**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**

**Facultad de Ingeniería, Arquitectura y Diseño**

**Ingeniero en Software y Tecnologías Emergentes**

Imagen que contiene Diagrama

Descripción generada automáticamente

**Nombre Alumno:**

Eliel Alfonso Ontiveros Ojeda

**Grupo:**

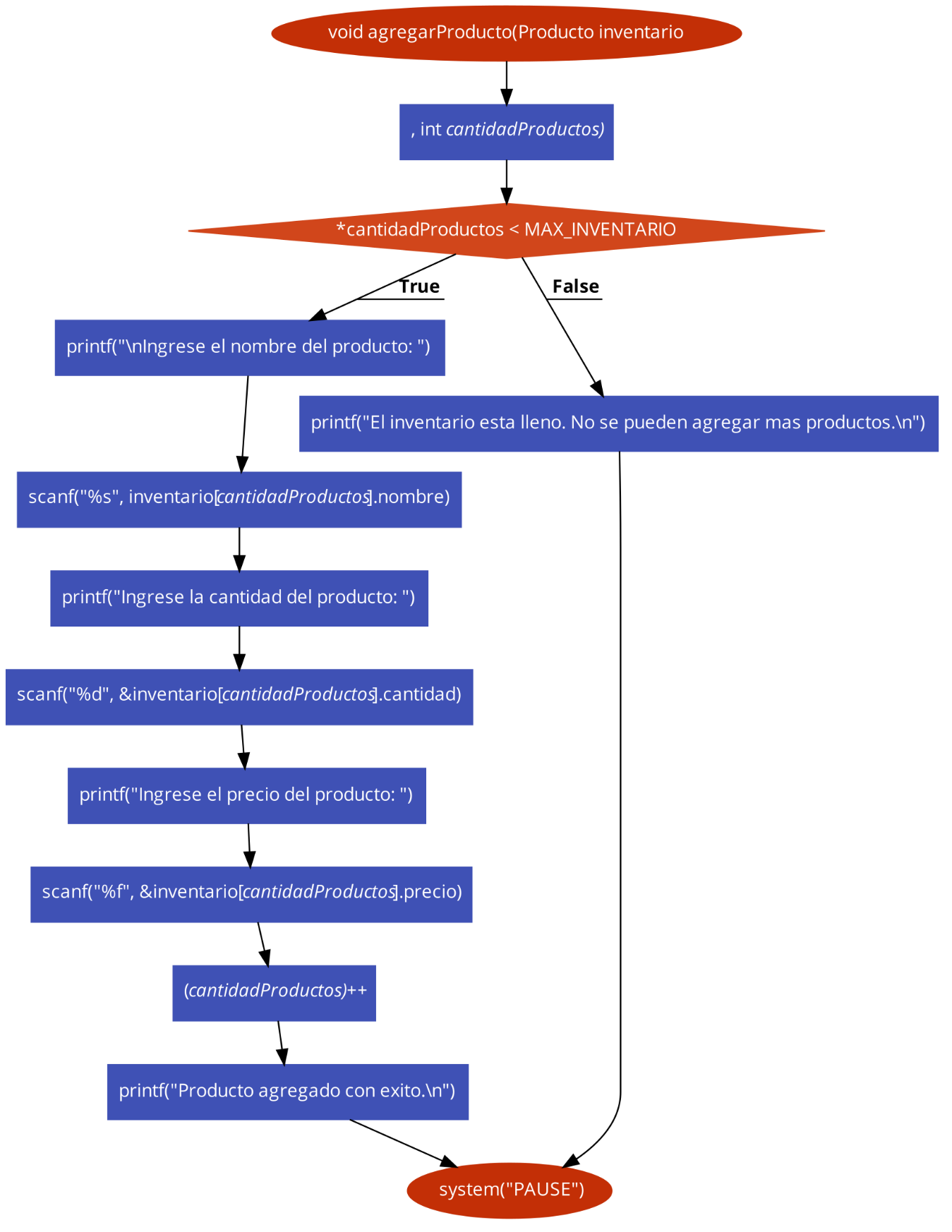
932

22/11/2023

a) Agregar elementos al inventario: Permite al usuario ingresar el nombre,

cantidad y precio del producto y agrega un nuevo elemento al inventario.

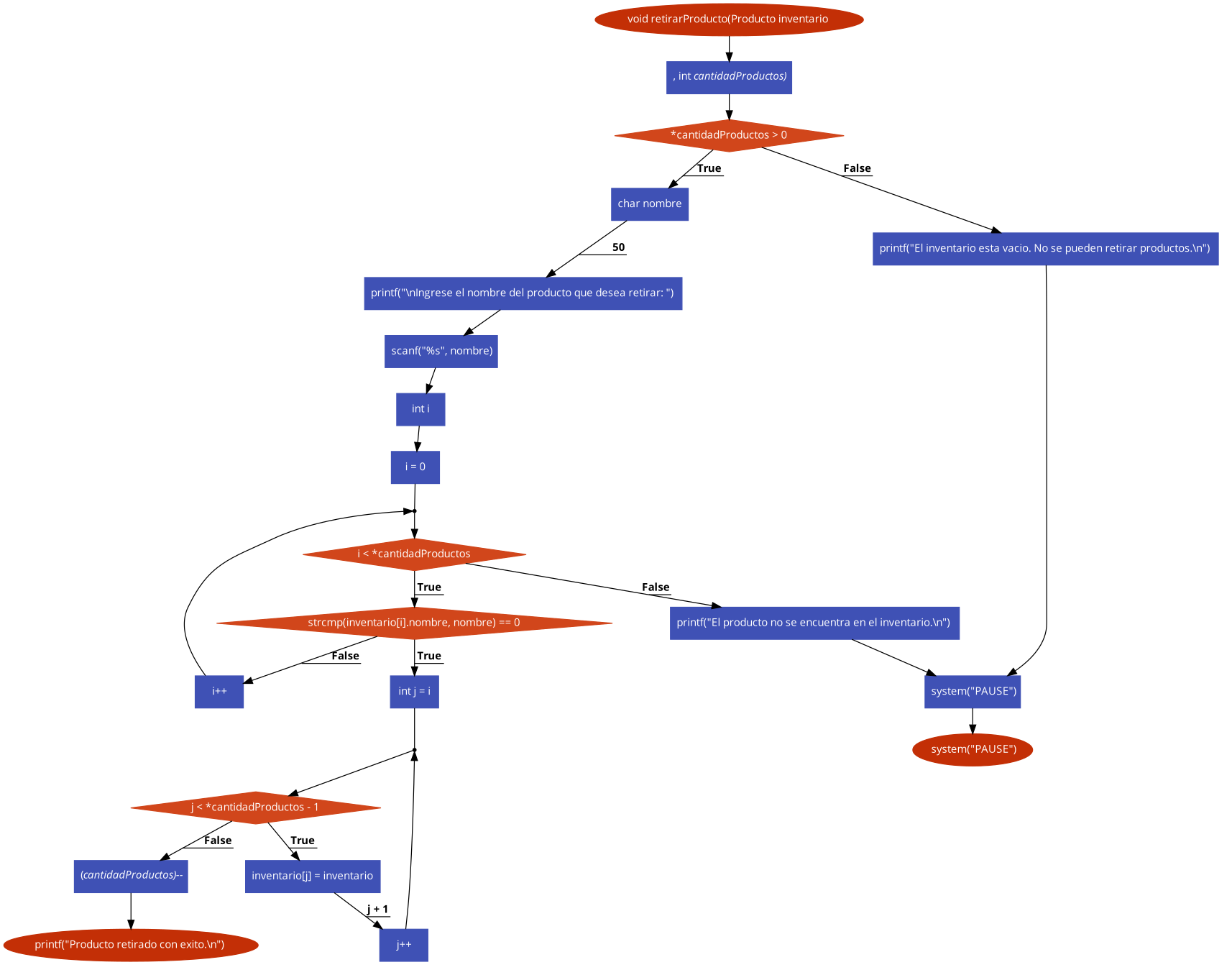
Asegúrate de manejar situaciones en las que el inventario esté lleno.



b) Retirar elementos del inventario: Permite al usuario ingresar el nombre del

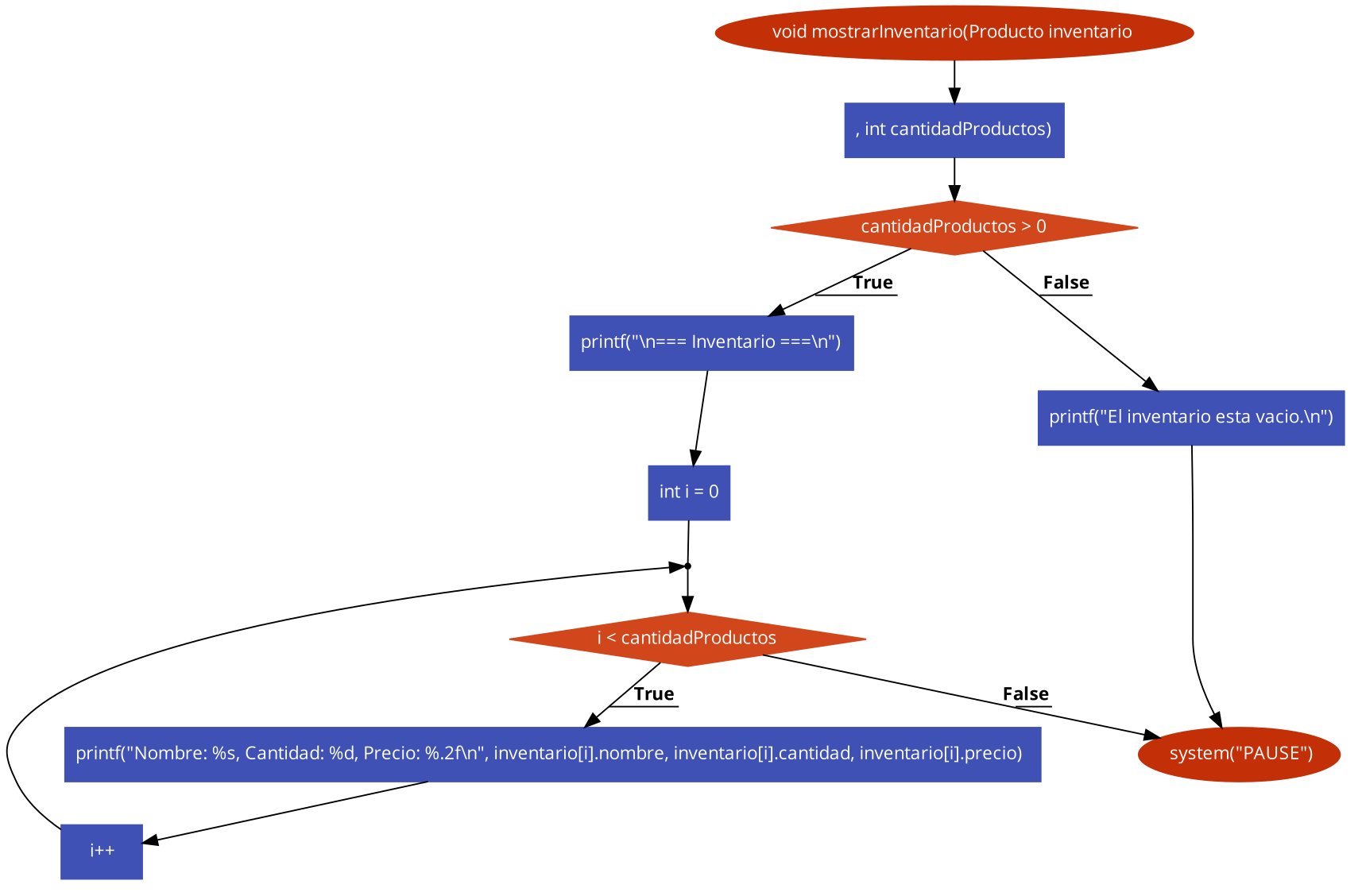
producto que desea retirar y elimina ese elemento del inventario. Asegúrate

de manejar casos en los que el elemento no esté en el inventario.



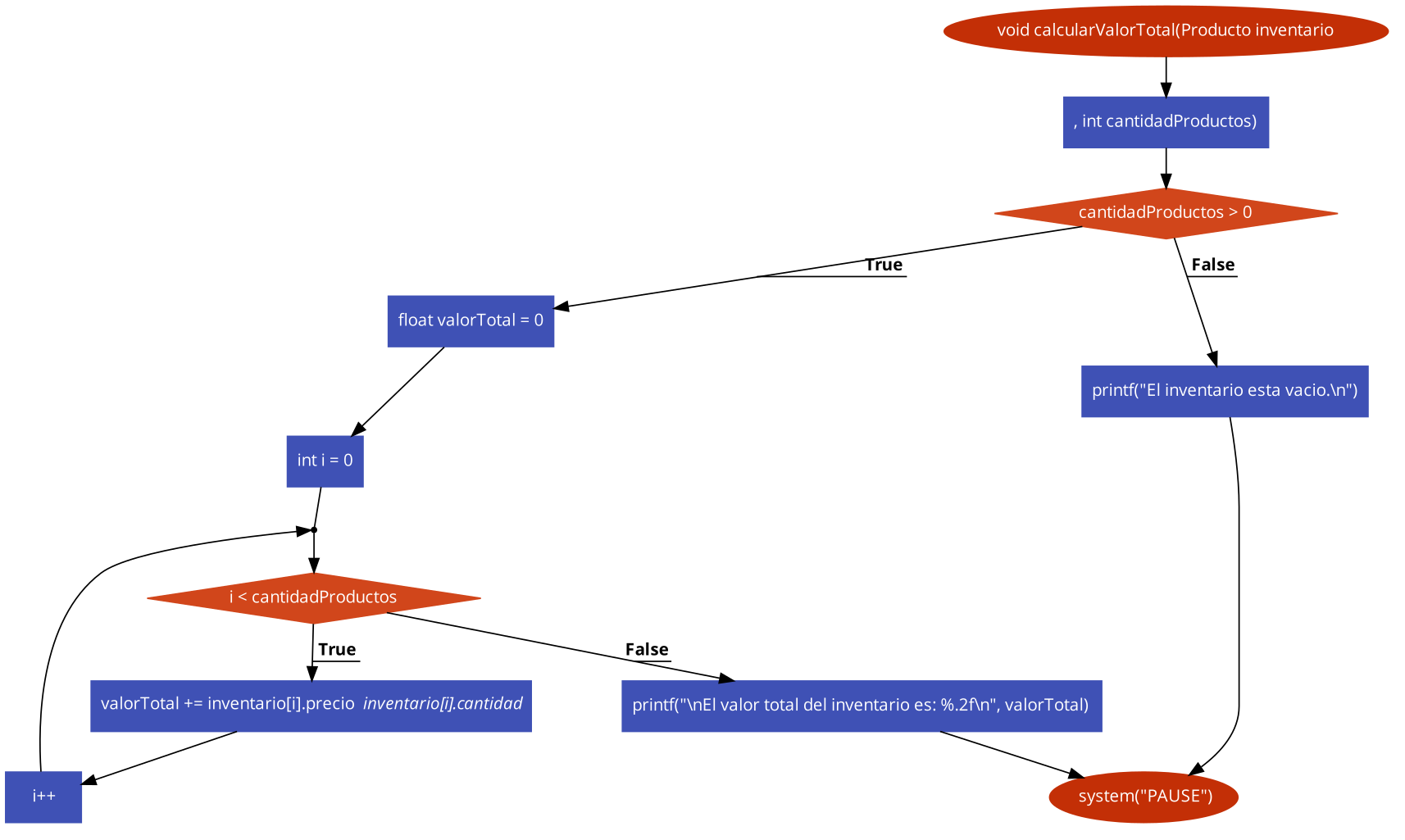
c) Mostrar el inventario: Muestra al usuario el contenido actual del inventario,

incluyendo el nombre, cantidad y precio de cada producto.

d) Calcular el valor total del inventario: Agrega una opción al menú que calcule

y muestre el valor total del inventario, que es la suma del precio de cada

producto multiplicado por su cantidad en stock.

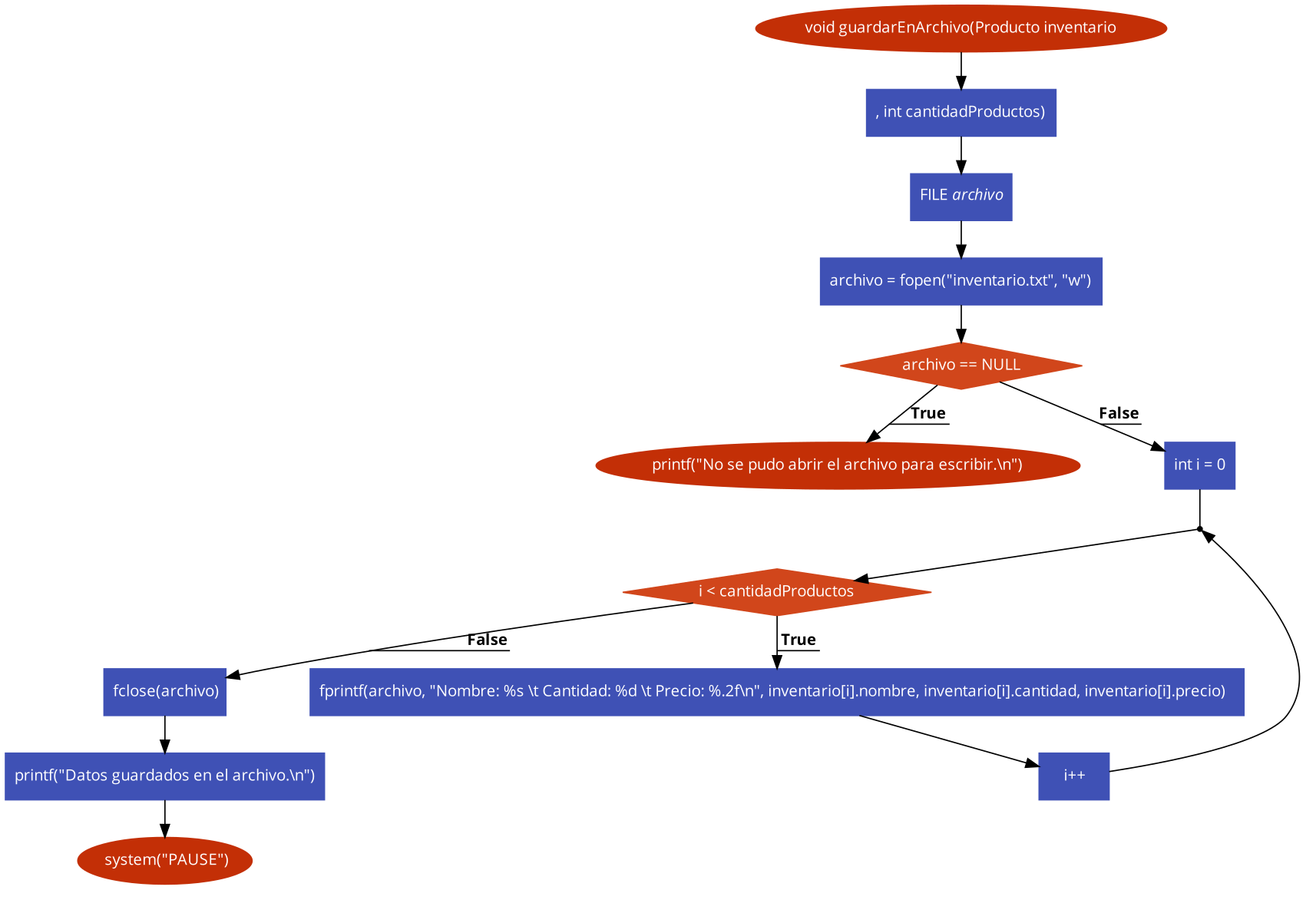
e) Función: guardarEnArchivo

Descripción: Guarda datos del inventario en un archivo.

Parámetros:

- inventario: Arreglo de productos a guardar.

- cantidadProductos: Número actual de productos en el inventario.

Valor de retorno: Ninguno.

f) Función: cargarDesdeArchivo

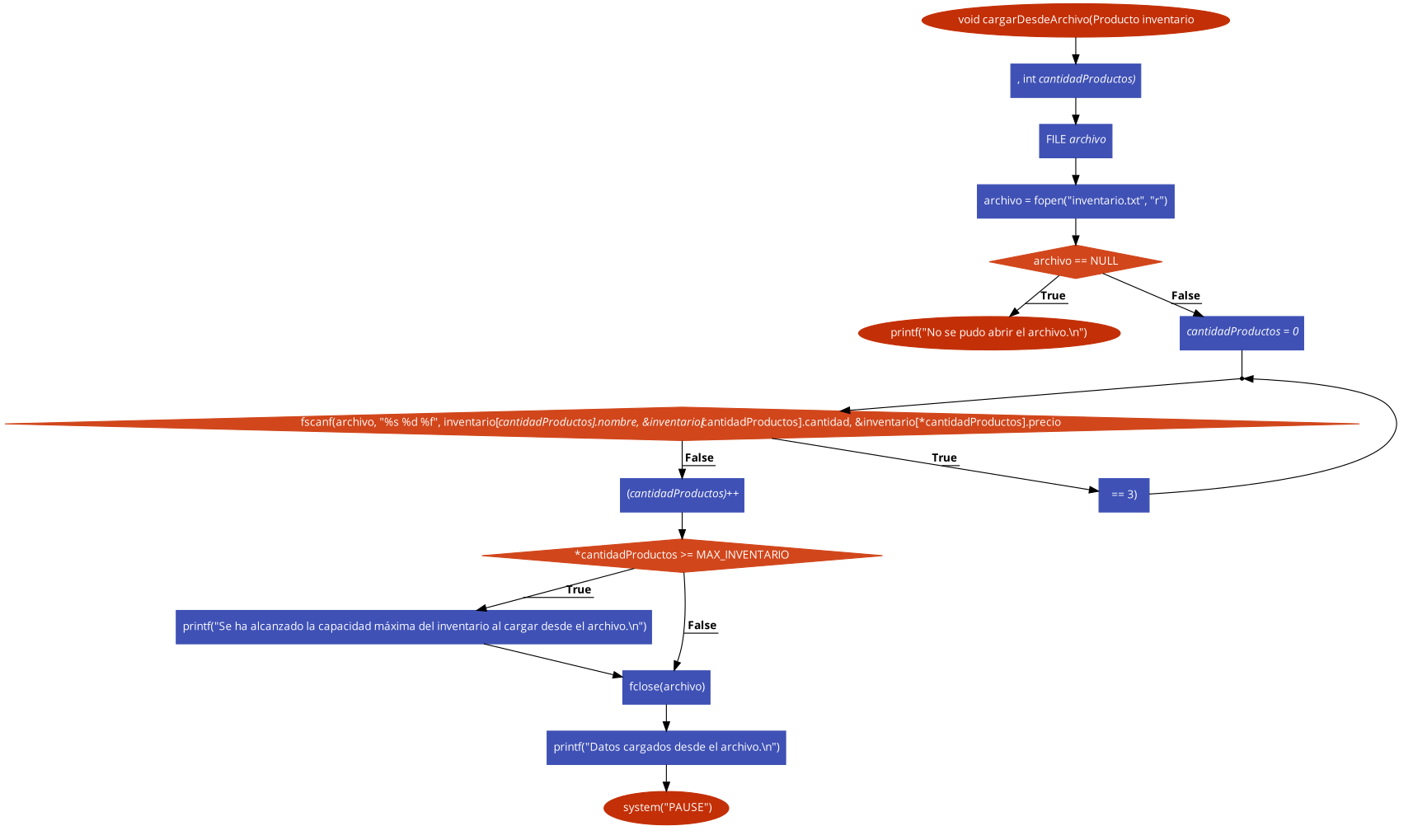
Descripción: Carga datos del inventario desde un archivo.

Parámetros:

- inventario: Arreglo de productos donde se cargarán los datos.

- cantidadProductos: Puntero al número actual de productos en el inventario.

Valor de retorno: Ninguno.



/\*

Sistema\_Inventario.cpp

Eliel Alfonso Ontiveros Ojeda\_368746

20/11/2023 - 22/11/2023

El objetivo de esta práctica es simular un sistema de gestión de inventario en C

utilizando estructuras (structs) donde los usuarios pueden agregar y retirar elementos

del inventario. También incorporaremos una instrucción de la Práctica 2 para mejorar

la funcionalidad del sistema de gestión de inventario.

\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_INVENTARIO 100

typedef struct

{

    char nombre[50];

    int cantidad;

    float precio;

} Producto;

void agregarProducto(Producto inventario[], int \*cantidadProductos);

void retirarProducto(Producto inventario[], int \*cantidadProductos);

void mostrarInventario(Producto inventario[], int cantidadProductos);

void calcularValorTotal(Producto inventario[], int cantidadProductos);

void ordenarInventario(Producto inventario[], int cantidadProductos, int opcion);

void cargarDesdeArchivo(Producto inventario[], int \*cantidadProductos);

void guardarEnArchivo(Producto inventario[], int cantidadProductos);

int main()

{

    Producto inventario[MAX\_INVENTARIO];

    int cantidadProductos = 0;

    int opcion;

    do

    {

        system("CLS");

        printf("\n=== Sistema de Gestion de Inventario ===\n");

        printf("1. Agregar Producto\n");

        printf("2. Retirar Producto\n");

        printf("3. Mostrar Inventario\n");

        printf("4. Calcular Valor Total del Inventario\n");

        printf("5. Ordenar Inventario\n");

        printf("6. Guardar en Archivo\n");

        printf("7. Cargar desde Archivo\n");

        printf("0. Salir\n");

        printf("Seleccione una opcion: ");

        scanf("%d", &opcion);

        switch (opcion)

        {

        case 1:

            agregarProducto(inventario, &cantidadProductos);

            break;

        case 2:

            retirarProducto(inventario, &cantidadProductos);

            break;

        case 3:

            mostrarInventario(inventario, cantidadProductos);

            break;

        case 4:

            calcularValorTotal(inventario, cantidadProductos);

            break;

        case 5:

            printf("\n1. Ordenar por Nombre\n2. Ordenar por Cantidad\n3. Ordenar por Precio\n");

            printf("Seleccione una opcion: ");

            scanf("%d", &opcion);

            ordenarInventario(inventario, cantidadProductos, opcion);

            break;

        case 6:

            guardarEnArchivo(inventario, cantidadProductos);

            break;

        case 7:

            cargarDesdeArchivo(inventario, &cantidadProductos);

            break;

        default:

        }

    } while (opcion != 0);

    printf("Saliendo del programa\n");

    return 0;

}

/\*

Funcion: agregarProducto

Descripcion: Agrega un producto al inventario.

Parametros:

- inventario: Arreglo de productos donde se agregará el nuevo producto.

- cantidadProductos: Puntero al número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void agregarProducto(Producto inventario[], int \*cantidadProductos)

{

    if (\*cantidadProductos < MAX\_INVENTARIO)

    {

        printf("\nIngrese el nombre del producto: ");

        scanf("%s", inventario[\*cantidadProductos].nombre);

        printf("Ingrese la cantidad del producto: ");

        scanf("%d", &inventario[\*cantidadProductos].cantidad);

        printf("Ingrese el precio del producto: ");

        scanf("%f", &inventario[\*cantidadProductos].precio);

        (\*cantidadProductos)++;

        printf("Producto agregado con exito.\n");

    }

    else

    {

        printf("El inventario esta lleno. No se pueden agregar mas productos.\n");

    }

    system("PAUSE");

}

/\*

Funcion: retirarProducto

Descripcion: Retira un producto del inventario.

Parametros:

- inventario: Arreglo de productos del cual se retirará el producto.

- cantidadProductos: Puntero al número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void retirarProducto(Producto inventario[], int \*cantidadProductos)

{

    if (\*cantidadProductos > 0)

    {

        char nombre[50];

        printf("\nIngrese el nombre del producto que desea retirar: ");

        scanf("%s", nombre);

        int i;

        for (i = 0; i < \*cantidadProductos; i++)

        {

            if (strcmp(inventario[i].nombre, nombre) == 0)

            {

                for (int j = i; j < \*cantidadProductos - 1; j++)

                {

                    inventario[j] = inventario[j + 1];

                }

                (\*cantidadProductos)--;

                printf("Producto retirado con exito.\n");

                return;

            }

        }

        printf("El producto no se encuentra en el inventario.\n");

    }

    else

    {

        printf("El inventario esta vacio. No se pueden retirar productos.\n");

    }

    system("PAUSE");

}

/\*

Funcion: mostrarInventario

Descripcion: Muestra los productos actuales en el inventario.

Parametros:

- inventario: Arreglo de productos a mostrar.

- cantidadProductos: Número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void mostrarInventario(Producto inventario[], int cantidadProductos)

{

    if (cantidadProductos > 0)

    {

        printf("\n=== Inventario ===\n");

        for (int i = 0; i < cantidadProductos; i++)

        {

            printf("Nombre: %s, Cantidad: %d, Precio: %.2f\n", inventario[i].nombre, inventario[i].cantidad, inventario[i].precio);

        }

    }

    else

    {

        printf("El inventario esta vacio.\n");

    }

    system("PAUSE");

}

/\*

Funcion: calcularValorTotal

Descripcion: Calcula el valor total del inventario.

Parametros:

- inventario: Arreglo de productos para calcular el valor total.

- cantidadProductos: Número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void calcularValorTotal(Producto inventario[], int cantidadProductos)

{

    if (cantidadProductos > 0)

    {

        float valorTotal = 0;

        for (int i = 0; i < cantidadProductos; i++)

        {

            valorTotal += inventario[i].precio \* inventario[i].cantidad;

        }

        printf("\nEl valor total del inventario es: %.2f\n", valorTotal);

    }

    else

    {

        printf("El inventario esta vacio.\n");

    }

    system("PAUSE");

}

/\*

Funcion: ordenarInventario

Descripcion: Ordena el inventario según un criterio seleccionado por el usuario.

Parametros:

- inventario: Arreglo de productos a ordenar.

- cantidadProductos: Número actual de productos en el inventario.

- opcion: Criterio de ordenamiento (1: nombre, 2: cantidad, 3: precio).

Valor de retorno: Ninguno.

\*/

void ordenarInventario(Producto inventario[], int cantidadProductos, int opcion)

{

    if (cantidadProductos > 0)

    {

        int i, j;

        Producto temp;

        switch (opcion)

        {

        case 1: // Ordenar por nombre

            for (i = 0; i < cantidadProductos - 1; i++)

            {

                for (j = 0; j < cantidadProductos - i - 1; j++)

                {

                    if (strcmp(inventario[j].nombre, inventario[j + 1].nombre) > 0)

                    {

                        temp = inventario[j];

                        inventario[j] = inventario[j + 1];

                        inventario[j + 1] = temp;

                    }

                }

            }

            printf("Inventario ordenado por nombre.\n");

            break;

        case 2: // Ordenar por cantidad

            for (i = 0; i < cantidadProductos - 1; i++)

            {

                for (j = 0; j < cantidadProductos - i - 1; j++)

                {

                    if (inventario[j].cantidad > inventario[j + 1].cantidad)

                    {

                        temp = inventario[j];

                        inventario[j] = inventario[j + 1];

                        inventario[j + 1] = temp;

                    }

                }

            }

            printf("Inventario ordenado por cantidad.\n");

            break;

        case 3: // Ordenar por precio

            for (i = 0; i < cantidadProductos - 1; i++)

            {

                for (j = 0; j < cantidadProductos - i - 1; j++)

                {

                    if (inventario[j].precio > inventario[j + 1].precio)

                    {

                        temp = inventario[j];

                        inventario[j] = inventario[j + 1];

                        inventario[j + 1] = temp;

                    }

                }

            }

            printf("Inventario ordenado por precio.\n");

            break;

        default:

            printf("Opcion de orden invalida.\n");

        }

    }

    else

    {

        printf("El inventario esta vacio.\n");

    }

    system("PAUSE");

}

/\*

Funcion: cargarDesdeArchivo

Descripcion: Carga datos del inventario desde un archivo.

Parametros:

- inventario: Arreglo de productos donde se cargarán los datos.

- cantidadProductos: Puntero al número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void cargarDesdeArchivo(Producto inventario[], int \*cantidadProductos)

{

    FILE \*archivo;

    archivo = fopen("inventario.txt", "r");

    if (archivo == NULL)

    {

        printf("No se pudo abrir el archivo.\n");

        return;

    }

    \*cantidadProductos = 0;

    while (fscanf(archivo, "%s %d %f", inventario[\*cantidadProductos].nombre, &inventario[\*cantidadProductos].cantidad, &inventario[\*cantidadProductos].precio) == 3)

    {

        (\*cantidadProductos)++;

        if (\*cantidadProductos >= MAX\_INVENTARIO)

        {

            printf("Se ha alcanzado la capacidad máxima del inventario al cargar desde el archivo.\n");

            break;

        }

    }

    fclose(archivo);

    printf("Datos cargados desde el archivo.\n");

    system("PAUSE");

}

/\*

Funcion: guardarEnArchivo

Descripcion: Guarda datos del inventario en un archivo.

Parametros:

- inventario: Arreglo de productos a guardar.

- cantidadProductos: Número actual de productos en el inventario.

Valor de retorno: Ninguno.

\*/

void guardarEnArchivo(Producto inventario[], int cantidadProductos)

{

    FILE \*archivo;

    archivo = fopen("inventario.txt", "w");

    if (archivo == NULL)

    {

        printf("No se pudo abrir el archivo para escribir.\n");

        return;

    }

    for (int i = 0; i < cantidadProductos; i++)

    {

        fprintf(archivo, "Nombre: %s \t Cantidad: %d \t Precio: %.2f\n", inventario[i].nombre, inventario[i].cantidad, inventario[i].precio);

    }

    fclose(archivo);

    printf("Datos guardados en el archivo.\n");

    system("PAUSE");

}