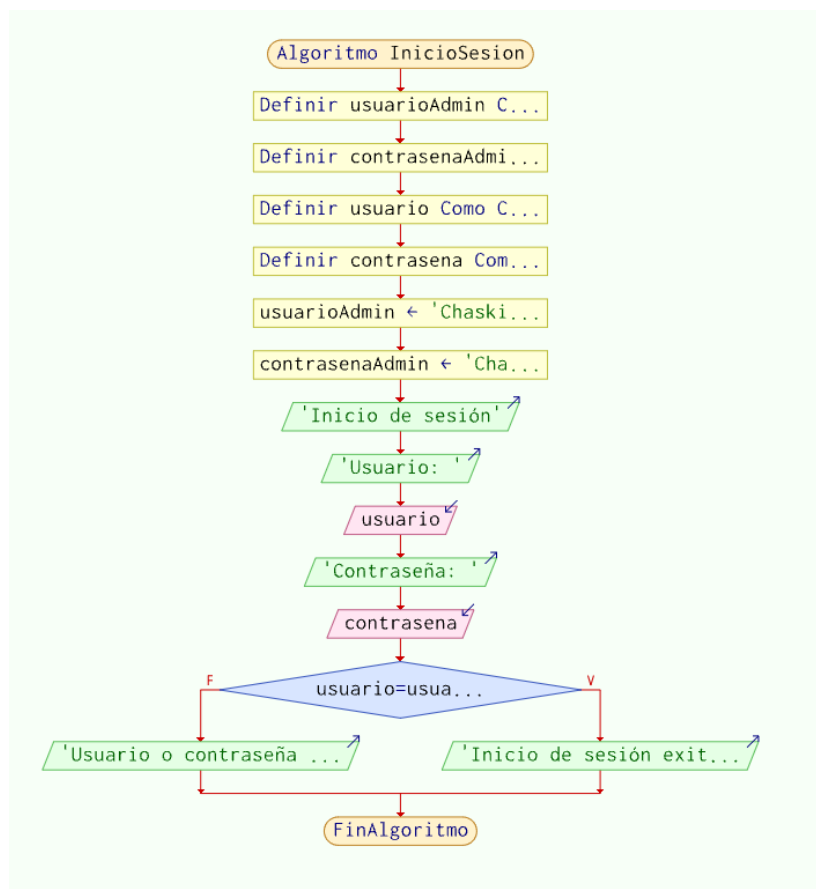
int main() {
    // Usuario y contraseña predefinidos
    char usuarioAdmin[] = "ChaskiBots1";
    char contraseñaAdmin[] = "ChaskiBots1";

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

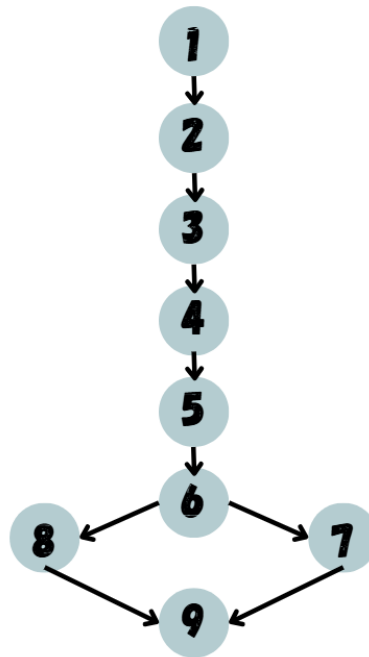
    printf("Inicio de sesión\n");
    printf("Usuario: ");
    scanf("%s", usuario);
    printf("Contraseña: ");
    scanf("%s", contraseña);

    // 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

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

// Estructura para representar un producto
struct Producto {
    int id;
    char nombre[50];
};

int main() {
    struct Producto producto;

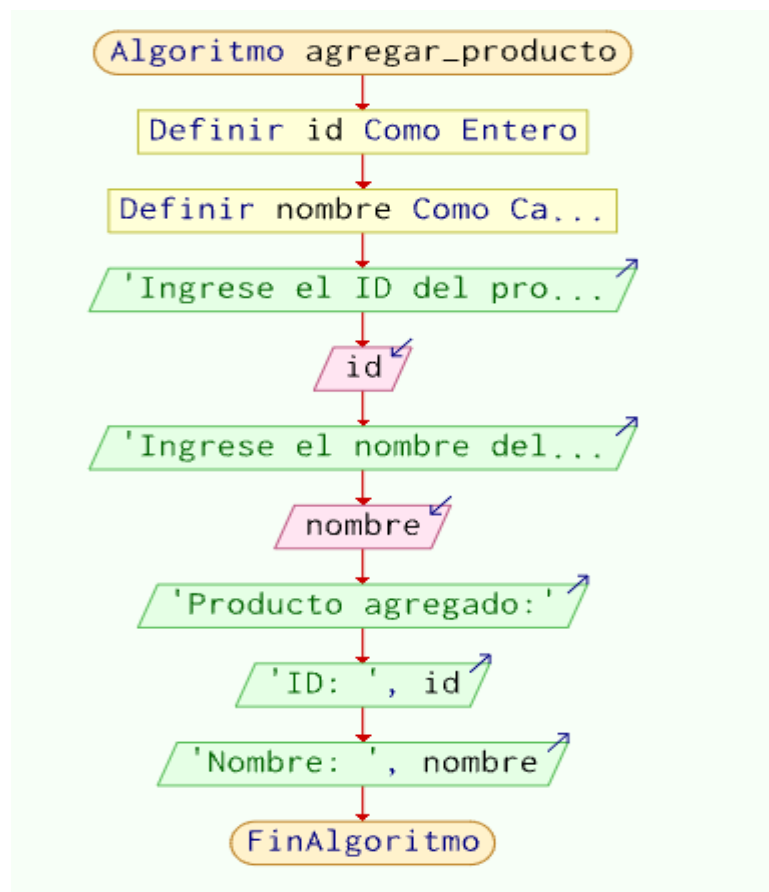
    // Pedir al usuario que ingrese el nombre y el ID del producto
    printf("Ingrese el ID del producto: ");
    scanf("%d", &producto.id);
    getchar(); // Para consumir el salto de línea después de ingresar el ID

    printf("Ingrese el nombre del producto: ");
    fgets(producto.nombre, sizeof(producto.nombre), stdin);

    // Eliminar el salto de línea al final del nombre, si existe
    producto.nombre[strcspn(producto.nombre, "\n")] = '\0';

    // Mostrar los datos ingresados
    printf("\nProducto agregado:\n");
    printf("ID: %d\n", producto.id);
    printf("Nombre: %s\n", producto.nombre);
    return 0;
}
```

### 2. DIAGRAMA DE FLUJO (DF)



## Prueba caja blanca de Modificar un producto

### 1. CÓDIGO FUENTE

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

// Estructura para representar un producto
struct Producto {
    int id;
    char nombre[50];
    int cantidad;
};

int main() {
    struct Producto producto;

    // Pedir al usuario que ingrese el ID, nombre y cantidad del producto
    printf("Ingrese el ID del producto: ");
    scanf("%d", &producto.id);
    getchar(); // Para consumir el salto de línea después de ingresar el ID

    printf("Ingrese el nombre del producto: ");
    fgets(producto.nombre, sizeof(producto.nombre), stdin);
    producto.nombre[strcspn(producto.nombre, "\n")] = '\0'; // Eliminar el salto de línea

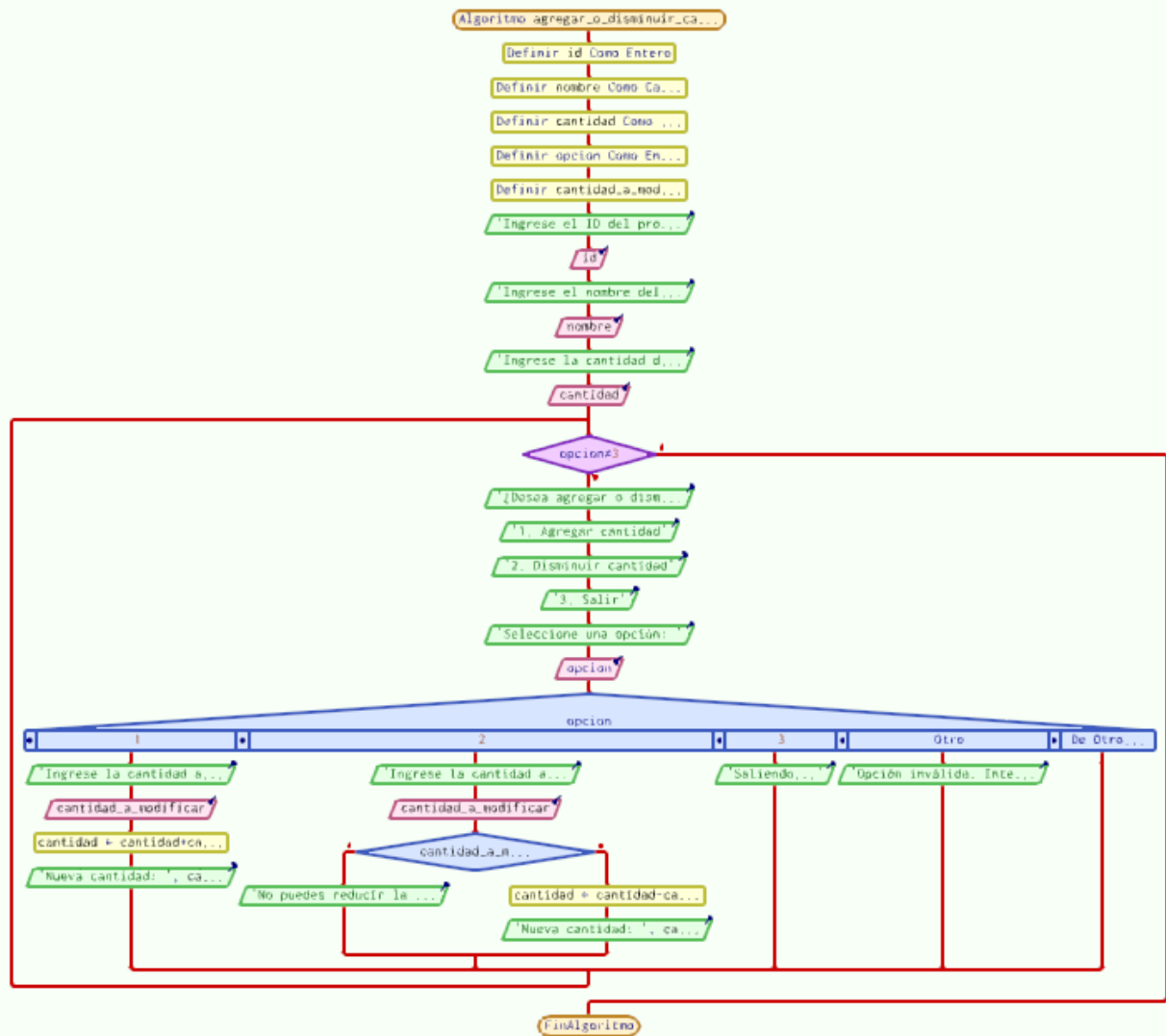
    printf("Ingrese la cantidad del producto: ");
    scanf("%d", &producto.cantidad);

    int opcion, cantidad;
    do {
        printf("\n¿Desea agregar o disminuir la cantidad?\n");
        printf("1. Agregar cantidad\n");
        printf("2. Disminuir cantidad\n");

        printf("3. Salir\n");
        printf("Seleccione una opción: ");
        scanf("%d", &opcion);

        switch(opcion) {
            case 1:
                printf("Ingrese la cantidad a agregar: ");
                scanf("%d", &cantidad);
                producto.cantidad += cantidad;
                printf("Nueva cantidad: %d\n", producto.cantidad);
                break;
            case 2:
                printf("Ingrese la cantidad a disminuir: ");
                scanf("%d", &cantidad);
                if (cantidad <= producto.cantidad) {
                    producto.cantidad -= cantidad;
                    printf("Nueva cantidad: %d\n", producto.cantidad);
                } else {
                    printf("No puedes reducir la cantidad a un valor negativo.\n");
                }
                break;
            case 3:
                printf("Saliendo...\n");
                break;
            default:
                printf("Opción inválida. Intenta nuevamente.\n");
        }
    } while(opcion != 3);
    return 0;
}
```

## 2. DIAGRAMA DE FLUJO (DF)



## Prueba caja blanca de Modificar un producto

### 1. CÓDIGO FUENTE

```
1  #include <stdio.h>
2  #include <string.h>
3  // Estructura para representar un producto
4  struct Producto {
5      int id;
6      char nombre[50];
7  };
8  int main() {
9      struct Producto producto;
10     // Pedir al usuario que ingrese el ID y nombre del producto
11     printf("Ingrese el ID del producto: ");
12     scanf("%d", &producto.id);
13     getchar(); // Para consumir el salto de línea después de ingresar el ID
14
15     printf("Ingrese el nombre del producto: ");
16     fgets(producto.nombre, sizeof(producto.nombre), stdin);
17     producto.nombre[strcspn(producto.nombre, "\n")] = '\0'; // Eliminar el salto de línea
18
19     int opcion;
20     printf("\n¿Desea eliminar el producto?\n");
21     printf("1. Eliminar producto\n");
22     printf("2. No eliminar\n");
23     printf("Seleccione una opción: ");
24     scanf("%d", &opcion);
25
26     if (opcion == 1) {
27         // Eliminamos el producto (borramos los datos)
28         producto.id = 0;
29         strcpy(producto.nombre, "");
30
31         printf("\nProducto eliminado.\n");
32     } else {
33         printf("\nEl producto no fue eliminado.\n");
34     }
35
36     // Mostrar los datos del producto (deberían ser 0 o vacíos si se eliminó)
37     printf("\nID: %d\n", producto.id);
38     printf("Nombre: %s\n", producto.nombre);
39
40     return 0;
41 }
```

### 2. DIAGRAMA DE FLUJO (DF)

