

SEMANA 8

TALLER COLABORATIVO

Integrantes: Francisco Mejia, Paul Chuquitarco, Adrian Llerena

1. Establezca grupos de trabajo de máximo 3 integrantes.
2. Lea detenidamente el siguiente problema:

Se desea desarrollar un sistema de inventarios, el cual implemente las siguientes funcionalidades:

- Ingresar, editar y eliminar productos.
- Listar productos ingresados.

Los productos ingresados deberán tener mínimo las siguientes variables:

- Nombre del producto
- Cantidad
- Precio

3. Formula el problema identificando sus principales variables, considerando un sector productivo donde podría utilizar un sistema de inventarios. Por ejemplo: venta de productos para mascotas, venta de fármacos, tienda minorista de viveres, tienda de repuestos de vehículos u otros. Llenar las siguientes tablas con la identificación del problema.

- Formulación del Problema

Nombre del sistema:	Tutienda
Objetivos:	Objetivo General: El objetivo general del programa es desarrollar un sistema de gestión de

	<p>inventario para una tienda que permita al usuario ingresar, visualizar, editar y eliminar productos de manera eficiente y sin errores. Este sistema ayudará a mantener un registro organizado de los productos disponibles y facilitará la administración del inventario. A continuación se detallan tres objetivos específicos que contribuyen al logro del objetivo general:</p> <p>Objetivos Específicos:</p> <p>Ingreso Eficiente de Productos:</p> <p>Implementar una interfaz de usuario amigable que solicite al usuario ingresar la cantidad de productos y los detalles de cada uno. Validar los datos ingresados para garantizar la integridad y consistencia del inventario.</p> <p>Visualización Clara y Completa de Productos:</p> <p>Desarrollar una función que muestre de manera clara y completa todos los productos disponibles, incluyendo su nombre, precio y cantidad. Diseñar la presentación de los productos de forma organizada y legible para facilitar la comprensión del inventario.</p> <p>Edición y Eliminación Precisa de Productos:</p> <p>Implementar funciones que permitan al usuario editar la información de un producto seleccionado, incluyendo su nombre, precio y cantidad. Desarrollar un mecanismo para eliminar un producto específico del inventario y ajustar correctamente la lista de productos restantes.</p>
<p>Contexto del problema (Considerar los contextos económico, social y ambiental, dentro del sector productivo en el que funcionaría el sistema de inventarios)</p>	<p>1. Contexto Económico:</p> <p>Competencia y Eficiencia: En un entorno económico altamente competitivo, las empresas deben optimizar sus procesos para mantenerse eficientes y rentables. Un sistema de gestión de inventario efectivo</p>

	<p>permite a las empresas minimizar costos al evitar la sobrecompra o la falta de existencias.</p> <p>Maximización de Utilidades: La eficiente gestión del inventario permite a las empresas maximizar sus utilidades al garantizar un equilibrio adecuado entre la oferta y la demanda, evitando la obsolescencia de productos y reduciendo los costos asociados con el almacenamiento excesivo.</p> <p>Tendencias del Mercado: Los datos generados por el sistema de inventario pueden proporcionar información valiosa sobre las tendencias del mercado y las preferencias de los clientes, lo que ayuda a las empresas a tomar decisiones informadas sobre la producción y el abastecimiento de productos.</p> <p>2. Contexto Social:</p> <p>Experiencia del Cliente: Un inventario bien gestionado contribuye a una experiencia positiva del cliente al garantizar la disponibilidad de productos deseados y reducir los tiempos de espera.</p> <p>Empleo y Estabilidad Laboral: La eficiencia en la gestión del inventario puede tener un impacto positivo en la estabilidad laboral al garantizar una demanda constante de productos y, por lo tanto, la continuidad en la producción y el empleo.</p> <p>Acceso a Productos: Un sistema de gestión de inventario efectivo asegura que los productos estén disponibles para satisfacer las necesidades de la comunidad, contribuyendo así al bienestar social.</p> <p>3. Contexto Ambiental:</p> <p>Reducción de Desperdicios: Una gestión eficiente del inventario ayuda a reducir los desperdicios al minimizar la obsolescencia de productos y evitar la sobreproducción.</p> <p>Optimización de Recursos: Al prevenir la escasez y el exceso de inventario, se reduce el uso innecesario de recursos naturales y energéticos asociados con la producción y el almacenamiento de productos.</p>
--	--

<p>Restricciones (Consideraciones externas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: porcentaje de impuesto, registro sanitario u otros)</p>	<p>Regulaciones Legales y Fiscales:</p> <p>Cumplimiento de normativas legales y fiscales locales relacionadas con la gestión de inventarios, como el registro de productos, impuestos sobre ventas, tasas de importación/exportación, entre otros.</p> <p>Respeto a las normativas de protección al consumidor, incluyendo etiquetado de productos, advertencias de seguridad y fecha de caducidad.</p> <p>Normativas Sanitarias y de Seguridad:</p> <p>Obtención de los registros sanitarios y certificaciones necesarias para la venta de productos específicos, especialmente aquellos relacionados con alimentos, productos farmacéuticos, cosméticos, etc.</p> <p>Cumplimiento de normativas de seguridad y control de calidad para garantizar que los productos no representen riesgos para la salud o seguridad de los consumidores.</p> <p>Restricciones de Almacenamiento y Transporte:</p> <p>Consideración de las limitaciones físicas del espacio de almacenamiento, como capacidad de almacenamiento, condiciones de temperatura y humedad, para garantizar la integridad de los productos.</p> <p>Cumplimiento de regulaciones de transporte y logística para el manejo adecuado de productos perecederos, peligrosos o sensibles.</p>

	<p>Costos y Presupuesto:</p> <p>Limitaciones presupuestarias para el desarrollo, implementación y mantenimiento del sistema de gestión de inventario, incluyendo costos de software, hardware, capacitación y soporte técnico. Evaluación de los costos asociados con el almacenamiento y mantenimiento de inventario, así como los costos de adquisición de nuevos productos y reposición de existencias.</p>
<p>Limitaciones (Consideraciones internas que se deben tener para diseñar y desarrollar el proyecto, ejemplo: lenguaje de programación, tipos de almacenamiento u otros)</p>	<p>Tecnología y Plataforma:</p> <p>Selección del lenguaje de programación y plataforma de desarrollo adecuados, teniendo en cuenta las habilidades del equipo de desarrollo, los requisitos de rendimiento y las necesidades del sistema.</p> <p>Limitaciones de hardware y software que puedan afectar la implementación del sistema, como capacidades de almacenamiento, potencia de procesamiento y compatibilidad con sistemas operativos.</p> <p>Requisitos Funcionales y Especificaciones:</p> <p>Definición clara de los requisitos funcionales y especificaciones del sistema, incluyendo funciones de ingreso de datos, consulta de inventario, generación de informes, etc.</p> <p>Limitaciones de tiempo y recursos para el desarrollo, que pueden afectar la cantidad y complejidad de las características que se pueden implementar.</p> <p>Diseño y Arquitectura del Sistema:</p>

	<p>Limitaciones de diseño y arquitectura del sistema, como la escalabilidad, la modularidad y la mantenibilidad, que pueden afectar la capacidad del sistema para adaptarse a cambios futuros y crecer con las necesidades del negocio.</p> <p>Consideraciones de seguridad y privacidad que deben ser integradas en el diseño del sistema para proteger los datos sensibles del inventario.</p>
--	--

- Identificación de variables entrada, salida y procesos asociados.

Variables de entrada

Variables de entrada	Descripción
Cantidad de Productos (cantidadProduct):	<p>Número de productos que el usuario desea ingresar al sistema.</p> <p>Tipo de Datos: Entero.</p> <p>Proceso Asociado: Determina la cantidad de veces que se ejecuta la función de leer producto para capturar los detalles de cada producto.</p>
Detalles de Productos (nombre, precio, cantidad)	<p>Detalles de cada producto, incluyendo su nombre, precio y cantidad.</p> <p>Tipo de Datos: Cadena (nombre), flotante (precio), entero (cantidad).</p> <p>Proceso Asociado: Se ingresan manualmente por el usuario durante la función de leer producto.</p>
Opción del Menú (opción)	<p>Selección realizada por el usuario en el menú de opciones (mostrar, editar, eliminar, salir).</p> <p>Tipo de Datos: Entero.</p> <p>Proceso Asociado: Determina la acción que se ejecutará en función de la elección del usuario.</p>
Índice del Producto (índice)	<p>Identificador del producto seleccionado por el usuario para editar o eliminar.</p> <p>Tipo de Datos: Entero.</p> <p>Proceso Asociado: Utilizado para identificar el producto específico en las funciones de editar y eliminar.</p>

Entradas de Menú (opciones de menú):	<p>Entradas adicionales necesarias para seleccionar una opción de menú específica.</p> <p>Tipo de Datos: Depende de la opción seleccionada (entero, cadena, etc.).</p> <p>Proceso Asociado: Captura las selecciones adicionales del usuario, como el índice del producto a editar o eliminar.</p>
--------------------------------------	---

Variables de Salida

Variables de Salida	Descripción
Lista de Productos (Productos):	<p>Lista completa de productos ingresados en el sistema, incluyendo sus detalles (nombre, precio, cantidad).</p> <p>Tipo de Datos: Matriz de cadenas.</p> <p>Proceso Asociado: Mostrada al usuario como resultado de la función de mostrar productos.</p>
Mostrar Productos (mostrarProductos)	<p>Descripción: Proceso para mostrar la lista completa de productos almacenados en el sistema.</p> <p>Entradas: Ninguna.</p> <p>Salidas: Lista de productos (Productos).</p> <p>Variables Asociadas: Productos.</p>
Editar Producto (editarProducto)	<p>Proceso para editar los detalles de un producto específico seleccionado por el usuario.</p> <p>Entradas: Índice del producto a editar, nuevos detalles del producto.</p> <p>Salidas: Detalles actualizados del producto.</p> <p>Variables Asociadas: indice, nombre, precio, cantidad.</p>

Procesos

Procesos	Descripcion
Leer Producto:	<p>Este proceso permite al usuario ingresar los detalles de cada producto en el sistema, incluyendo el nombre, precio y cantidad. Se solicita al usuario que ingrese la cantidad de productos que desea ingresar, y luego se capturan los detalles de cada producto uno por uno.</p> <p>Entradas: Cantidad de productos a ingresar.</p> <p>Salidas: Detalles de cada producto (nombre, precio, cantidad).</p>

	Variables Asociadas: cantidadProduct, nombre, precio, cantidad.
Mostrar Producto	<p>Este proceso muestra al usuario la lista completa de productos almacenados en el sistema, incluyendo sus detalles (nombre, precio, cantidad). La lista se presenta de manera clara y legible para facilitar la visualización y comprensión.</p> <p>Entradas: Ninguna.</p> <p>Salidas: Lista de productos.</p> <p>Variables Asociadas: Productos.</p>
Editar Producto	<p>Este proceso permite al usuario editar los detalles de un producto específico seleccionado. Se solicita al usuario que ingrese el índice del producto que desea editar, y luego se le permite modificar el nombre, precio y cantidad del producto según sea necesario.</p> <p>Entradas: Índice del producto a editar, nuevos detalles del producto.</p> <p>Salidas: Detalles actualizados del producto.</p> <p>Variables Asociadas: indice, nombre, precio, cantidad.</p>
Eliminar Producto	<p>Este proceso permite al usuario eliminar un producto específico seleccionado del sistema. Se solicita al usuario que ingrese el índice del producto que desea eliminar, y luego se elimina el producto de la lista de productos.</p> <p>Entradas: Índice del producto a eliminar.</p> <p>Salidas: Lista actualizada de productos después de la eliminación.</p> <p>Variables Asociadas: indice.</p>

INFORME

Introducción: En el contexto de la administración empresarial, la gestión eficiente del inventario es crucial para garantizar un flujo de operaciones suave y rentable. Los sistemas de gestión de inventario automatizados ofrecen una solución efectiva para manejar la entrada, salida y control de productos en una tienda o empresa. En este informe, se presenta un sistema de gestión de inventario diseñado para ayudar a las empresas a optimizar sus procesos de inventario y mejorar su eficiencia operativa.

Objetivos del Trabajo: El objetivo principal de este trabajo es diseñar y desarrollar un sistema de gestión de inventario que permita a las empresas ingresar, visualizar, editar y eliminar productos de manera eficiente y precisa. Los objetivos específicos incluyen:

4. Identificar las variables de entrada, salida y procesos asociados en el sistema de gestión de inventario.
5. Formular el problema a resolver por el sistema de gestión de inventario.
6. Desarrollar tablas que describan la formulación del problema y la identificación de variables.

Planteamiento de propuestas de solución.

Opción 1:

<https://github.com/Leixter/tt.git>

Introducción: El programa desarrollado es una herramienta de gestión de productos diseñada para ayudar a administrar inventarios en una tienda. Facilita la tarea de agregar, editar, eliminar y mostrar productos, proporcionando una interfaz fácil de usar para el usuario.

Objetivos del trabajo: El objetivo principal del proyecto es crear un sistema eficiente para la gestión de productos que permita a los usuarios llevar un registro de los productos disponibles en una tienda. Los objetivos específicos incluyen:

1. Desarrollar un programa que permita al usuario ingresar la cantidad de productos que desea administrar.
2. Implementar funciones para agregar, editar y eliminar productos.
3. Proporcionar una interfaz de usuario intuitiva y amigable.
4. Garantizar la integridad y seguridad de los datos del inventario.

Opción 2:

<https://github.com/Leixter/Code1.git>

Introducción: En el contexto de la administración empresarial, la gestión eficiente del inventario es crucial para garantizar un flujo de operaciones suave y rentable. Los sistemas de gestión de inventario automatizados ofrecen una solución efectiva para manejar la entrada, salida y control de productos en una tienda o empresa. En este informe, se presenta un

sistema de gestión de inventario diseñado para ayudar a las empresas a optimizar sus procesos de inventario y mejorar su eficiencia operativa.

Objetivos del Trabajo: El objetivo principal de este trabajo es diseñar y desarrollar un sistema de gestión de inventario que permita a las empresas ingresar, visualizar, editar y eliminar productos de manera eficiente y precisa. Los objetivos específicos incluyen:

7. Identificar las variables de entrada, salida y procesos asociados en el sistema de gestión de inventario.
8. Formular el problema a resolver por el sistema de gestión de inventario.
9. Desarrollar tablas que describan la formulación del problema y la identificación de variables.

Diagramas de flujo

Opción 1:

https://lucid.app/lucidchart/8c277c79-2a84-4855-93c5-91c4df75e086/edit?viewport_loc=-6478%2C-4903%2C21123%2C11654%2C0_0&invitationId=inv_f9ff715c-033a-4247-964b-77bfcc338103

Opción 2:

https://lucid.app/lucidchart/8c277c79-2a84-4855-93c5-91c4df75e086/edit?viewport_loc=-785%2C-1053%2C5105%2C2816%2C0_0&invitationId=inv_f9ff715c-033a-4247-964b-77bfcc338103

Análisis de propuestas de solución

1. Reutilización de Código:

Al separar las funciones de manejo de inventario en un archivo aparte (funciones.c), el código se vuelve más reutilizable. Estas funciones pueden ser fácilmente importadas y utilizadas en otros programas si es necesario, lo que mejora la modularidad y facilita la mantenibilidad del código.

2. Escalabilidad:

La estructura modular del segundo programa lo hace más escalable. Si se desea agregar nuevas funcionalidades o modificar las existentes, es más fácil hacerlo sin afectar otras partes del código. Además, es más sencillo trabajar en equipo cuando el código está organizado de esta manera.

3. Legibilidad:

La división del código en diferentes archivos mejora la legibilidad del programa, ya que cada archivo se centra en una tarea específica. Esto facilita la identificación de errores y la depuración del código.

4. Facilidad de Mantenimiento:

La estructura modular del segundo programa facilita la identificación y corrección de errores, así como la incorporación de nuevas características. Esto reduce el riesgo de introducir errores en otras partes del código al realizar cambios.

En resumen, mientras que ambos programas cumplen la función de gestionar un inventario, la modularidad, claridad y estructura del segundo programa lo hacen más fácil de entender, mantener y escalar a medida que el proyecto crece. Esto lo hace una opción más robusta y preferible en la mayoría de los casos.

Imágenes de las partes más importantes del código explicadas.

```
#include <stdio.h>
#include <string.h>
#include "funciones.h"
#include "lecturas.h"

#define MaxProducto 1000
#define MaxNombre 60
#define MaxDatos 1000

int main(void) {
    int cantidadProduct;
    char Productos[MaxProducto][MaxDatos];
    int opcion;
    int indice;

    printf("      Bienvenidos a TuTienda\n");
    printf("Cantidad de productos a ingresar (máximo %d): ", MaxProducto);
    scanf("%d", &cantidadProduct);
    getchar();

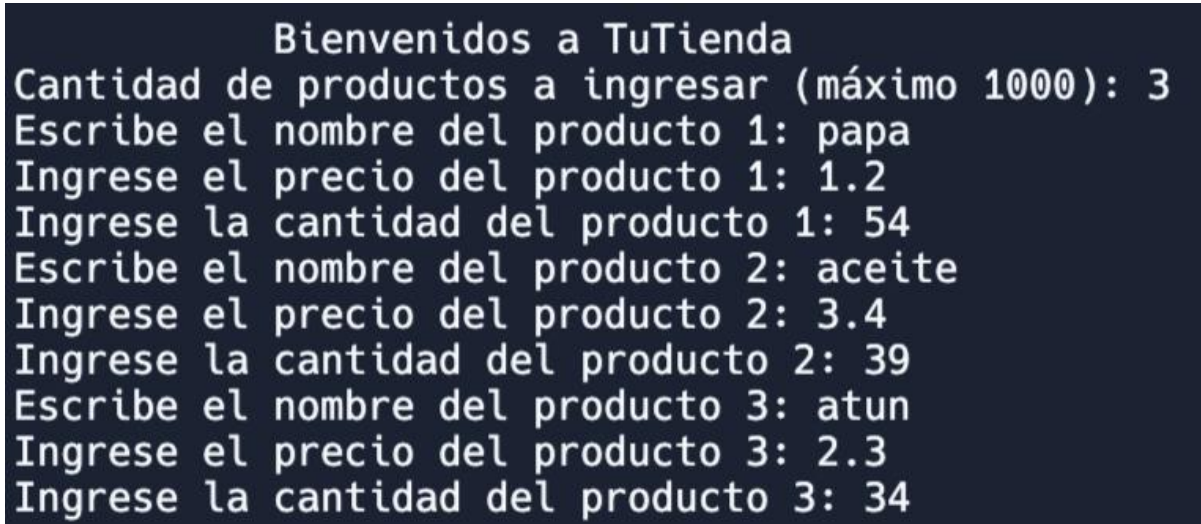
    leerProducto(cantidadProduct, Productos);

    do {
        printf("\nMenu:\n");
        printf("1. Mostrar productos\n");
        printf("2. Editar producto\n");
        printf("3. Eliminar producto\n");
        printf("4. Salir\n\n");
```

El main es la parte más importante de un programa por varias razones fundamentales. En primer lugar, sirve como el punto de entrada del programa, siendo la primera función que se ejecuta cuando el programa se inicia. Desde aquí, se coordina toda la ejecución del programa, interactuando con el usuario al presentar mensajes, solicitar entrada y gestionar acciones. Además, el main es responsable de dirigir la ejecución de otras funciones y módulos del programa, organizando el flujo de operaciones y decidiendo qué funciones se llaman y en qué orden. También alberga la lógica principal del programa, incluyendo cálculos, manipulación de datos, toma de decisiones y cualquier operación central. Además, controla el flujo del programa, decidiendo qué operaciones realizar basadas en condiciones específicas y cuándo finalizar la ejecución del programa. Por último, el main a menudo se encarga de gestionar

errores y excepciones, detectando problemas, informando al usuario y tomando medidas correctivas. En resumen, el `main` es la columna vertebral del programa, coordinando su funcionamiento y asegurando que cumpla su propósito de manera efectiva.

Imágenes de la ejecución de cada sección del programa explicadas.



```
Bienvenidos a TuTienda
Cantidad de productos a ingresar (máximo 1000): 3
Escribe el nombre del producto 1: papa
Ingresa el precio del producto 1: 1.2
Ingresa la cantidad del producto 1: 54
Escribe el nombre del producto 2: aceite
Ingresa el precio del producto 2: 3.4
Ingresa la cantidad del producto 2: 39
Escribe el nombre del producto 3: atun
Ingresa el precio del producto 3: 2.3
Ingresa la cantidad del producto 3: 34
```

En esta primera sección se están leyendo los productos, específicamente tres en este caso.

```
Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir

Seleccione una opción: 1
Producto 1:
Nombre: papa
Precio: 1.20
Cantidad: 54

Producto 2:
Nombre: aceite
Precio: 3.40
Cantidad: 39

Producto 3:
Nombre: atun
Precio: 2.30
Cantidad: 34
```

En esta segunda sección se exhiben los productos introducidos antes.

```
Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir

Seleccione una opción: 2
Ingrese el producto a editar (1-3): 1
Escribe el nuevo nombre del producto 1: azucar
Ingrese el nuevo precio del producto 1: 3.4
Ingrese la nueva cantidad del producto 1: 87

Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir

Seleccione una opción: 1
Producto 1:
Nombre: azucar
Precio: 3.40
Cantidad: 87

Producto 2:
Nombre: aceite
Precio: 3.40
Cantidad: 39

Producto 3:
Nombre: atun
Precio: 2.30
Cantidad: 34
```

En esta tercera sección, observamos el proceso de edición del primer producto, el cual cambia de "papa" a "azúcar".

```
Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir

Seleccione una opción: 3
Ingrese el producto a eliminar (1-3): 3

Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir

Seleccione una opción: 1
Producto 1:
Nombre: azucar
Precio: 3.40
Cantidad: 87

Producto 2:
Nombre: aceite
Precio: 3.40
Cantidad: 39

Menu:
1. Mostrar productos
2. Editar producto
3. Eliminar producto
4. Salir
```

En esta cuarta sección, se muestra cómo eliminar el producto número tres del inventario.

Conclusiones y recomendaciones.

Recomendaciones:

1. Validación de Entrada de Datos:

Implementa una validación más robusta para asegurar que los datos ingresados por el usuario sean correctos y válidos. Por ejemplo, podrías agregar comprobaciones para garantizar que el precio y la cantidad sean valores numéricos positivos.

2. Manejo de Errores:

Agrega mecanismos de manejo de errores para informar al usuario sobre cualquier problema que ocurra durante la ejecución del programa. Esto puede incluir mensajes de error específicos y la posibilidad de volver a intentar la operación.

3. Optimización del Código:

Considera optimizar el código para hacerlo más eficiente en términos de uso de memoria y tiempo de ejecución. Por ejemplo, podrías buscar formas de reducir la cantidad de variables globales y mejorar el rendimiento de las operaciones de copia de cadenas.

4. Interfaz de Usuario Mejorada:

Mejora la interfaz de usuario para hacerla más intuitiva y amigable. Por ejemplo, podrías agregar mensajes de confirmación antes de realizar operaciones de edición o eliminación, y proporcionar instrucciones claras sobre cómo usar el programa.

Conclusiones:

1. Funcionalidad Básica Implementada:

El código proporcionado cumple con la funcionalidad básica de un sistema de gestión de inventario al permitir al usuario ingresar, mostrar, editar y eliminar productos.

2. Estructura Modular y Clara:

La división del código en funciones modulares facilita la comprensión y el mantenimiento del programa. La separación de la lógica de negocio en funciones independientes mejora la organización del código.

3. Áreas de Mejora Identificadas:

Aunque el programa funciona correctamente, existen áreas de mejora identificadas en términos de validación de entrada, manejo de errores, optimización del código y mejora de la interfaz de usuario.

4. Potencial para Expansión y Desarrollo:

El código sirve como una base sólida que puede expandirse y desarrollarse más para satisfacer necesidades específicas o agregar nuevas funcionalidades en el futuro.

En resumen, el código proporcionado establece una base funcional para un sistema de gestión de inventario, pero hay áreas que pueden mejorarse para hacer el programa más robusto y fácil de usar. Con algunas mejoras y ajustes adicionales, este sistema podría ser una herramienta efectiva para administrar inventarios en una variedad de entornos comerciales.