

# Prueba de Caja Blanca

---

*“Gestión de inventarios de una tienda de ropa”*

**Integrantes:**

Pacha Shakira

Sivinta Jahir

Ulcuango Josselyn

Fecha	Versión	Descripción	Autores
16/Enero/2025	1	Versión inicial	Shakira Pacha Jahir Sivinta Josselyn Ulcuango
22/Enero/2025	2	Segunda versión	Shakira Pacha Jahir Sivinta Josselyn Ulcuango
28/Enero/2025	3	Tercera versión	Shakira Pacha Jahir Sivinta Josselyn Ulcuango
3/Febrero/2025	4	Cuarta versión	Shakira Pacha Jahir Sivinta Josselyn Ulcuango
6/Febrero/2025	5	Quinta versión	Shakira Pacha Jahir Sivinta Josselyn Ulcuango
13/Febrero	6	Sexta versión	Shakira Pacha Jahir Sivinta Josselyn Ulcuango

## Prueba caja blanca

### Requisito numero 1: Inicio de sesión

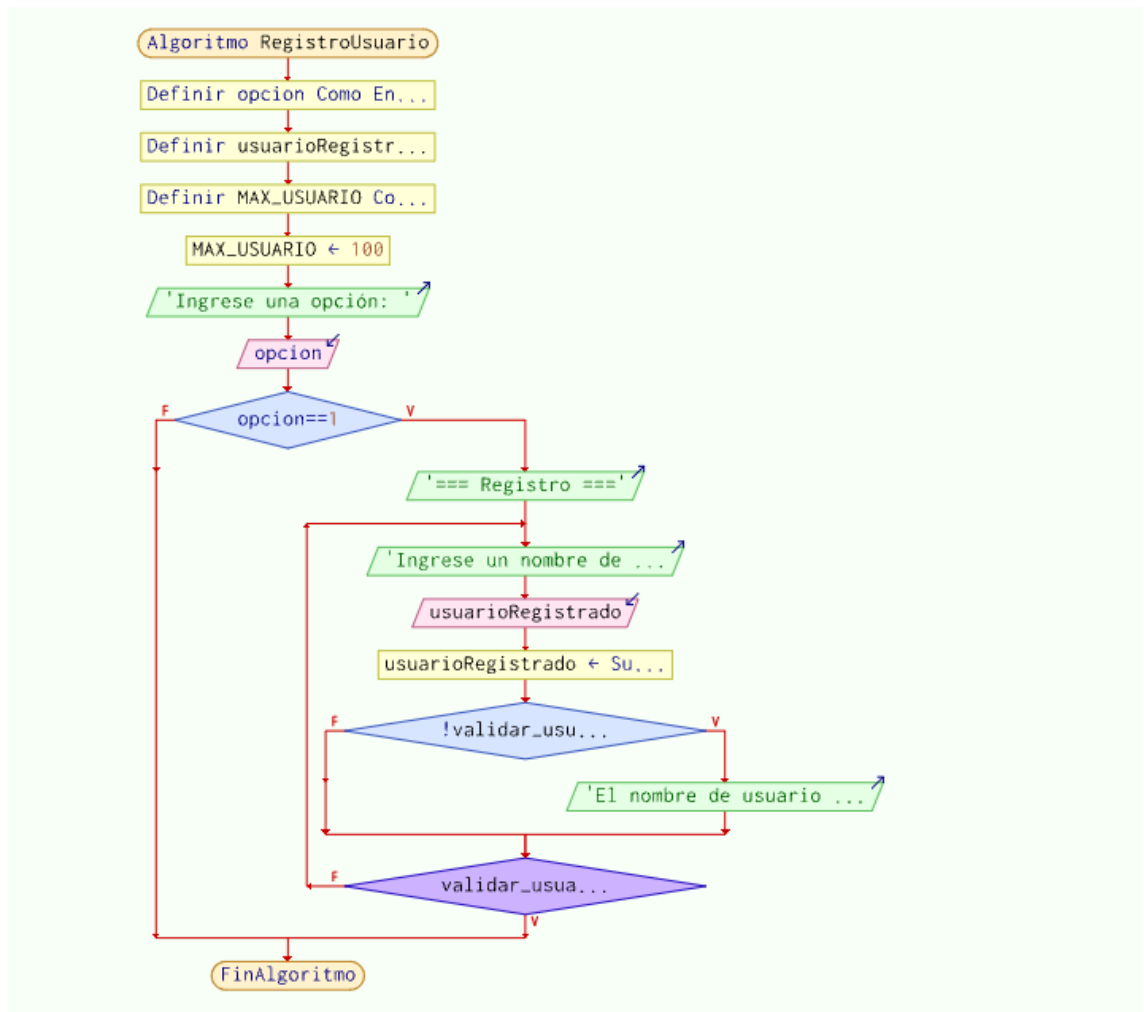
#### 1. Describa el requisito funcional

```

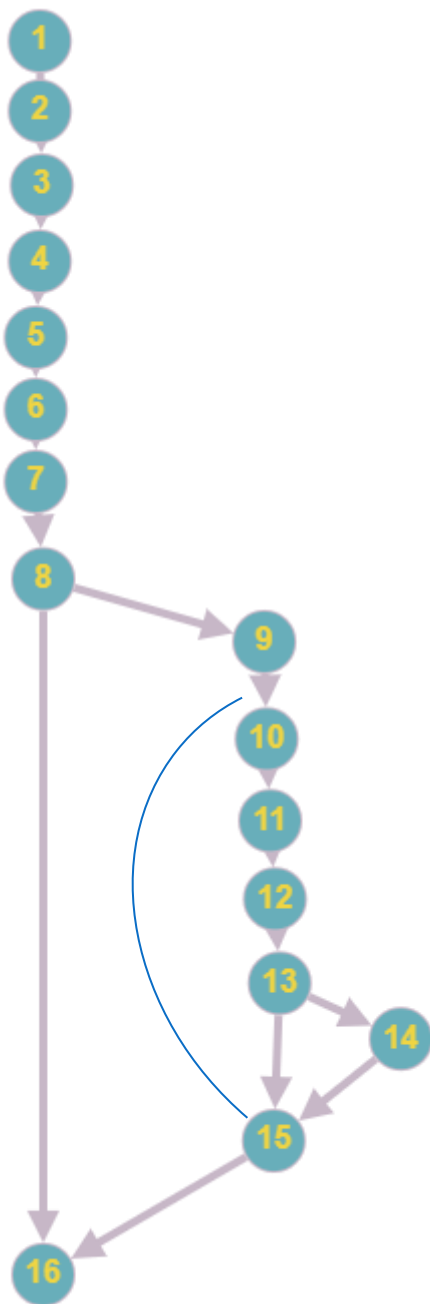
43     printf("==== Menú ==== \n");
44     printf("1. Registrarse \n");
45     printf("2. Iniciar sesión \n");
46     printf("3. Salir \n");
47     printf("Seleccione una opción: ");
48     scanf("%d", &opcion);
49     getchar(); // Consumir el salto de línea restante
50
51     if (opcion == 1) { // Registro de usuario
52         printf("==== Registro === \n");
53
54         do {
55             printf("Ingrese un nombre de usuario: ");
56             fgets(usuarioRegistrado, MAX_USUARIO, stdin);
57             usuarioRegistrado[strcspn(usuarioRegistrado, "\n")] = '\0'; // Eliminar el salto de línea
58
59             if (!validar_usuario(usuarioRegistrado)) {
60                 printf("El nombre de usuario debe contener al menos un carácter alfabético o un número. \n");
61             }
62         } while (!validar_usuario(usuarioRegistrado));
63
64         do {
65             printf("Ingrese una contraseña: ");
66             fgets(contrasenaRegistrada, MAX_CONTRASEÑA, stdin);
67             contrasenaRegistrada[strcspn(contrasenaRegistrada, "\n")] = '\0'; // Eliminar el salto de línea

```

#### 2. DIAGRAMA DE FLUJO (DF)



### 3. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

##### **RUTAS**

**R1:** 1,2,3,4,5,6,7,8,16

**R2:** 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

**R3:** 1,2,3,4,5,6,7,8,9,10,11,12,13,15,10,11,12,13,14,15,16

**R4:** 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,10,11,12,13,14,15,16

#### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichos(decisiones)} + 1$   
 $V(G) = 3 + 1 = 4$

- $V(G) = A - N + 2$   
 $V(G) = 18 - 16 + 2 = 4$

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

Requisito numero 2: Añadir nuevas prendas

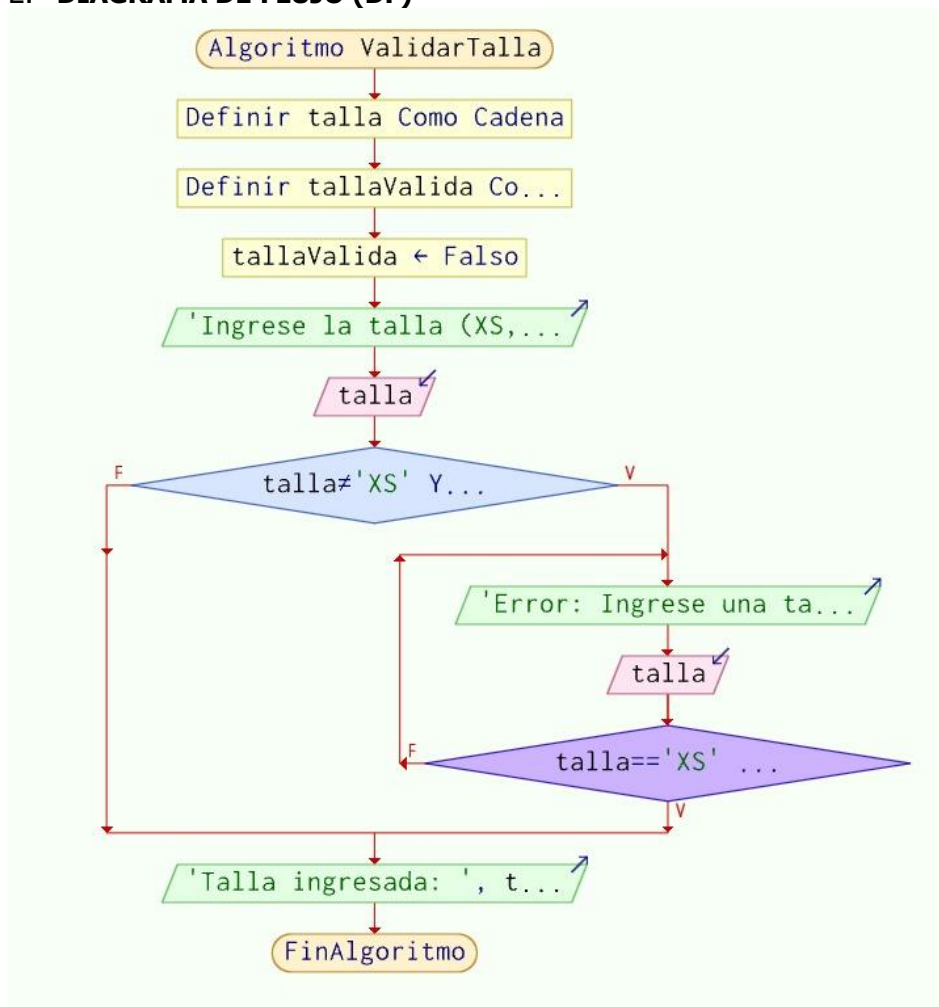
1. describa el requisito funcional

```

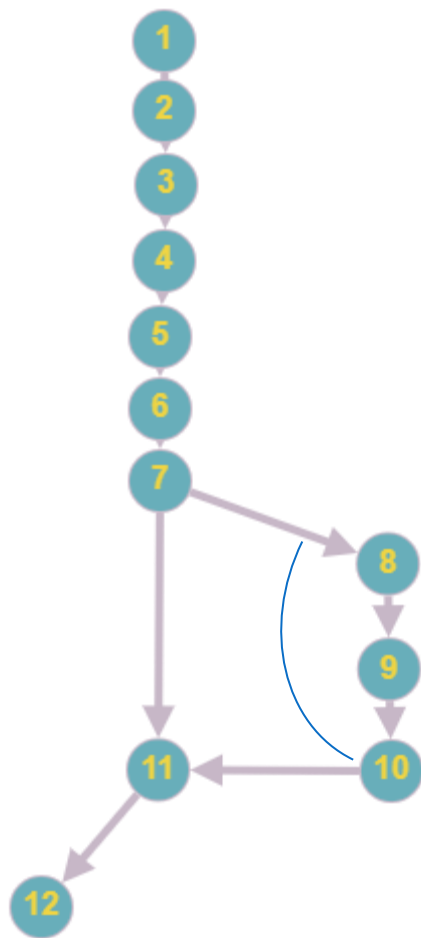
22 // Ingresar talla (XS, S, M, L)
23 printf("Ingrese la talla (XS, S, M, L): ");
24 scanf("%s", p->talla);
25 if (strcmp(p->talla, "XS") != 0 && strcmp(p->talla, "S") != 0 && strcmp(p->talla, "M") != 0 && strcmp(p->talla, "L") != 0) {
26     do {
27         printf("Error: Ingrese una talla valida (XS, S, M, L): ");
28         scanf("%s", p->talla);
29     } while (strcmp(p->talla, "XS") != 0 && strcmp(p->talla, "S") != 0 && strcmp(p->talla, "M") != 0 && strcmp(p->talla, "L") != 0);
30 }
31
32 // Ingresar color
33 printf("Ingrese el color de la prenda: ");
34 scanf("%s", p->color);
35

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

##### **RUTAS**

**R1:** 1,2,3,4,5,6,7,11,12

**R2:** 1,2,3,4,5,6,7,8,9,10,11,12

**R3:** 1,2,3,4,5,6,7,8,9,10,8,9,10,11,12

##### **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 13 - 12 + 2 = 3$

DONDE:

**P:** Número de nodos predichado

**A:** Número de aristas

**N:** Número de nodos

Requisito numero 3: Filtro de búsqueda

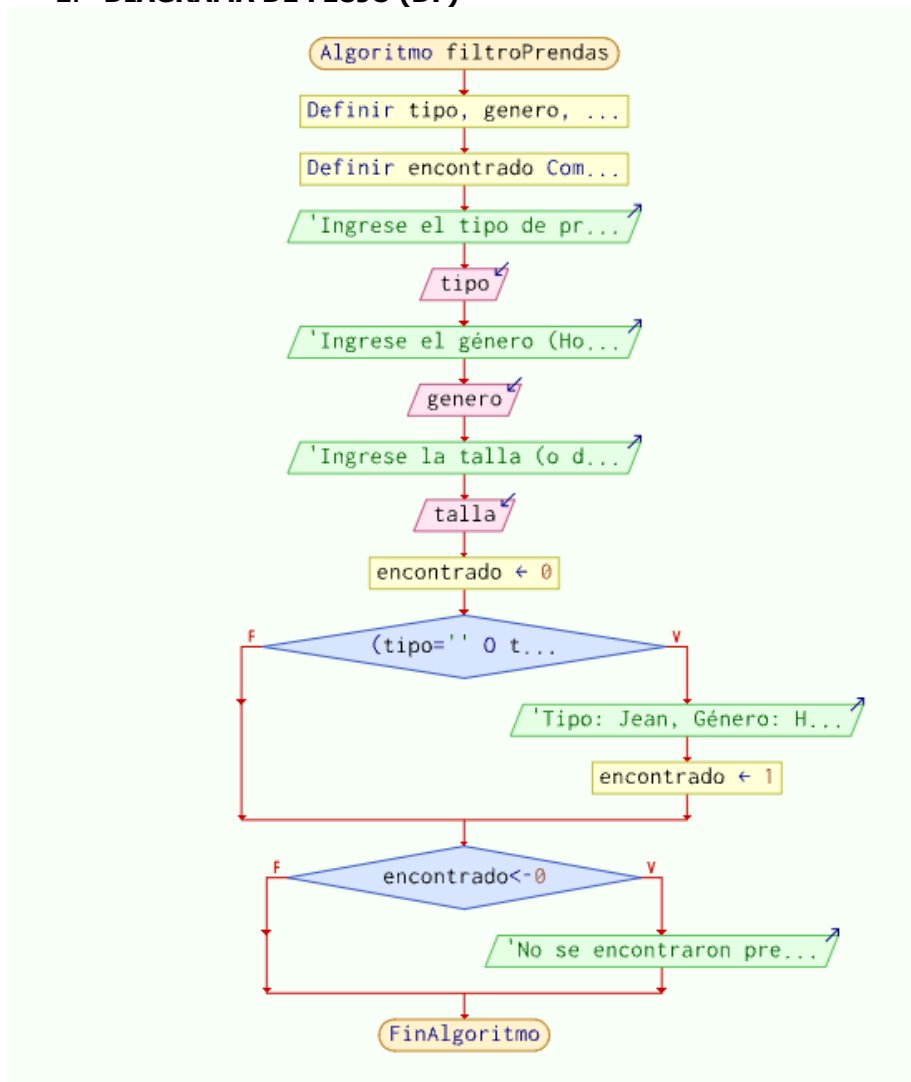
## 1. Describa el requisito funcional

```
14 // Función para filtrar y mostrar las prendas según los criterios
15 void filtrarPrendas(Prenda prendas[], int total, const char* tipo, const char* genero, const char* talla) {
16     int encontrado = 0;
17     printf("\nResultados de la búsqueda:\n");
18     printf("-----\n");
19     for (int i = 0; i < total; i++) {
20         if ((strcmp(tipo, "") == 0 || strcmp(prendas[i].tipo, tipo) == 0) &&
21             (strcmp(genero, "") == 0 || strcmp(prendas[i].genero, genero) == 0) &&
22             (strcmp(talla, "") == 0 || strcmp(prendas[i].talla, talla) == 0)) {
23             printf("Tipo: %s, Género: %s, Talla: %s, Cantidad: %d\n",
24                 prendas[i].tipo, prendas[i].genero, prendas[i].talla, prendas[i].cantidad);
25             encontrado = 1;
26         }
27     }
28     if (!encontrado) {
29         printf("No se encontraron prendas que coincidan con los criterios.\n");
30     }
31 }
32
```

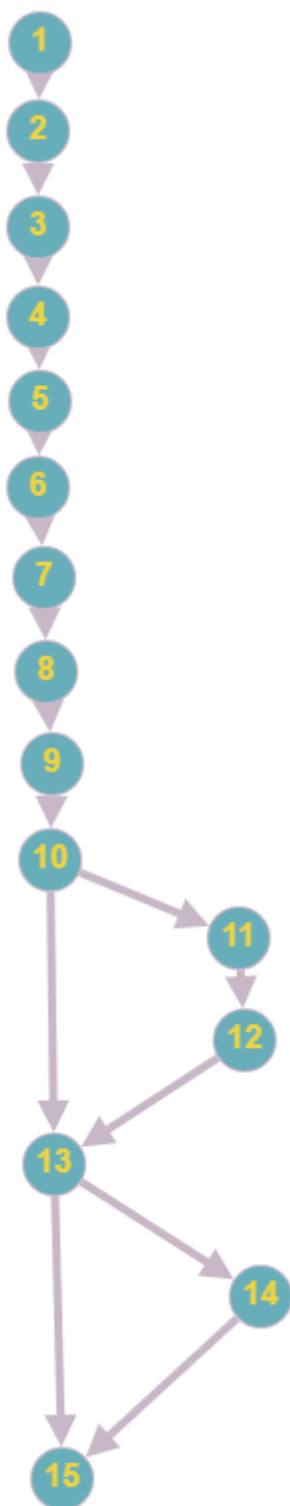
1

2

## 2. DIAGRAMA DE FLUJO (DF)



## 3. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

##### **RUTAS**

**R1:** 1,2,3,4,5,6,7,8,9,13,15

**R2:** 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

**R3:** 1,2,3,4,5,6,7,8,9,10,11,12,13,15

#### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:



- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 16 - 15 + 2 = 3$

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

Requisito numero 4: Filtro de búsqueda

## 6. Describa el requisito funcional

```

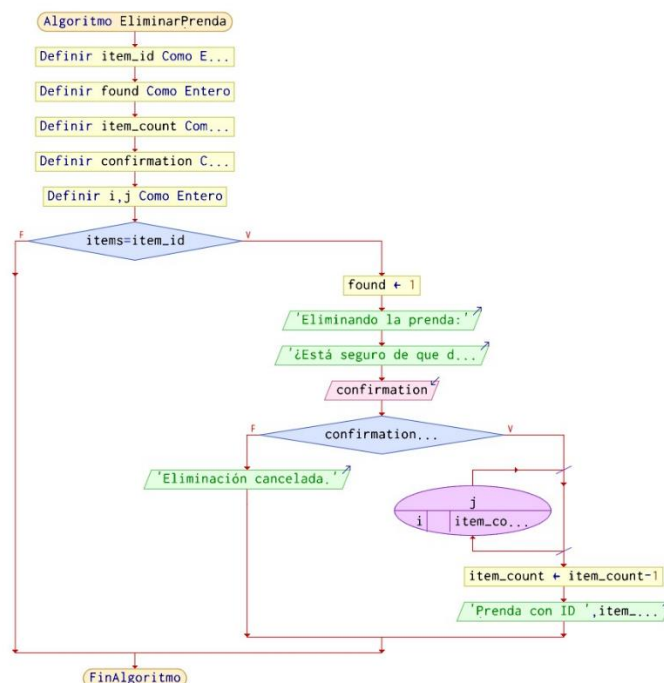
36 // Función para eliminar una prenda del inventario por ID
37 void remove_item(Inventory *inventory) {
38     char item_id;
39     char confirmation[10]; // Para almacenar respuestas de "si" o "no"
40     int found = 0;
41
42     // Listar las prendas disponibles
43     list_items(inventory);
44     printf("Para eliminar una prenda, debe ingresar un ID en letra mayúscula.\n");
45
46     printf("Ingrese el ID de la prenda a eliminar (letra del alfabeto en mayúscula): ");
47     scanf(" %c", &item_id); // Se utiliza un espacio antes de %c para ignorar espacios en blanco
48
49     // Verificar si el inventario está vacío
50     if (inventory->item_count == 0) {
51         printf("Error: El inventario está vacío.\n");
52         return;
53     }
54
55     // Buscar la prenda con el ID especificado
56     for (int i = 0; i < inventory->item_count; i++) {
57         if (inventory->items[i].item_id == item_id) {
58             found = 1;
59
60             // Mostrar detalles de la prenda que se está eliminando

```

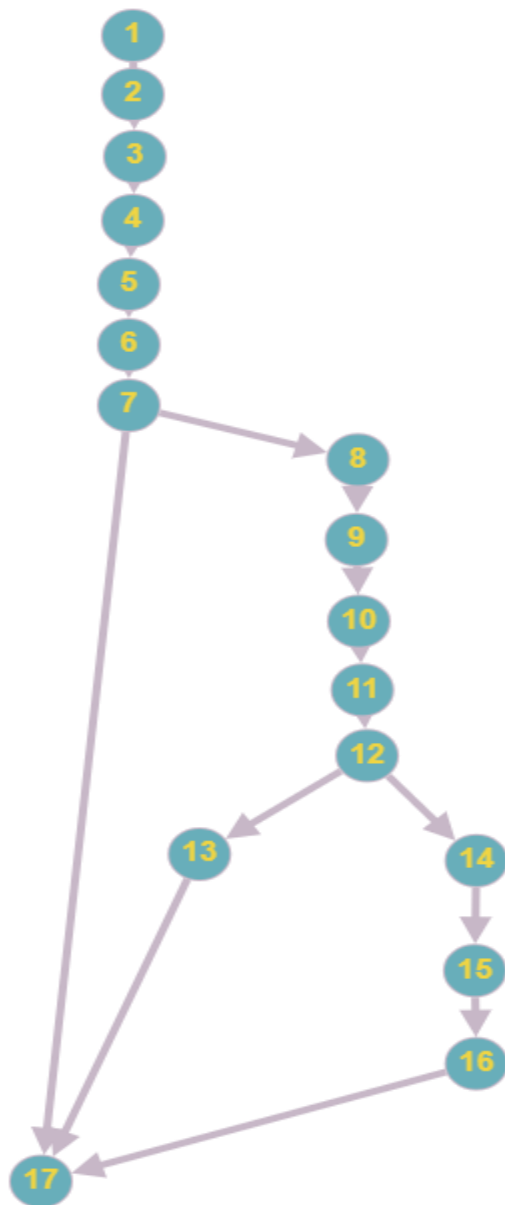
1

2

## 7. DIAGRAMA DE FLUJO (DF)



## 8. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

### RUTAS

**R1:** 1,2,3,4,5,6,7,17

**R2:** 1,2,3,5,6,7,8,9,10,11,12,13,17

**R3:** 1,2,3,5,6,7,8,9,10,11,12,14,15,16,17

## 10. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 18 - 17 + 2 = 3$

DONDE:

**P:** Número de nodos prediado  
**A:** Número de aristas  
**N:** Número de nodos

## Requisito numero 5: Editar información

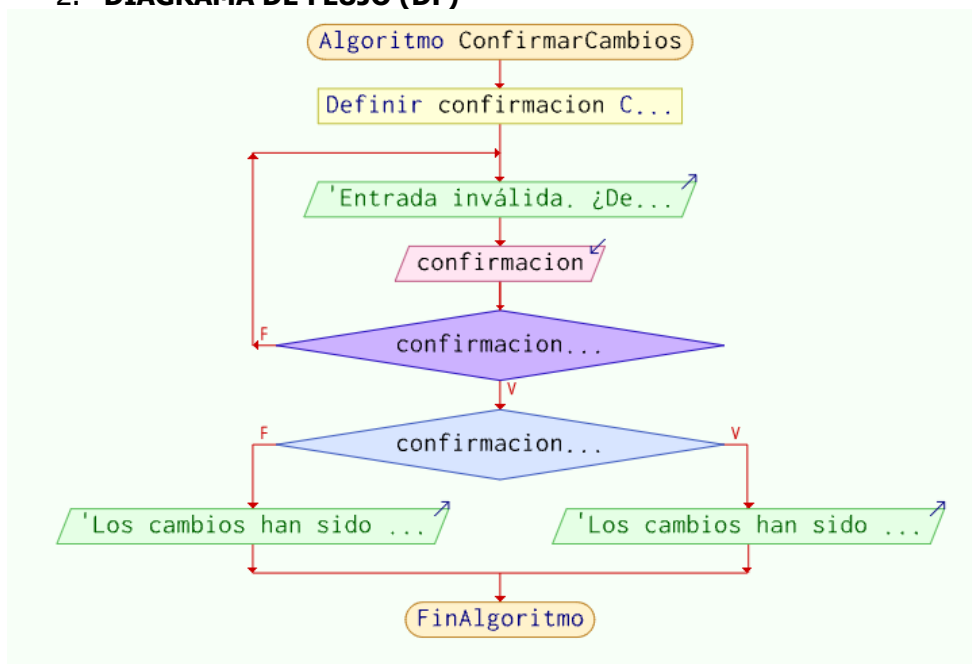
### 1. Describa el requisito funcional

```

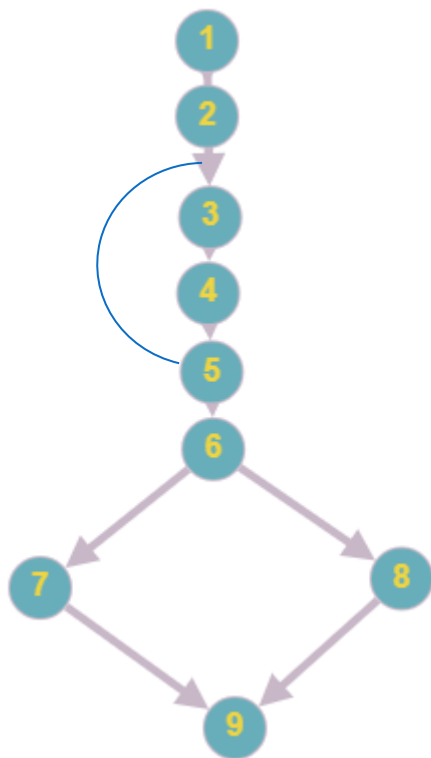
66     char confirmacion;
67     printf("\n¿Desea guardar los cambios realizados? (S/N): ");
68     scanf(" %c", &confirmacion);
69
70     while (confirmacion != 'S' && confirmacion != 's' && confirmacion != 'N' && confirmacion != 'n') {
71         printf("Entrada inválida. ¿Desea guardar los cambios realizados? (S/N): ");
72         scanf(" %c", &confirmacion);
73     }
74
75     if (confirmacion == 'N' || confirmacion == 'n') {
76         printf("Los cambios han sido descartados.\n");
77     } else {
78         printf("Los cambios han sido guardados exitosamente.\n");
79     }
80 }
81
82 int main() {
83     struct Prenda miPrenda = {"Mujer", "M", "Rojo", "Chompa", 5};
84     int opcion;
85

```

### 2. DIAGRAMA DE FLUJO (DF)



### 3. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

##### **RUTAS**

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

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

**R3:** 1,2,3,4,5,3,4,5

#### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 9 - 8 + 2 = 3$

DONDE:

**P:** Número de nodos predichado

**A:** Número de aristas

**N:** Número de nodos