

Taller Colaborativo

ABET

Nombre de Integrantes: Andrés Mullo, Cristian Tutin

9

**Introducción**

**Una tienda de víveres se ha visto afectada debido a la poca efectiva forma de manejo del almacenamiento de los productos. Mediante el uso de agendas se ha mantenido la información de los productos disponibles en el local de forma manual, sin embargo, tanta información escrita en agendas se ha perdido y la confusión ha causado conflictos al momento de registrar los ingresos y la cantidad de productos que se contaban al principio. El uso de tácticas antiguas como las agendas no han dado los frutos esperados para los encargados de este negocio. Por lo tanto, se busca encontrar una manera moderna y eficiente para poder crear un sistema de inventario que guarde la información importante registrada por los encargados del negocio, de forma que no haya más problemas y el negocio pueda funcionar de una mejor manera.**

**Objetivos**

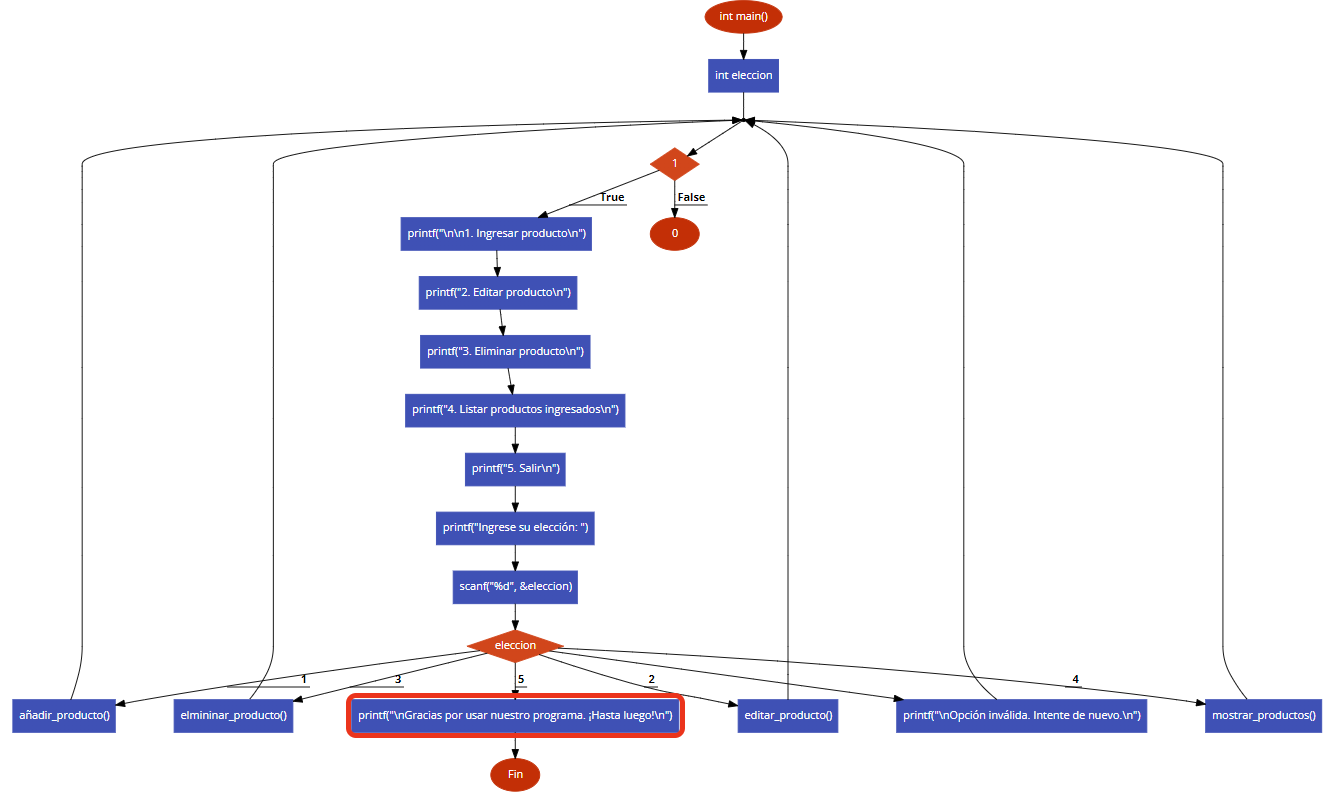
* Usar los conocimientos adquiridos para solucionar el problema y crear un sistema de inventario eficiente que facilite el almacenamiento de los productos.
* Exponer los resultados del programa propuesto y dar una clara explicación sobre el funcionamiento del programa.
* Analizar los métodos necesarios para que el programa pueda funcionar de mejor manera.

|  |  |
| --- | --- |
| **Nombre del sistema** | *Tienda minorista de víveres* |
| **Usuarios** | Encargados de la tienda |
| **Objetivo** | Almacenar información esencial (nombre, cantidad y precio) de los productos disponibles en la tienda |
| **Contexto del problema (Considerar los contextos económico, social y ambiental, dentro del sector productivo en el que funcionaría el sistema de inventarios)** | La administración del inventario es un tema central para evitar problemas financieros en las organizaciones, es un componente fundamental en la productividad de una empresa, ya que es el activo corriente de menor liquidez que manejan y que además contribuye a generar rentabilidad. Es el motor que mueve a la organización, pues es la base para la comercialización de la empresa que le permite obtener ganancias.  (Durán, Y. 2012. Pg.3) |
| **Restricciones (Consideraciones externas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: porcentaje de impuesto, registro sanitario u otros)** | Existen consideraciones externas mayormente enfocadas en como realizar correctamente un sistema de inventario que pueda beneficiar a la tienda en cuestión, ya que como todo negocio se necesita ser cauteloso al momento de administrar correctamente sus productos. se requiere del uso de diferentes técnicas de inventario, a fin de determinar su nivel óptimo y así disminuir los costos totales implicados en el inventario y optimizar las utilidades. |
| **Limitaciones (Consideraciones internas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: lenguaje de programación, tipos de almacenamiento u otros)** | Las consideraciones internas se relacionan al proceso del desarrollo del programa ya que, siendo un desarrollo, un tanto extenso se debe tener cuidado al momento al momento de usar las sentencias y las funciones, ya que un error puede dificultar la ejecución del programa. Además, que se debió adaptar el problema a las herramientas disponibles en el entorno de programación. |

|  |  |
| --- | --- |
| Variables de entrada | Descripción |
| 1. Nombre del producto:   Char nombres[PRODUCTOS\_MAXIMOS][50];   1. Precio del producto:   float precios[PRODUCTOS\_MAXIMOS];   1. Cantidad del producto   int cantidades[PRODUCTOS\_MAXIMOS];   1. Número de productos   int numero\_productos = 0; | 1. El nombre del producto es una variable de tipo char y es un arreglo bidimensional que almacena los nombres de la cantidad máxima de productos y que puede almacenar una cadena de hasta 49 caracteres, sin contar el espacio adicional para el carácter nulo. 2. El precio es una variable de tipo flotante que almacena el precio de los productos máximos que serían 100. 3. La cantidad es una variable de tipo entero que contiene la cantidad de productos máximos almacenados. 4. El número es una variable de tipo entero que pretende funcionar como un contador inicializado en 0. |
| Procesos | Descripción |
| 1. Uso de headers 2. Funciones:  * void añadir\_producto() * void editar\_producto() * void elmininar\_producto()  1. Función añadir producto 2. Función editar producto 3. Función eliminar producto 4. Función mostrar producto 5. Sentencia switch | 1. Los datos de entrada se colocaron en un archivo.h para poder crear una biblioteca que facilite el proceso principal que es permitir al usuario enlistar, añadir o quitar productos. Se definió una variable global que hace que los productos máximos sean 100. 2. En estas funciones se encuentra lo que el usuario desea realizar con el programa todas inician con una variable de tipo void ya que no se espera que se devuelva algún valor. 3. En esta función el usuario puede añadir productos hasta que llegue al limite de productos máximos. 4. Mediante la función strcpy que permite copiar una cadena de caracteres almacenados en una fuente hacia un destino, para así almacenar la nueva información del nombre, precio, cantidad de un producto en la matriz. 5. En la función eliminar producto, se pide al usuario el índice del producto que desea eliminar. Se usa una función strcpy y una sentencia for para eliminar un producto en cuestión y almacenar el restante. 6. La función permite mediante una sentencia for mostrar los productos enlistados. 7. En esta sentencia el usuario escoge la opción entre ingresar un producto, editar un producto, eliminar un producto, enlistar los productos ingresados o dar por terminado el programado dependiendo de cuál sea la elección del usuario. |
| Variables de Salida | Descripción |
| * Nombre del producto * Precio * Cantidad | El objetivo del programa es enlistar la información proporcionada por el usuario sobre los productos de la tienda, así que, como resultado el sistema de inventario debe mostrar los datos que proporciono el usuario. |

**Diagramas de flujo**

Para poder tener una idea de cómo realizar el algoritmo en el lenguaje C se desarrolló un simple diagrama de flujo para reflejar lo que vera el usuario, ósea escoger entre 5 opciones para que se de paso a la ejecución de las instrucciones que están contenidas en una función.



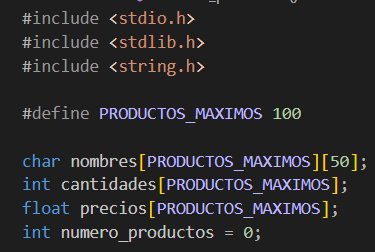
**GitHub**

<https://github.com/AndresMV10/TALLER-COLABORATIVO-ABET---JOSUE-MULLO---CRISTIAN-TUTIN>

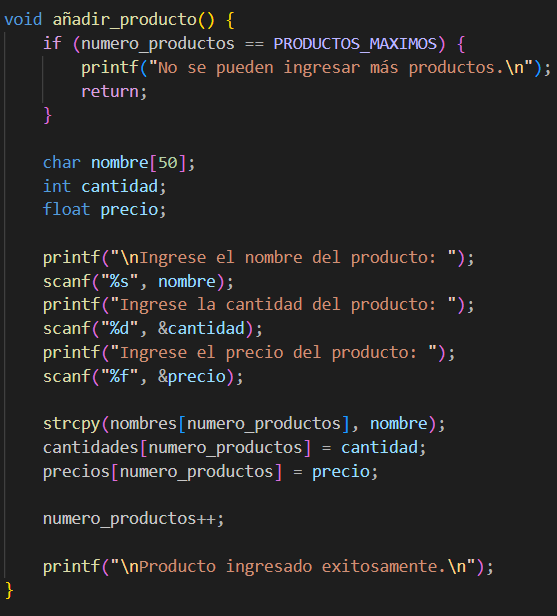
**Explicación del algoritmo implementado en C**

**Funcionalidades.h**

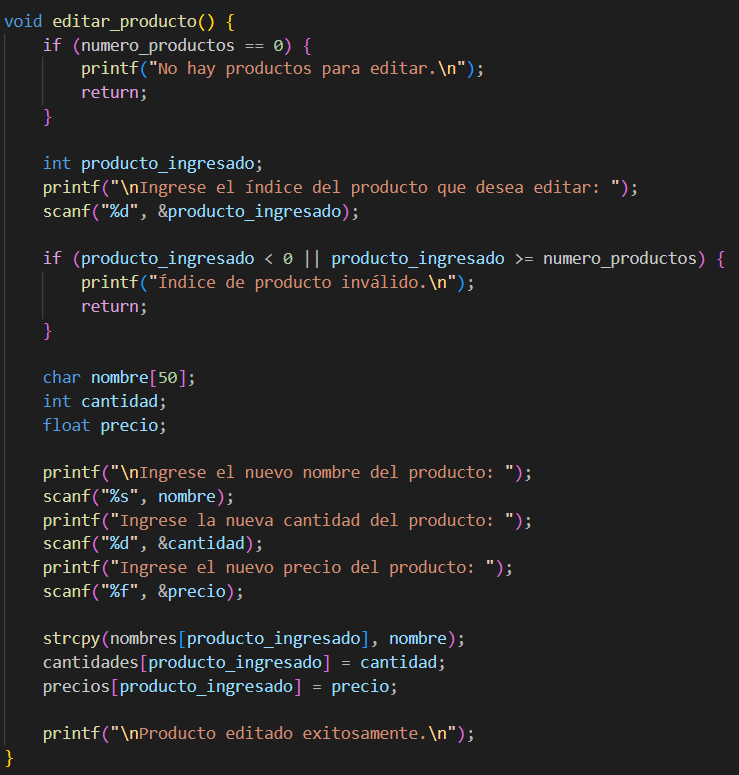
1. Este parte del código incluye las librerías necesarias para el desarrollo de las funciones, además declaramos algunas variables y arreglos globales, como también una macro que servirá como límite para ingresar productos.

****

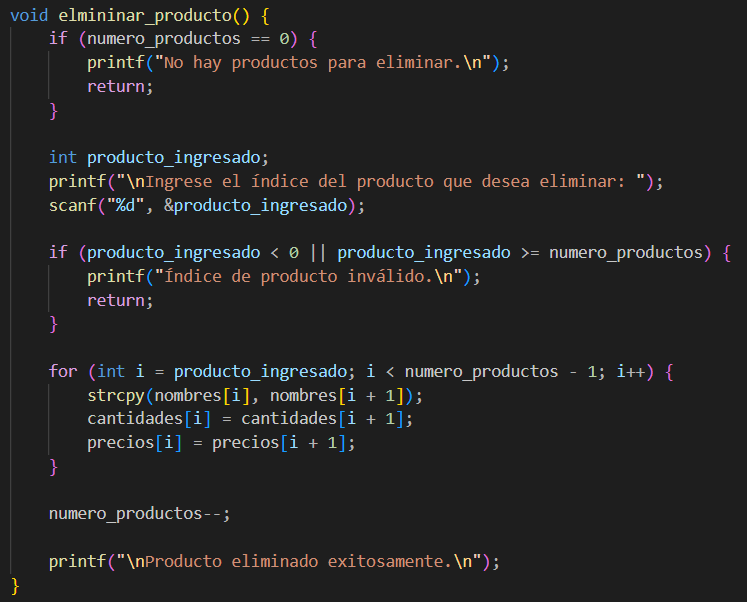
1. Hacemos uso de un procedimiento para implementar la funcionalidad de añadir productos, en la primera parte podemos ver una estructura if que nos ayudara a restringir la adicción de productos si se llega al máximo. Después declaramos algunas variables con las cuales les asignaremos datos mediante el usuario, por ultimo guardaremos la variable obtenida en su arreglo correspondiente, y también actualizamos el contador de números de productos para un próximo ingreso de datos. Consecuentemente mostraremos al usuario un mensaje para informar el ingreso de datos correcto.

****

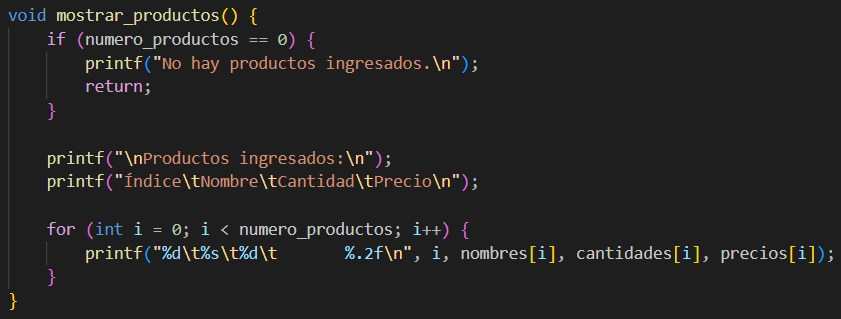
1. **Como en la explicación anterior, aquí también hacemos uso de un procedimiento, en este caso controlamos con un “if” que cuando no haya ningún producto ingresado, aparezca un mensaje informándonos que no hay producto para ingresar. En esta función declaramos una variable de tipo entera llamada “producto\_ingresado”, el cual nos ayudara a pedir el índice del producto que desea editar (el índice comienza en 0), después se hará una comprobación para ver si el índice que nos pide es menor a 0 o es mayor al número de productos existentes, lo cual no tendría sentido porque no existen, por lo tanto se nos mostrara un mensaje de índice de producto invalido, lo cual nos regresara al menú principal. Después de hacer la validación del índice, pediremos los datos nuevamente con variables antes vistas, y haremos el mismo procedimiento que en la función anterior, pero guardaremos en los arreglos en la posición correspondiente que nos indica el índice. Por ultimo mostraremos un mensaje información de que el producto se ha editado.**

****

1. **Comenzamos con la verificación de la existencia de productos con la estructura “if”, si no hay productos ingresados pues no se podrá eliminar. Seguiremos con la validación del indice, si es un indice existente o no, como vemos en la imagen nos mostrara un mensaje en caso de que el indice sea invalido. Despues mediante un bucle “for” recorreremos las listas de nombres, cantidades y precios a partir de la posición producto\_ingresado. Para cada elemento en estas listas, se copia el contenido del elemento siguiente en el mismo índice, efectivamente eliminando el producto en la posición producto\_ingresado. Por último se decrementa el contador “numero\_productos”, para asi poder reflejar la nueva cantidad de productos existentes, además de mostrar un mensaje indicando que se eliminó el producto.**

****

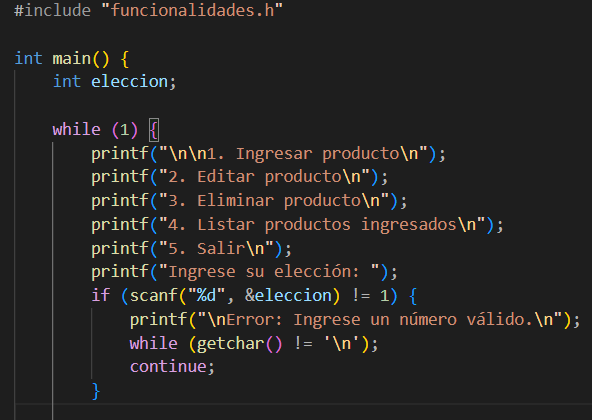
1. **En este procedimiento llamado “mostrar\_productos” primero verificamos si hay un producto para mostrar o no, después imprimos los productos ingresados, con ayuda de dos “printfs” y un bucle for, que recorrerá cada arreglo desde la posición cero hasta la cantidad de números existentes.**

****

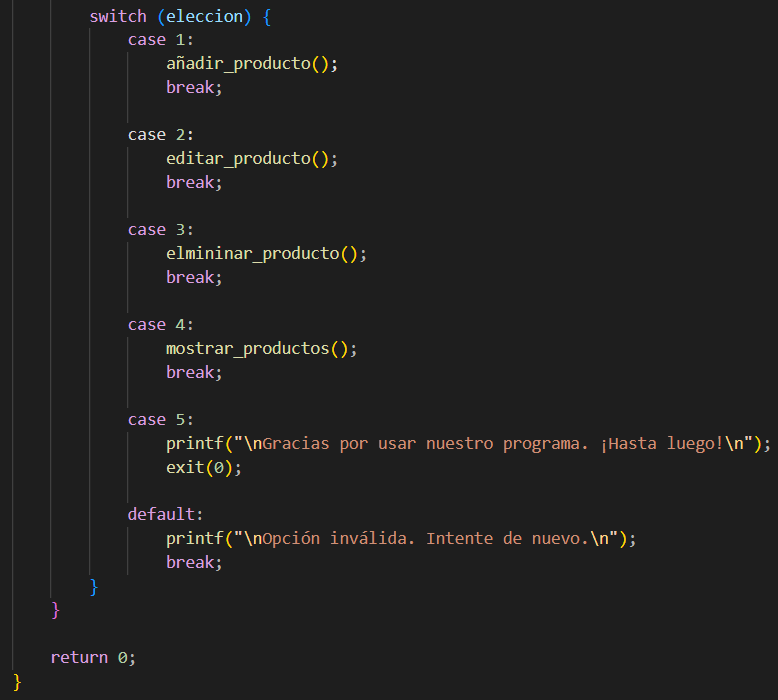
**Proyecto.c**1. Agregamos el archivo “funcionalidades”, la cual es una librería que contiene las funciones antes descritas.

2. Creamos un menú simple en el cual el usuario puede eligir entre las cinco opciones existentes mediante el uso de los números en el teclado, además controlamos que no se ingrese caracteres, puesto que este generaría un bucle infinito.

3. Controlamos que el ingreso de datos de la variable elección sea un número, mediante un if, que comparara si el dato obtenido mediante la función scanf de tipo entero es diferente de 1, pues en términos técnicos, el 1 es verdadero, y el 0 es falso, y como vemos en la imagen estamos pidiendo un dato de tipo entero, si no es un numero mostraremos un mensaje de error, después para limpiar el búfer de entrada, utilizaremos un bucle “while” que utilizará la función “getchar” para leer caracteres del búfer hasta que se encuentra con un salto de línea. Esto nos ayudara a comprobar que caracteres no deseados se limpie del búfer antes de continuar. Por ultimo con la instrucción “continue” se reiniciará el bucle while para solicitar una nueva entrada después de eliminar el carácter no deseado.

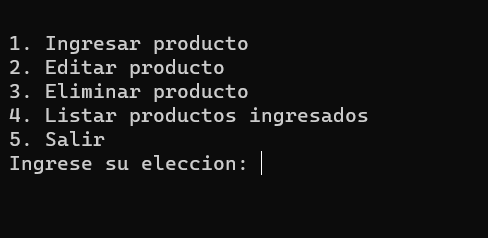
****

4. Ahora dentro de la estructura “while” y con el dato “elección” obtenido mediante el usuario, utilizaremos una estructura “switch” en la cual habrá 5 casos, los cuales se ejecutarán según la elección del usuario, además podemos ver que en en los primeros 4 casos están como instrucción ejecutar la función respectiva a lo que se pide, en el caso 5, que es la opción de salir, se mostrara un mensaje de salida, y cerraremos el programa mediante la instrucción “exit(0)”, y por ultimo tenemos el caso “default” que ocurrirá cuando el usuario ingrese un número que no ejecute ninguna instrucción, lo cual mostrada un mensaje de “intentar de nuevo” y nos llevara a la parte de pedir elegir al usuario que desea hacer.

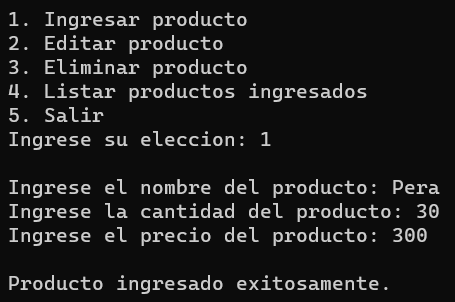
****

**Explicación de la ejecución de cada sección del programa**

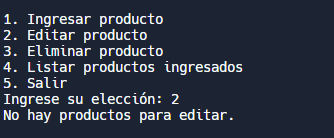
1. Al ejecutar el programa, nos aparece esta pantalla con un menú simple, en el cual hay 5 opciones a elegir por el usuario, y claro su mensaje pidiendo su elección.

****

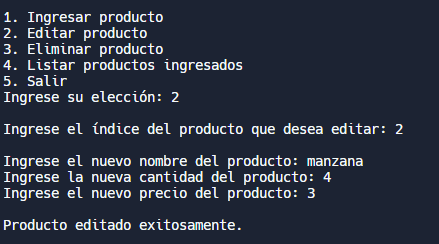
1. Si elegimos la primera opción, el programa nos pedirá el nombre, cantidad y precio del producto, consecuentemente nos aparecerá un mensaje de “producto ingresado “exitosamente” y se nos devolverá al menú.

****

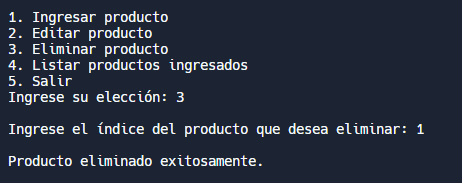
1. Si elegimos la segunda opción, nos pedirá ingresar el índice del producto que se quiere editar. Si no hay un producto ingresado entonces aparecerá el siguiente mensaje:



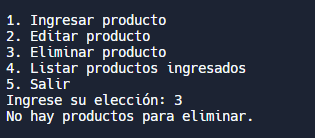
Por el contrario, si se ha ingresado cierta cantidad de productos al momento de editar la información podremos escoger entre los productos que se ingresaron y cambiar tanto el nombre, el precio y la cantidad para enlistarlo.

****

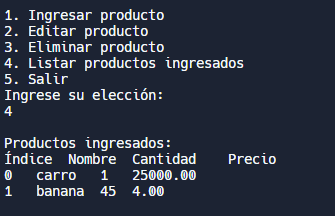
**4,-** Si se elige la tercera opción, nos pedirá ingresar el índice del producto que se desea eliminar de la lista:

****

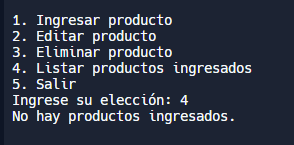
Si no hay productos que eliminar el programa mostrará el siguiente mensaje:

****

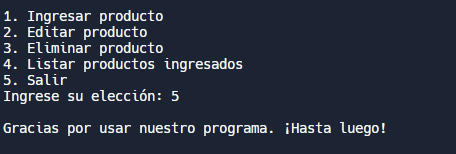
**5.-** Al elegir la cuarta opción, se desplegará la lista de los productos ingresados hasta el momento, junto con toda la información que se proporcionó.

****

Si no hay productos para mostrar el programa desplegará lo siguiente:

****

**6.-** Al escoger la quinta opción, el programa se dará por finalizado y nos agradecerá por su uso.

****

**Recomendaciones**

* Colocar las librerías principales, datos de salida y parte del proceso en un header ya que de esa manera se ahorrará espacio en las líneas de código del programa en C, para poder evitar errores y tener una mejor organización.
* Usar una sentencia switch para poder crear el menú, de las sentencias de repetición es el que tiene una estructura que puede facilitar el proceso.
* Las estructuras de repetición if son muy útiles para evitar que se ingresen datos erróneos al momento de pedir información del producto. Por ejemplo, que se ingresen números cuando se pide el nombre.

**Conclusiones**

En conclusión, el sistema de inventario es necesario para facilitar el almacenamiento de los productos de una tienda, para realizar correctamente este sistema se debe tener en cuenta el uso correcto de las sentencias de repetición, de condición y las funciones ya que de esta forma el código puede resultar más funcional.