|  |
| --- |
|  |
| Prueba de Caja Blanca |
| ***“Gestión de inventarios de una tienda de ropa”*** |
|  |
|  |
| **Integrantes:**  Pacha Shakira  Sivinta Jahir  Ulcuango Josselyn  **Fecha 2025-01-16** |

**Prueba caja blanca de**  describa el requisito funcional

1. **CÓDIGO FUENTE**

**#include <stdio.h>**

**#include <string.h>**

**struct Prenda {**

**char genero[10];**

**char talla[3];**

**char color[20];**

**int cantidad;**

**};**

**void ingresarDatos(struct Prenda \*p) {**

**// Ingresar genero (hombre o mujer)**

**printf("Ingrese el genero (Hombre/Mujer): ");**

**scanf("%s", p->genero);**

**while (strcmp(p->genero, "Hombre") != 0 && strcmp(p->genero, "Mujer") != 0) {**

**printf("Error: Ingrese un genero valido (Hombre/Mujer): ");**

**scanf("%s", p->genero);**

**}**

**// Ingresar talla (XS, S, M, L)**

**printf("Ingrese la talla (XS, S, M, L): ");**

**scanf("%s", p->talla);**

**while (strcmp(p->talla, "XS") != 0 && strcmp(p->talla, "S") != 0 && strcmp(p->talla, "M") != 0 && strcmp(p->talla, "L") != 0) {**

**printf("Error: Ingrese una talla valida (XS, S, M, L): ");**

**scanf("%s", p->talla);**

**}**

**// Ingresar color**

**printf("Ingrese el color de la prenda: ");**

**scanf("%s", p->color);**

**// Ingresar cantidad**

**printf("Ingrese la cantidad de la prenda: ");**

**scanf("%d", &p->cantidad);**

**while (p->cantidad <= 0) {**

**printf("Error: Ingrese una cantidad valida (mayor que 0): ");**

**scanf("%d", &p->cantidad);**

**}**

**}**

**void guardarDatos(struct Prenda p) {**

**// Aquí puedes guardar los datos en un archivo o base de datos**

**printf("\nDatos de la prenda guardados correctamente:\n");**

**printf("Genero: %s\n", p.genero);**

**printf("Talla: %s\n", p.talla);**

**printf("Color: %s\n", p.color);**

**printf("Cantidad: %d\n", p.cantidad);**

**}**

**void mostrarMenu() {**

**printf("\nSelecciona una opción:\n");**

**printf("1. Ingresar nueva prenda\n");**

**printf("2. Mostrar datos de la prenda\n");**

**printf("3. Salir\n");**

**}**

**int main() {**

**struct Prenda nuevaPrenda;**

**int opcion;**

**do {**

**mostrarMenu();**

**printf("Ingrese su opción: ");**

**scanf("%d", &opcion);**

**switch (opcion) {**

**case 1:**

**ingresarDatos(&nuevaPrenda);**

**guardarDatos(nuevaPrenda);**

**break;**

**case 2:**

**printf("\nDatos de la prenda:\n");**

**printf("Genero: %s\n", nuevaPrenda.genero);**

**printf("Talla: %s\n", nuevaPrenda.talla);**

**printf("Color: %s\n", nuevaPrenda.color);**

**printf("Cantidad: %d\n", nuevaPrenda.cantidad);**

**break;**

**case 3:**

**printf("Saliendo del programa.\n");**

**break;**

**default:**

**printf("Opción no válida. Intente nuevamente.\n");**

**}**

**} while (opcion != 3);**

**return 0;**

**}**

1. **DIAGRAMA DE FLUJO (DF)**

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

1. **GRAFO DE FLUJO (GF)**

**0**

1. **IDENTIFIACCIÒN DE LAS RUTAS (Camino basico)**

Determinar en base al GF del numeral 4

**RUTAS**

**R1:** 1,2,3,2

**R2:** 1,2,3,4,5,6,7,8,9,10,11,12,13,12,14,115,16,17,18,17,19,20,21,22,23,24,25,26

**R3:** 1,2,3,4,5,6,7,8,10,11,12,14,15,16,17,19,20,21,22,23,24,25,26

**R4:** 1,2,3,4,5,6,7,10,11,15,16,17,18,19,20,21,23,25,27

**R2:** 1,2,3,4,5,21,23,25,27

1. **COMPLEJIDAD CICLOMÁTICA**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)=17

* V(G) = A – N + 2

V(G)= 10

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

**1.- Código Fuente**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_PRENDAS 50**

**#define MAX\_CATEGORIAS 5**

**#define MAX\_GENEROS 2**

**#define MAX\_TALLAS 5**

**// Declaración de variables**

**char categoria[MAX\_PRENDAS][20];**

**char genero[MAX\_PRENDAS][10];**

**char talla[MAX\_PRENDAS][5];**

**int cantidad[MAX\_PRENDAS];**

**int main() {**

**// Inicialización de listas básicas**

**char categorias[MAX\_CATEGORIAS][20] = {"Jean", "Blusa", "Chaqueta", "Camiseta", "Vestido"};**

**char generos[MAX\_GENEROS][10] = {"Hombre", "Mujer"};**

**char tallas[MAX\_TALLAS][5] = {"XS", "S", "M", "L", "XL"};**

**// Inicialización del inventario**

**int totalPrendas = 0;**

**for (int i = 0; i < MAX\_CATEGORIAS; i++) {**

**for (int j = 0; j < MAX\_GENEROS; j++) {**

**for (int k = 0; k < MAX\_TALLAS; k++) {**

**strcpy(categoria[totalPrendas], categorias[i]);**

**strcpy(genero[totalPrendas], generos[j]);**

**strcpy(talla[totalPrendas], tallas[k]);**

**cantidad[totalPrendas] = 100; // Cantidad inicial de cada prenda**

**totalPrendas++;**

**}**

**}**

**}**

**// Entrada de filtros de búsqueda**

**char categoriaFiltro[20], generoFiltro[10], tallaFiltro[5];**

**printf("Filtro de Búsqueda:\n");**

**printf("Ingrese la categoría de prenda (Jean, Blusa, etc.) o deje en blanco para no filtrar:\n");**

**fgets(categoriaFiltro, sizeof(categoriaFiltro), stdin);**

**categoriaFiltro[strcspn(categoriaFiltro, "\n")] = 0; // Eliminar el salto de línea**

**printf("Ingrese el género (Hombre/Mujer) o deje en blanco para no filtrar:\n");**

**fgets(generoFiltro, sizeof(generoFiltro), stdin);**

**generoFiltro[strcspn(generoFiltro, "\n")] = 0; // Eliminar el salto de línea**

**printf("Ingrese la talla (XS, S, M, etc.) o deje en blanco para no filtrar:\n");**

**fgets(tallaFiltro, sizeof(tallaFiltro), stdin);**

**tallaFiltro[strcspn(tallaFiltro, "\n")] = 0; // Eliminar el salto de línea**

**// Proceso de filtrado**

**int encontrado = 0;**

**printf("\nResultados de la búsqueda:\n");**

**printf("--------------------------------------------\n");**

**for (int i = 0; i < totalPrendas; i++) {**

**// Verificar coincidencias con filtros**

**if ((strlen(categoriaFiltro) == 0 || strcmp(categoria[i], categoriaFiltro) == 0) &&**

**(strlen(generoFiltro) == 0 || strcmp(genero[i], generoFiltro) == 0) &&**

**(strlen(tallaFiltro) == 0 || strcmp(talla[i], tallaFiltro) == 0)) {**

**printf("%d. Categoría: %s, Género: %s, Talla: %s, Cantidad: %d\n",**

**i + 1, categoria[i], genero[i], talla[i], cantidad[i]);**

**encontrado = 1;**

**}**

**}**

**// Mostrar mensaje si no se encontraron resultados**

**if (!encontrado) {**

**printf("No se encontraron prendas que coincidan con los criterios.\n");**

**} else {**

**// Elección de la prenda**

**int prendaSeleccionada;**

**printf("Seleccione una prenda ingresando el número correspondiente:\n");**

**scanf("%d", &prendaSeleccionada);**

**// Verificación de selección válida**

**if (prendaSeleccionada < 1 || prendaSeleccionada > totalPrendas || cantidad[prendaSeleccionada - 1] <= 0) {**

**printf("Selección inválida. No se puede procesar.\n");**

**} else {**

**// Descontar del inventario**

**cantidad[prendaSeleccionada - 1]--;**

**printf("Has seleccionado la prenda: %s, Género: %s, Talla: %s\n",**

**categoria[prendaSeleccionada - 1], genero[prendaSeleccionada - 1], talla[prendaSeleccionada - 1]);**

**printf("Cantidad restante: %d\n", cantidad[prendaSeleccionada - 1]);**

**}**

**}**

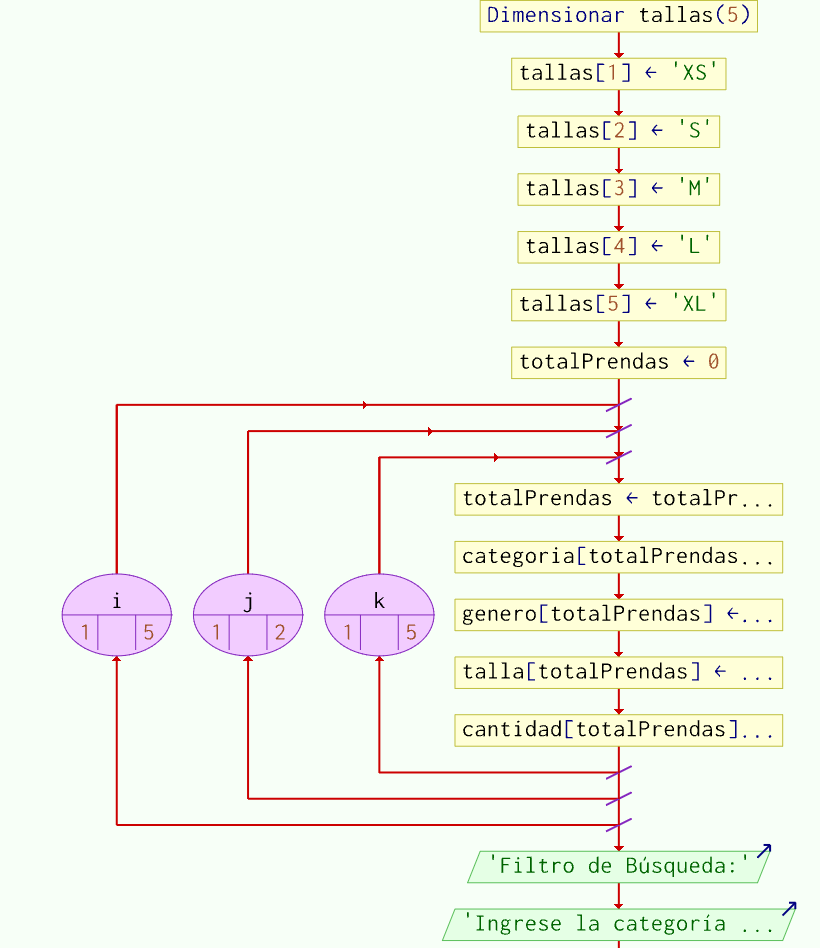
**return 0;**

**}**

**2.- Diagrama de Flujo**

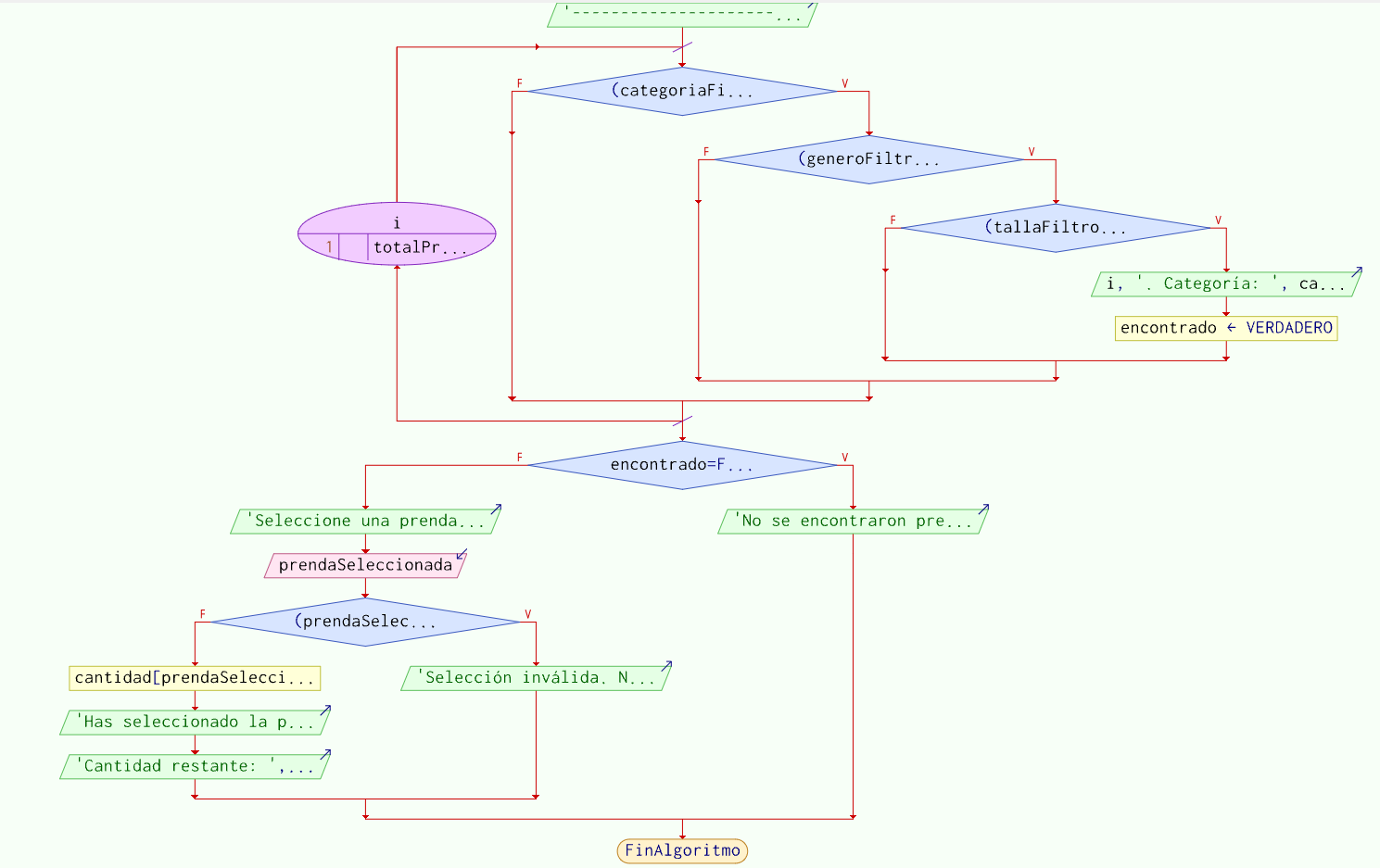
Tabla

Descripción generada automáticamente



Diagrama

Descripción generada automáticamente



3.- Grafo de flujo

**4.-Rutas**

**R1**: 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 16, 17

**R2**: 1, 2, 3, 4, 5, 6, 8, 8, 10, 11, 12, 13, 14, 16, 17

**R3:** 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 16, 17

**R4**: 1, 2, 3, 4, 5, 6, 8, 15, 16, 17

**5.- Complejidad Ciclomática**

* V(G) = número de nodos predicados(decisiones)+1

V(G)=10

* V(G) = A – N + 2

V(G)= 5

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos