

Prueba de Caja Blanca

“Sistema automatizado de control de Inventario”

Versión 4.0

Integrantes:

Caizapanta Muela Tammy Amarilis

Guaiguacundo Aguirre Valeria Naomi

Pincha Llanos Estefany Anahi

Robalino Zaldumbide Alejandro Benjamín

Fecha

2025-02-11

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 2
		Fecha: 15/02/2025

ÍNDICE

ÍNDICE	2
Historial de versionamiento	4
1. REQ 001 - Agregar productos	5
1.1. CÓDIGO FUENTE	5
1.2. DIAGRAMA DE FLUJO	6
1.3. GRAFO	6
1.4. RUTAS	7
1.5. COMPLEJIDAD CICLOMÁTICA	7
2. REQ 002 - Buscar productos	8
2.1. CÓDIGO FUENTE	8
2.2. DIAGRAMA DE FLUJO	8
2.3. GRAFO	9
2.4. RUTAS	9
2.5. COMPLEJIDAD CICLOMÁTICA	9
3. REQ 003 - Entrada de productos	10
3.1. CÓDIGO FUENTE	10
3.2. DIAGRAMA DE FLUJO	11
3.3. GRAFO	12
3.4. RUTAS	12
3.5. COMPLEJIDAD CICLOMÁTICA	12
4. REQ 004 - Salida de Productos	13
4.1. CÓDIGO FUENTE	13
4.2. DIAGRAMA DE FLUJO	14
4.3. GRAFO	15
4.4. RUTAS	15
4.5. COMPLEJIDAD CICLOMÁTICA	15

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 3
		Fecha: 15/02/2025

5. REQ 005 - Eliminar Productos	16
5.1. CÓDIGO FUENTE	16
5.2. DIAGRAMA DE FLUJO	17
5.3. GRAFO	17
5.4. RUTAS	18
5.5. COMPLEJIDAD CICLOMÁTICA	18
6. REQ 006 - Generar reporte de la disponibilidad de productos	19
6.1. CÓDIGO FUENTE	19
6.2. DIAGRAMA DE FLUJO	19
6.3. GRAFO	20
6.4. RUTAS	20
6.5. COMPLEJIDAD CICLOMÁTICA	20

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 4
		Fecha: 15/02/2025

Historial de versionamiento

Fecha	Versión	Descripción	Autores
23 / 01 /2025	1	Versión inicial	Caizapanta Tammy Guaiguacundo Valeria Pincha Estefany Robalino Alejandro
28 / 01 /2025	2	Modificación de diagramas de flujo	Caizapanta Tammy Guaiguacundo Valeria Pincha Estefany Robalino Alejandro
11 / 02 /2025	3	Modificación de grafos	Caizapanta Tammy Guaiguacundo Valeria Pincha Estefany Robalino Alejandro
15 / 02 /2025	4	Asignar numeración al código fuente Diseñar DF Y GR conforme al código fuente. Corrección de rutas. Se verificó correspondencia entre diagramas de flujo, grafos y rutas.	Caizapanta Tammy Guaiguacundo Valeria Pincha Estefany Robalino Alejandro

Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 5
		Fecha: 15/02/2025

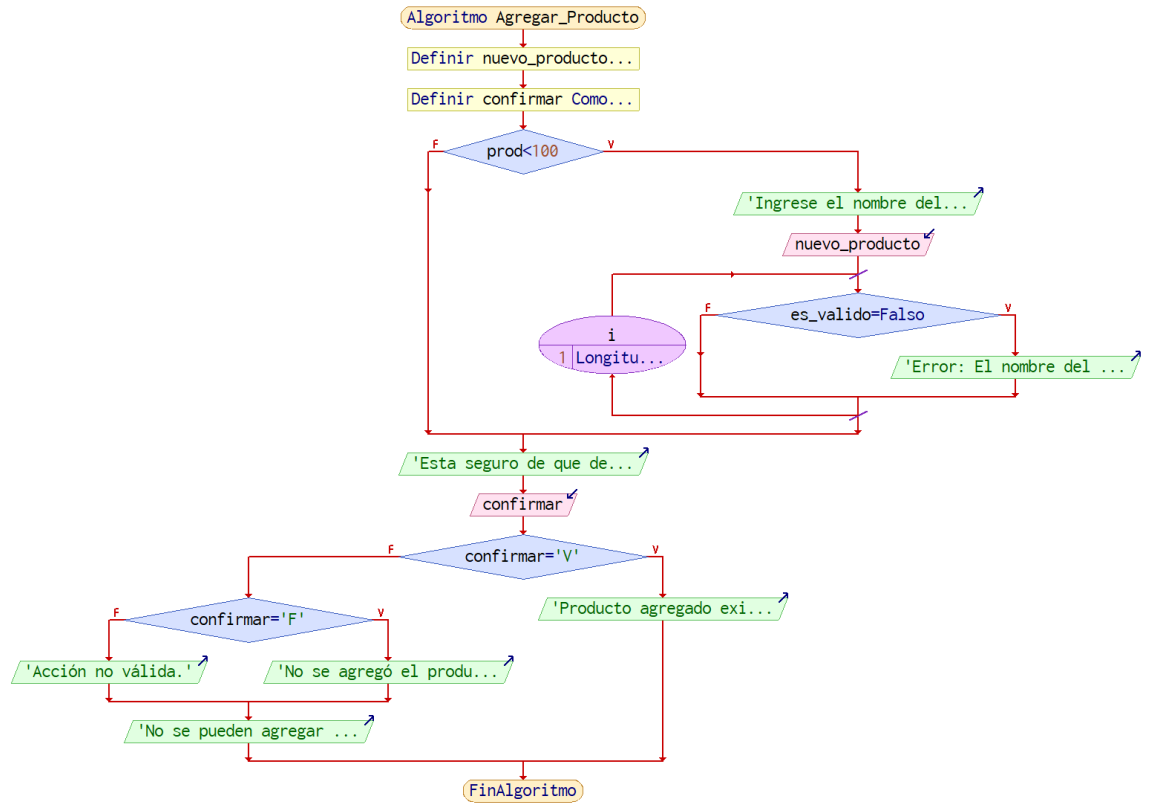
1. REQ 001 - Agregar productos

1.1. CÓDIGO FUENTE

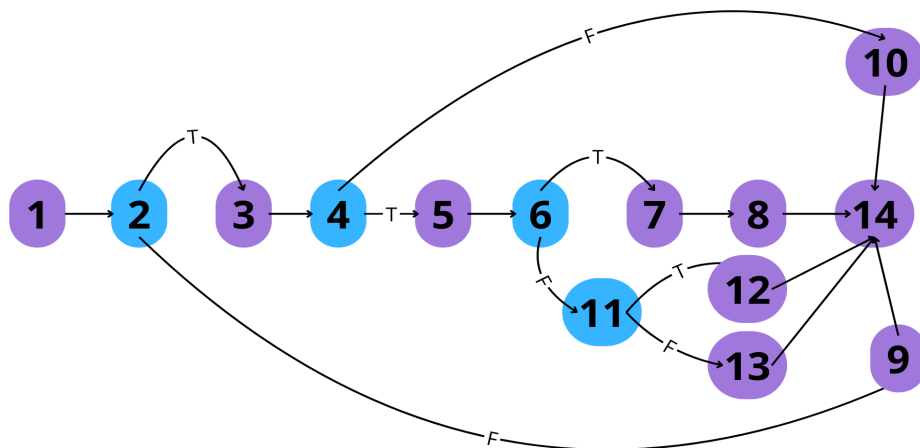
```
38 // Funcion para agregar productos
39 void agregar_producto() {
40     1 if (prod < 100) {
41         char nuevo_producto[50];
42         char confirmar;
43
44         printf("Ingrese el nombre del nuevo producto: ");
45         getchar();
46         fgets(nuevo_producto, 50, stdin);
47
48         nuevo_producto[strcspn(nuevo_producto, "\n")] = '\0';
49
50
51         for (int i = 0; nuevo_producto[i] != '\0'; i++) {
52             2 if (isdigit(nuevo_producto[i])) {
53                 printf("Error: El nombre del producto no puede contener numeros.\n");
54                 return;
55             }
56         }
57
58         printf("Esta seguro de que desea agregar el producto '%s' (Y/N): ", nuevo_producto);
59         scanf(" %c", &confirmar);
60         confirmar = toupper(confirmar);
61
62         3 if (confirmar == 'Y') {
63             sprintf(producto[prod].product, "%03d.%s", prod + 1, nuevo_producto);
64             prod++;
65             printf("Producto agregado exitosamente.\n");
66             4 } else if (confirmar == 'N') {
67                 printf("No se agrego el producto.\n");
68             } else {
69                 printf("Accion no valida. \n");
70             }
71         } else {
72             printf("No se pueden agregar mas productos. Capacidad maxima alcanzada.\n");
73         }
74     }
```

Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 6
		Fecha: 15/02/2025

1.2. DIAGRAMA DE FLUJO



1.3. GRAFO



<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 7
		Fecha: 15/02/2025

1.4. RUTAS

R1: 1→2→3→4→10→14

R2: 1→2→3→4→5→6→7→8→14

R3: 1→2→3→4→5→6→11→12→14

R4: 1→2→3→4→9→10→11→13→14

R5: 1→2→9→14

1.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

$$V(G) = 4 + 1 = 5$$

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

$$V(G) = 17 - 14 + 2$$

$$V(G) = 5$$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

<p>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</p>	<p>Caja Blanca</p>	<p>Versión: 4.0</p>
		<p>Página: 8</p>
		<p>Fecha: 15/02/2025</p>

2. REQ 002 - Buscar productos

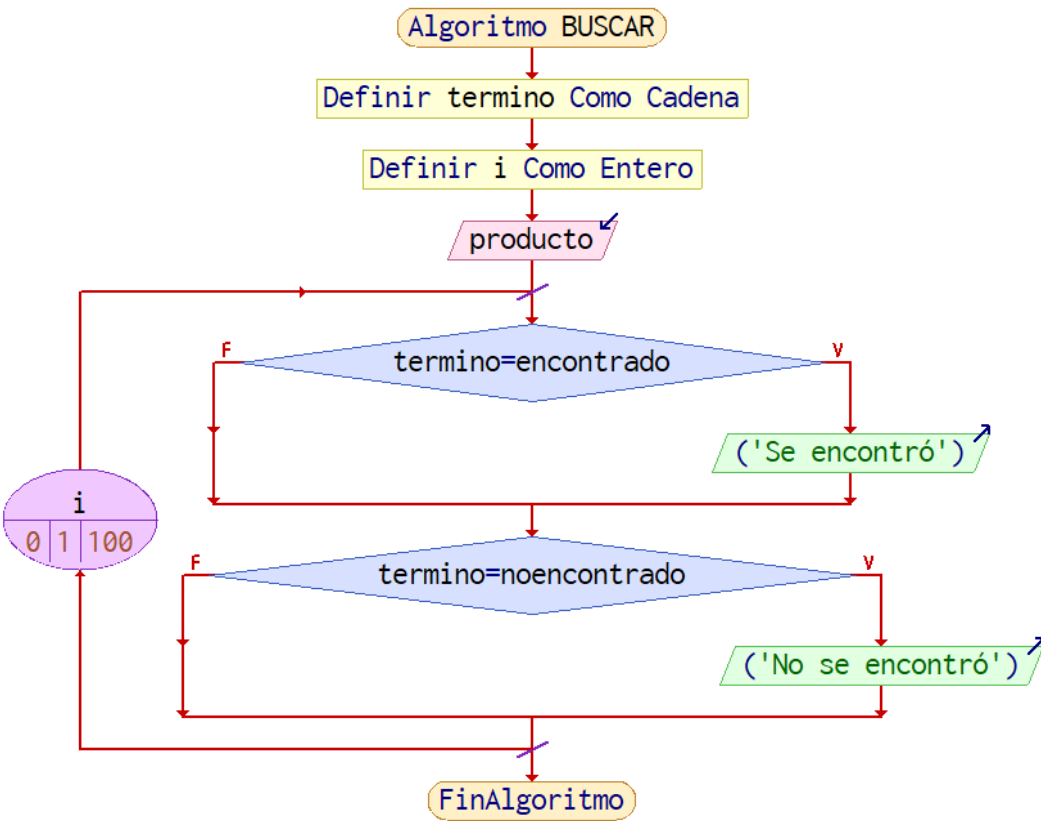
2.1. CÓDIGO FUENTE

```

77 // Funcion para buscar productos
78 void buscar_producto() {
79     char termino[50];
80     int encontrado = 0;
81
82     printf("Ingrese el nombre o fragmento del producto a buscar: ");
83     getchar();
84     fgets(termino, 50, stdin);
85
86     termino[strcspn(termino, "\n")] = '\0';
87
88     printf("\nResultados de búsqueda:\n");
89     for (int i = 0; i < prod; i++) {
90         1 if (strstr(producto[i].product, termino) != NULL) {
91             printf(" %d: [Producto: %s] (Cantidad: %d)\n", i + 1, producto[i].product, producto[i].cantidad);
92             encontrado = 1;
93         }
94     }
95
96     2 if (!encontrado) {
97         printf("No se encontraron productos que coincidan con '%s'.\n", termino);
98     }
99 }

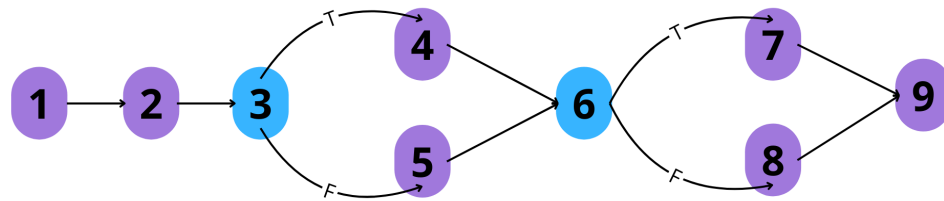
```

2.2. DIAGRAMA DE FLUJO



<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 9
		Fecha: 15/02/2025

2.3. GRAFO



2.4. RUTAS

R1: 1→2→3→4→6→8→9

R2: 1→2→3→5→6→7→9

R3: 1→2→3→5→6→8→9

2.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$
 $V(G) = 10 - 9 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 10
		Fecha: 15/02/2025

3. REQ 003 - Entrada de productos

3.1. CÓDIGO FUENTE

```
// Funcion para entrada de productos
void entrada_producto() {
    char termino[50];
    int cantidad;
    int encontrado = 0;
    char confirmar;

    printf("Ingrese el codigo o nombre del producto para aumentar su cantidad: ");
    getchar();
    fgets(termino, 50, stdin);
    termino[strcspn(termino, "\n")] = '\0';

    1 if (strlen(termino) == 0) {
        printf("Error: No ingreso ningun valor. Intente nuevamente.\n");
        return;
    }

    for (int i = 0; i < prod; i++) {
        2 if (strstr(producto[i].product, termino) != NULL) {
            printf("\nProducto encontrado: %s (Cantidad actual: %d)\n", producto[i].product, producto[i].cantidad);
            printf("Ingrese la cantidad a agregar: ");
            3 if (scanf("%d", &cantidad) != 1 || cantidad <= 0) {
                printf("Error: Cantidad invalida. Solo se aceptan numeros positivos.\n");
                return;
            }

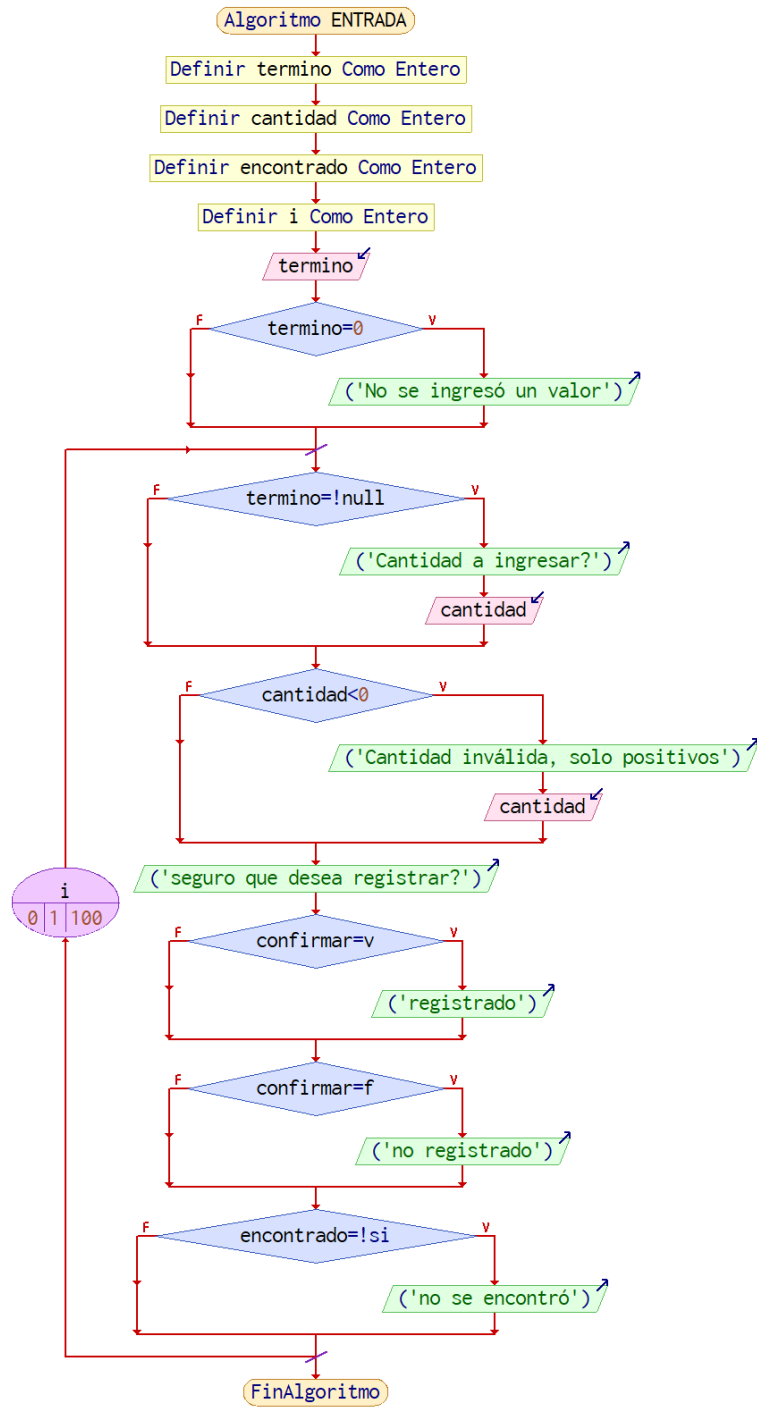
            printf("Esta seguro de que desea agregar %d a '%s' (V/F): ", cantidad, producto[i].product);
            scanf(" %c", &confirmar);
            confirmar = toupper(confirmar);

            4 if (confirmar == 'V') {
                producto[i].cantidad += cantidad;
                printf("\nCantidad actualizada. Nueva cantidad: %d\n", producto[i].cantidad);
            } 5 else if (confirmar == 'F') {
                printf("\n No se realizo ningun cambio.\n");
            } else {
                printf("\nOpcion no valida..\n");
            }

            encontrado = 1;
            break;
        } 6 } if (!encontrado) {
            printf("\nNo se encontro el producto con el codigo o nombre '%s'.\n", termino);
        }
    }
}
```

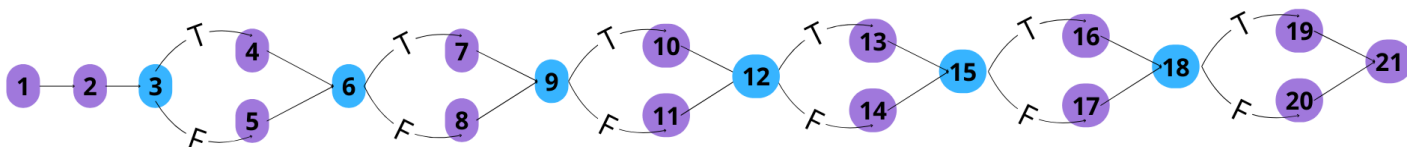
<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 11
		Fecha: 15/02/2025

3.2. DIAGRAMA DE FLUJO



Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 12
		Fecha: 15/02/2025

3.3. GRAFO



3.4. RUTAS

R1: 1 - 2 - 3 - 4 - 6 - 7 - 9 - 10 - 12 - 13 - 15 - 16 - 18 - 19 - 21

R2: 1 - 2 - 3 - 5 - 6 - 8 - 9 - 11 - 12 - 14 - 15 - 17 - 18 - 20 - 21

R3: 1 - 2 - 3 - 4 - 6 - 8 - 9 - 10 - 12 - 14 - 15 - 16 - 18 - 20 - 21

R4: 1 - 2 - 3 - 5 - 6 - 7 - 9 - 11 - 12 - 13 - 15 - 17 - 18 - 19 - 21

R5: 1 - 2 - 3 - 4 - 6 - 7 - 9 - 11 - 12 - 14 - 15 - 16 - 18 - 19 - 21

R6: 1 - 2 - 3 - 5 - 6 - 8 - 9 - 10 - 12 - 13 - 15 - 17 - 18 - 20 - 21

R7: 1 - 2 - 3 - 4 - 6 - 7 - 9 - 10 - 12 - 14 - 15 - 17 - 18 - 20 - 21

3.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

$$V(G) = 6 + 1 = 7$$

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

$$V(G) = 26 - 21 + 2$$

$$V(G) = 7$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 13
		Fecha: 15/02/2025

4. REQ 004 - Salida de Productos

4.1. CÓDIGO FUENTE

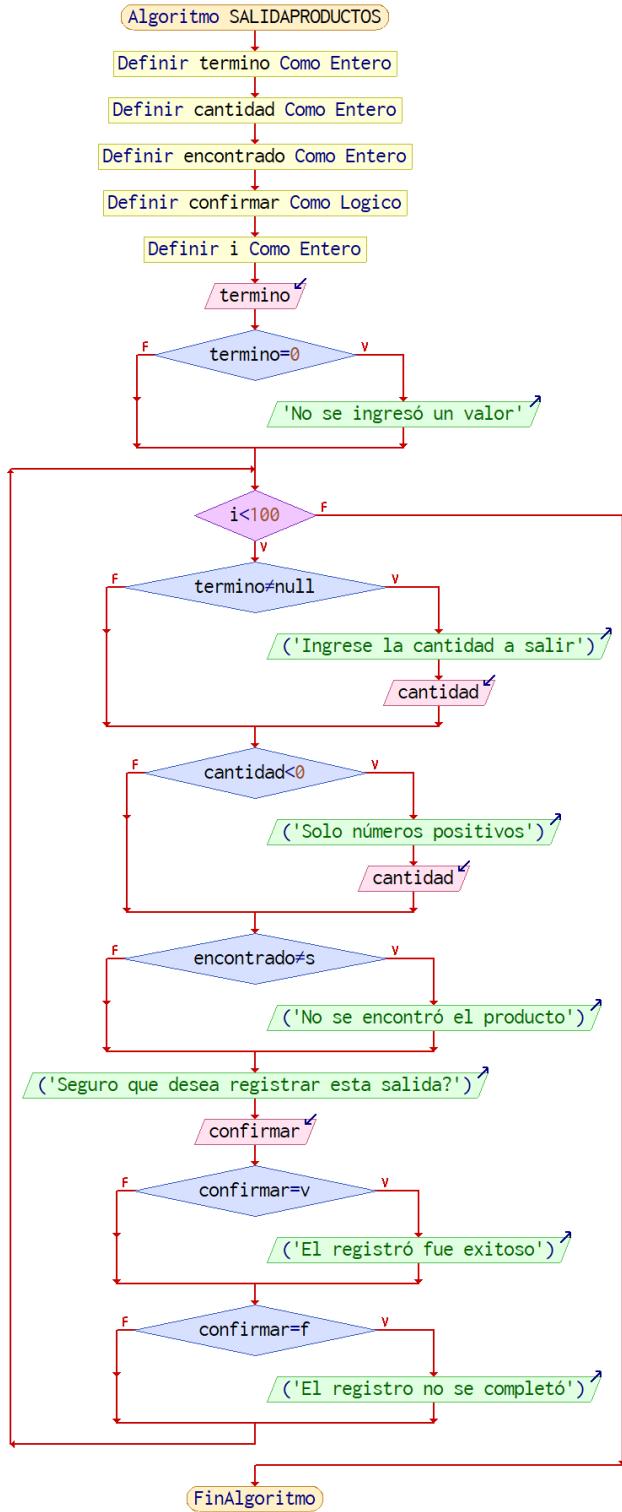
```

150 // Función para salida de productos
151 void salida_producto() {
152     char termino[50];
153     int cantidad;
154     int encontrado = 0;
155     char confirmar;
156
157     printf("Ingrese el código o nombre del producto para disminuir su cantidad: ");
158     getchar();
159     fgets(termino, 50, stdin);
160     termino[strcspn(termino, "\n")] = '\0';
161
162     for (int i = 0; i < prod; i++) {
163         1 if (strcmp(producto[i].product, termino) != NULL) {
164             printf("\nProducto encontrado: %s (Cantidad actual: %d)\n", producto[i].product, producto[i].cantidad);
165             printf("Ingrese la cantidad a retirar: ");
166             2 if (scanf("%d", &cantidad) != 1 || cantidad <= 0) {
167                 printf("Error: Cantidad invalida.\n");
168                 3 while (getchar() != '\n');
169                 return;
170             }
171
172             printf("Esta seguro de que desea retirar %d de '%s' (V/F): ", cantidad, producto[i].product);
173             scanf(" %c", &confirmar);
174             confirmar = toupper(confirmar);
175
176             4 if (confirmar == 'V') {
177                 5 if (cantidad <= producto[i].cantidad) {
178                     producto[i].cantidad -= cantidad;
179                     printf("\nCantidad actualizada. Nueva cantidad: %d\n", producto[i].cantidad);
180                 } else {
181                     printf("\nCantidad insuficiente en inventario.\n");
182                 }
183                 6 else if (confirmar == 'F') {
184                     printf("\nNo se realiza ningun cambio.\n");
185                 } else {
186                     printf("\nOpcion no valida.\n");
187                 }
188
189                 encontrado = 1;
190                 break;
191             }
192         }
193     }
194     7 if (!encontrado) {
195         printf("\nNo se encontro el producto con el código o nombre '%s'.\n", termino);
196     }
197 }

```

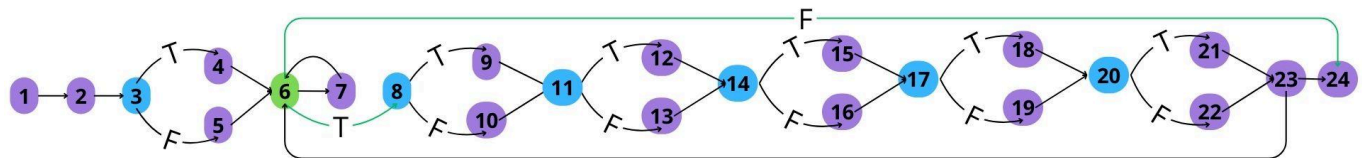
<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 14
		Fecha: 15/02/2025

4.2. DIAGRAMA DE FLUJO



<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 15
		Fecha: 15/02/2025

4.3. GRAFO



4.4. RUTAS

- R1: 1-2-3-4-6-7-6-24
- R2: 1-2-3-5-6-7-6-24
- R3: 1-2-3-4-6-7-6-8-9-11-12-14-15-17-18-20-21-23-24
- R4: 1-2-3-5-6-7-6-8-10-11-13-14-16-17-19-20-22-23-24
- R5: 1-2-3-4-6-7-6-8-9-11-12-14-16-17-19-20-21-23-24
- R6: 1-2-3-5-6-7-6-8-10-11-13-14-15-17-18-20-22-23-24
- R7: 1-2-3-4-6-7-6-8-9-11-12-14-15-17-19-20-22-23-24
- R8: 1-2-3-5-6-7-6-8-10-11-13-14-16-17-18-20-21-23-24

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 16
		Fecha: 15/02/2025

4.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 7 + 1 = 8$
- $V(G) = A - N + 2$
 $V(G) = 30 - 24 + 2$
 $V(G) = 8$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos

Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 17
		Fecha: 15/02/2025

5. REQ 005 - Eliminar Productos

5.1. CÓDIGO FUENTE

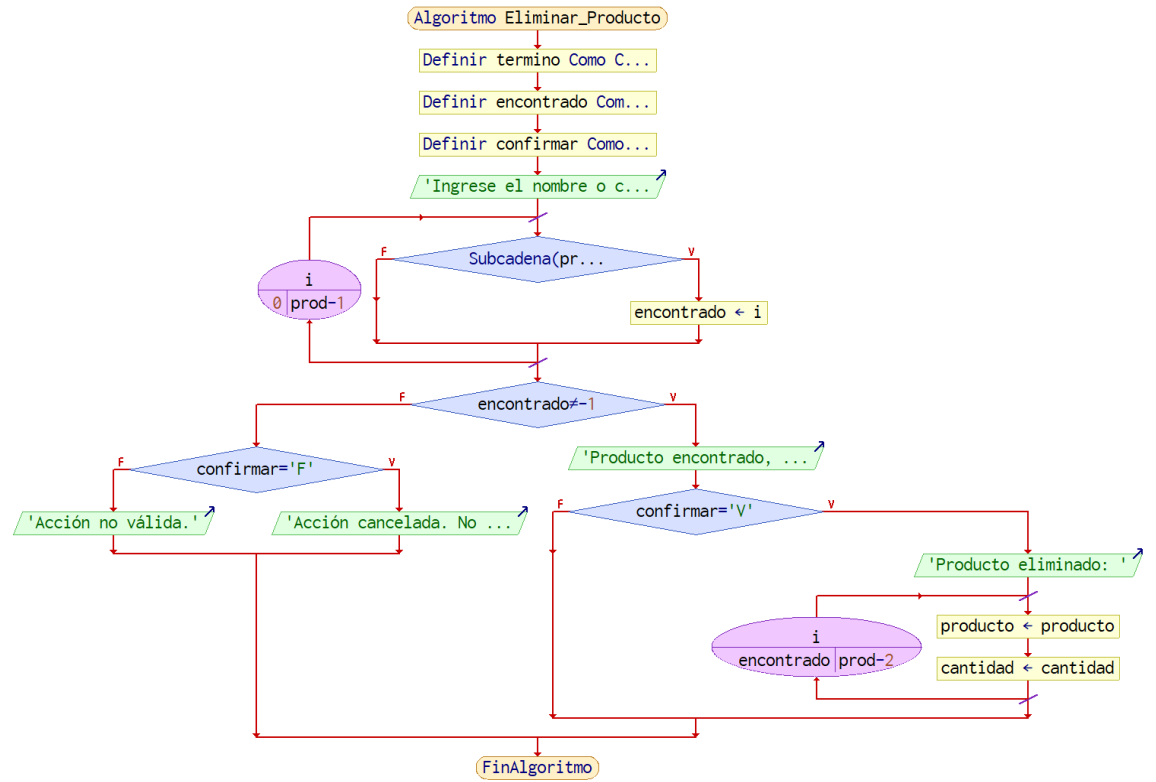
```

199 // Funcion para eliminar productos
200 void eliminar_producto() {
201     char termino[50];
202     int encontrado = -1;
203     char confirmar;
204
205     printf("Ingresa el nombre o código del producto a eliminar: ");
206     getchar();
207     fgets(termino, 50, stdin);
208     termino[strcspn(termino, "\n")] = 0;
209
210     for (int i = 0; i < prod; i++) {
211         if (strstr(producto[i].product, termino) != NULL) {
212             encontrado = i;
213             break;
214         }
215     }
216
217     if (encontrado != -1) {
218         // Preguntar si se desea eliminar el producto
219         printf("Producto encontrado: %s. Esta seguro de que desea eliminarlo (Y/N): ", producto[encontrado].product);
220         scanf(" %c", &confirmar);
221         confirmar = toupper(confirmar);
222
223         if (confirmar == 'Y') {
224             printf("Producto eliminado: %s\n", producto[encontrado].product);
225             for (int i = encontrado; i < prod - 1; i++) {
226                 strcpy(producto[i].product, producto[i + 1].product);
227                 producto[i].cantidad = producto[i + 1].cantidad;
228             }
229             prod--;
230         } else if (confirmar == 'N') {
231             printf("Acción cancelada. No se eliminó ningún producto.\n");
232         } else {
233             printf("Acción no válida.\n");
234         }
235     } else {
236         printf("No se encontró un producto que coincida con '%s'.\n", termino);
237     }
238 }

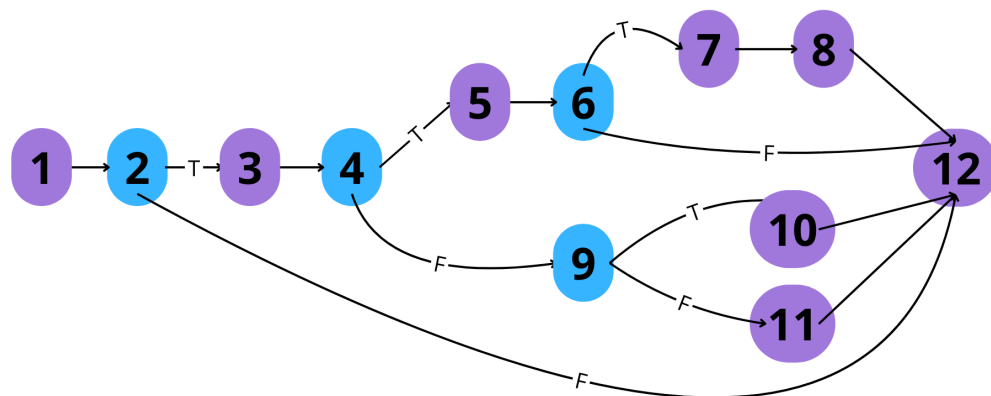
```

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 18
		Fecha: 15/02/2025

5.2. DIAGRAMA DE FLUJO



5.3. GRAFO



<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 19
		Fecha: 15/02/2025

5.4. RUTAS

R1 : 1→2→3→4→5→6→7→8→12

R2 : 1→2→3→4→5→6→12

R3 : 1→2→3→4→9→10→12

R4 : 1→2→3→4→9→11→12

R5 : 1→2→12

5.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

$$V(G) = 4 + 1 = 5$$

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

$$V(G) = 15 - 12 + 2$$

$$V(G) = 5$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”	Caja Blanca	Versión: 4.0
		Página: 20
		Fecha: 15/02/2025

6. REQ 006 - Generar reporte de la disponibilidad de productos

6.1. CÓDIGO FUENTE

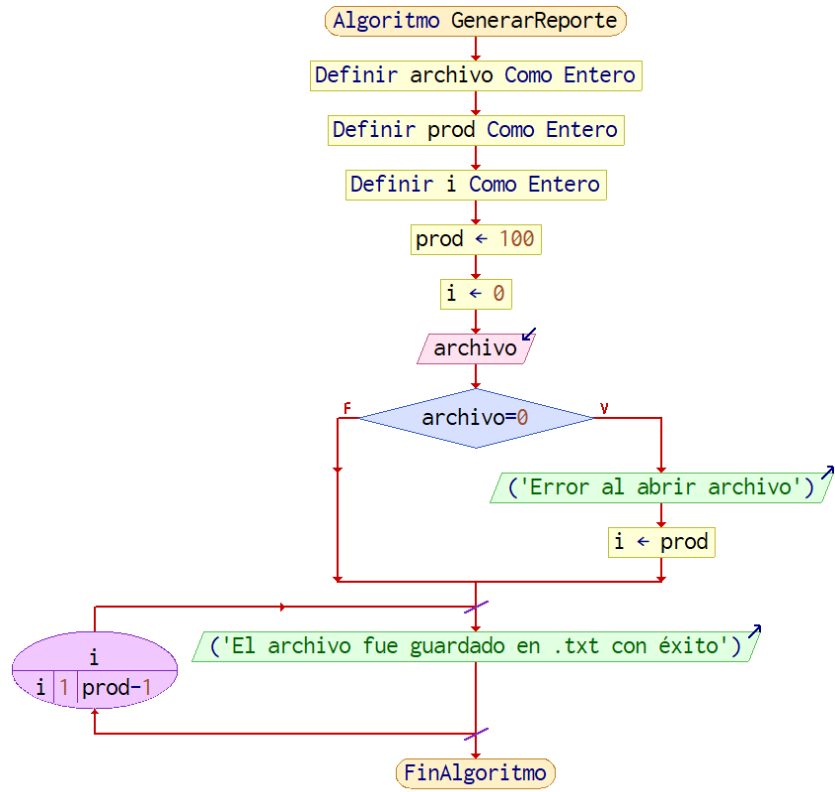
```

240 //Funcion para guardar el archivo
241
242 void guardar() {
243     // Abre el archivo en modo escritura
244     FILE *archivo = fopen("productos.txt", "w");
245     if (archivo == NULL) {
246         printf("Error al abrir el archivo.\n");
247         return;
248     }
249     for (int i = 0; i < prod; i++) {
250         fprintf(archivo, "%d: [Producto: %s]\n", i + 1, producto[i].product);
251     }
252     fclose(archivo);
253
254     printf("Lista de productos guardada exitosamente en 'productos.txt'.\n");
255 }
256

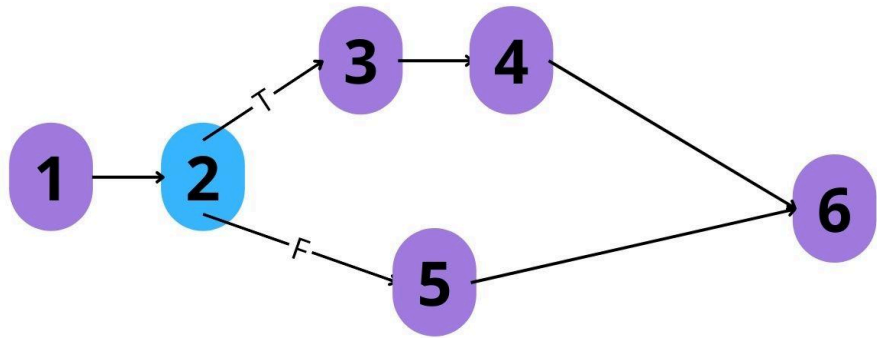
```

6.2. DIAGRAMA DE FLUJO

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 21
		Fecha: 15/02/2025



6.3. GRAFO



6.4. RUTAS

R1: 1 → 2 → 3 → 4 → 6

R2: 1 → 2 → 5 → 6

<i>Sistema automatizado de control de Inventario para Cafetería Vegana “Padmitos”</i>	Caja Blanca	Versión: 4.0
		Página: 22
		Fecha: 15/02/2025

6.5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$
 $V(G) = 6 - 6 + 2$
 $V(G) = 2$

DONDE:

P: Número de nodos predichado

A: Número de aristas

N: Número de nodos