

Universidad Americana (UAM)

Facultad de Ingeniería y Arquitectura

Ingeniería en Sistemas de Información

Sistema de Gestión de Inventario para Gasolinera Puma

Asignatura: Introducción a la Programación

Integrantes:

1. Jimmy Leandro Selva García
2. Erick Antonio Arana Espinoza
3. Jesser Jadiel Rodríguez Chavarría
4. Brandon Javier Rocha Varela

Docente: Silvia Gigdalia Ticay López

Managua, 8 de Julio 2025

Índice

[Introducción 3](#_Toc202873161)

[Objetivos 5](#_Toc202873162)

[Descripción del problema 6](#_Toc202873163)

[Análisis del problema 8](#_Toc202873164)

[Lista de requerimientos 9](#_Toc202873165)

[Recolección de datos 11](#_Toc202873166)

[Diseño del algoritmo 12](#_Toc202873167)

[Codificación. 15](#_Toc202873168)

[Documentación 18](#_Toc202873169)

[Diagrama de estructura 19](#_Toc202873170)

[Conclusión 20](#_Toc202873171)

[Recomendaciones 21](#_Toc202873172)

[Referencias 22](#_Toc202873173)

# Introducción

El presente informe expone el desarrollo de un sistema de gestión de inventario aplicado al contexto operativo de una gasolinera, específicamente orientado al control y organización de productos dentro de su tienda. Este proyecto fue realizado como parte de los requerimientos académicos de la asignatura Introducción a la Programación en Equipo, y tiene como propósito aplicar los fundamentos adquiridos en clase a un caso práctico y funcional. En este documento se describe el proceso completo de elaboración del sistema: desde la identificación del problema y planteamiento de objetivos, hasta la implementación técnica del programa, su estructura modular en Python, y las funcionalidades desarrolladas. Asimismo, se incluye una revisión de los resultados obtenidos, las dificultades enfrentadas durante el trabajo colaborativo, y una propuesta de mejoras para versiones futuras.

El control de inventario representa uno de los procesos más importantes para cualquier negocio que maneje productos físicos, y en el caso particular de una gasolinera, este control es aún más crucial debido a la variedad de artículos como lubricantes, aditivos, aceites y otros productos que deben mantenerse organizados. En este contexto, surge la necesidad de un sistema que permita registrar, actualizar y visualizar en tiempo real toda la información relacionada al stock. Este proyecto propone una solución funcional mediante el desarrollo de un programa en Python que cumple con estos objetivos de forma sencilla, accesible y adaptada a las necesidades reales de una gasolinera.

El sistema de gestión desarrollado funciona desde la consola y permite realizar operaciones fundamentales como el registro de productos, el control de entradas y salidas del inventario, y la consulta del valor total del stock disponible. Además, incluye una opción para el registro de usuarios y el control de accesos, lo que permite garantizar un uso responsable de la herramienta. Todo esto fue implementado de forma modular, dividiendo el código en archivos separados por funcionalidad, lo cual facilita su mantenimiento y mejora su organización interna.

Este proyecto ha sido desarrollado como parte de la asignatura Introducción a la Programación, permitiendo aplicar diversos conceptos técnicos aprendidos durante el curso, como el uso de funciones, estructuras condicionales, ciclos, manejo de archivos, y la organización del código por módulos. A través del proceso de análisis, diseño, codificación y documentación, se logró construir una solución que no solo resuelve un problema técnico, sino que también sirve como experiencia práctica para consolidar habilidades clave en el área de programación, utilizando el lenguaje de programación Python

El alcance de este proyecto trasciende el campo técnico, ya que se encuentra directamente vinculado con el ámbito de los negocios, particularmente en la gestión operativa y administrativa de inventario. Una correcta administración de productos tiene un impacto directo en la eficiencia, rentabilidad y toma de decisiones comerciale**s** dentro de cualquier empresa, en este caso, una gasolinera. Por tanto, el sistema desarrollado no solo resuelve una necesidad tecnológica, sino que también se posiciona como una herramienta clave para mejorar la organización, reducir pérdidas y optimizar los recursos disponibles en un entorno comercial.

Este trabajo busca aportar una herramienta funcional que pueda ser utilizada como base para futuras mejoras, como la integración de una interfaz gráfica o una base de datos más robusta. Al mismo tiempo, demuestra cómo una solución bien pensada y bien implementada puede tener un impacto real en la forma en que se organiza y se gestiona un inventario, especialmente en entornos como el de una gasolinera, donde la eficiencia operativa es clave

# Objetivos

**Objetivo general.**

Desarrollar un sistema básico en Python para gestionar el inventario de productos en una gasolinera Puma.

**Objetivos Específicos.**

Analizar y recolectar los datos necesarios para estructurar adecuadamente la información que manejará el sistema de gestión de inventario.

Implementar el programa de gestión de inventario en Python, utilizando archivos para la persistencia y organización clara de los datos registrados.

Presentar la documentación completa del programa, explicando su funcionamiento de forma clara y comprensible para facilitar su uso y mantenimiento.

# Descripción del problema

En nuestro entorno actual, es muy común ver cómo muchas personas deciden emprender por su cuenta, ya sea vendiendo algún producto desde casa, participando en ferias o abriendo pequeños negocios. Aunque estas iniciativas son una buena forma de generar ingresos, muchas veces se enfrentan a un mismo problema: no tienen una forma ordenada y eficiente de llevar el control de lo que venden ni de lo que tienen en inventario. La mayoría lo hace anotando en una libreta, en hojas sueltas o simplemente confiando en la memoria, lo cual puede causar errores o confusiones.

Por ejemplo, en una feria estudiantil, dond**e** hay poco tiempo para atender a los clientes y muchas cosas que recordar, es fácil equivocarse con los precios, no saber cuántos productos quedan o incluso vender algo que ya no está disponible. Todo esto puede llevar a pérdidas de dinero o a que los clientes se sientan mal atendidos. Además, muchos de estos emprendedores no cuentan con dinero para pagar programas caros, ni con los conocimientos necesarios para usar sistemas complicados.

Pensando en esta situación, se vio la necesidad de crear una herramienta sencilla y fácil de usar que ayude a llevar el control de los productos, los precios, las ventas y hasta poder generar facturas. Así nació la idea de este proyecto: un programa hecho en Python que funciona desde la consola (pantalla negra), sin necesidad de internet ni instalaciones difíciles. Es una solución pensada especialmente para personas que no son expertas en computación, pero que necesitan organizar mejor su negocio.

Este programa permite registrar nuevos productos, ver cuántos hay disponibles, actualizar su información si cambia algo, eliminar productos que ya no se venden, ver el valor total del inventario y también registrar ventas. Al momento de vender, el sistema descuenta automáticamente los productos del inventario y genera una factura. Además, tiene una opción para registrar usuarios y controlar el inicio de sesión, lo cual ayuda a que no cualquier persona pueda modificar los datos.

Con este proyecto no solo se resuelve una necesidad real, sino que también fue una buena forma de poner en práctica todo lo que se ha aprendido en la clase de Introducción a la Programación. Permite aplicar conceptos como el uso de funciones, estructuras condicionales, ciclos, manejo de archivos y organización del código por partes (módulos), todo de una manera clara y útil.

Lo mejor de este tipo de sistemas es que no requiere de grandes recursos para funcionar. Puede usarse desde una computadora sencilla, sin conexión a internet, y está pensado para que cualquier persona con un poco de práctica lo pueda usar. Es una solución que está al alcance de cualquiera que quiera mejorar la forma en que lleva sus cuentas y su inventario.

En resumen, el problema que se quiso resolver fue la falta de un sistema fácil y económico para llevar el control de productos y ventas en negocios pequeños o en ferias. La solución que se propuso cumple con lo necesario: es funcional, práctica y se adapta a lo que muchas personas necesitan. Con esto, no solo se mejora la organización de los emprendedores, sino que también se demuestra que la programación puede servir para solucionar problemas reales de la vida diaria.

# 

# Análisis del problema

Entradas requeridas:  
- Usuario y contraseña para iniciar sesión.  
- Datos de productos: nombre, cantidad, precio, código.  
- Opciones del menú seleccionadas por el usuario que permiten realizar acciones como lo son ver el inventario con los productos ingresados con sus datos esperados, generar un registro del inventario en formato .txt, eliminar y actualizar productos, vender productos y generar su respectiva factura.

Salidas esperadas:

-Ingresar o generar un usuario de ser necesario para acceder al programa  
- Visualización de productos en inventario.  
- Generar factura con los productos ingresados al momento del uso del programa  
- Confirmación de operaciones realizadas (registro, venta, eliminación).  
- Valor total del inventario.

Las salidas esperadas se producen mediante el funcionamiento de diferentes módulos con variables y definiciones que permiten el

Restricciones:  
- El sistema es de consola (sin interfaz gráfica).  
- No utiliza base de datos, solo archivos planos para persistencia.  
- Requiere conocimientos básicos de Python para su mantenimiento.

# Lista de requerimientos

**Requerimientos funcionales:**

Inicio de sesión de usuarios

El sistema debe permitir que un usuario existente inicie sesión mediante su nombre de usuario y contraseña.

Función involucrada: INICIO()

Registro de nuevos usuarios

El sistema debe permitir crear un nuevo usuario con nombre y contraseña.

Función: agregar\_usuario()

Agregar productos al inventario

El sistema debe permitir ingresar nuevos productos al inventario con sus datos básicos (nombre, precio, cantidad, etc.).

Función: AgregarProducto()

Actualizar productos existentes

Permite modificar la información de productos ya registrados (cantidad, precio u otros datos).

Función: ActualizarProducto()

Ver inventario

Muestra la lista de productos actuales con su información y existencia.

Función: VerInventario()

Eliminar productos

Posibilidad de quitar un producto del inventario.

Función: EliminaProducto()

Calcular valor total del inventario

El sistema debe poder mostrar el valor económico total de los productos almacenados.

Función: ValorTotal()

**Requerimientos no funcionales:**

Seguridad básica

Añade una capa de seguridad básica mediante la creación de usuarios y contraseñas para limitar el acceso a usuarios no deseados aumentando la protección del inventario

Interfaz clara por consola:

El menú es textual y fácil de navegar para usuarios con conocimientos básicos de computación.

Modularidad:

El programa está dividido en múltiples módulos (Productos, Factura, Usuario, etc.), lo cual mejora la organización y mantenibilidad del código.

Facilidad de mantenimiento:

Gracias a su estructura modular, el código puede actualizarse fácilmente sin afectar otras funcionalidades.

Rápida ejecución:

Al ser un programa en consola sin interfaz gráfica pesada, el sistema ofrece tiempos de respuesta rápidos.

# Recolección de datos

Para la recolección de datos, utilizamos la herramienta de la observación, de forma que al ponerse de acuerdo en la búsqueda de una necesidad a suplir, nos encontramos con la curiosidad, de que en algunas gasolineras, la mayoría de veces fallan en su sistema de inventariado.

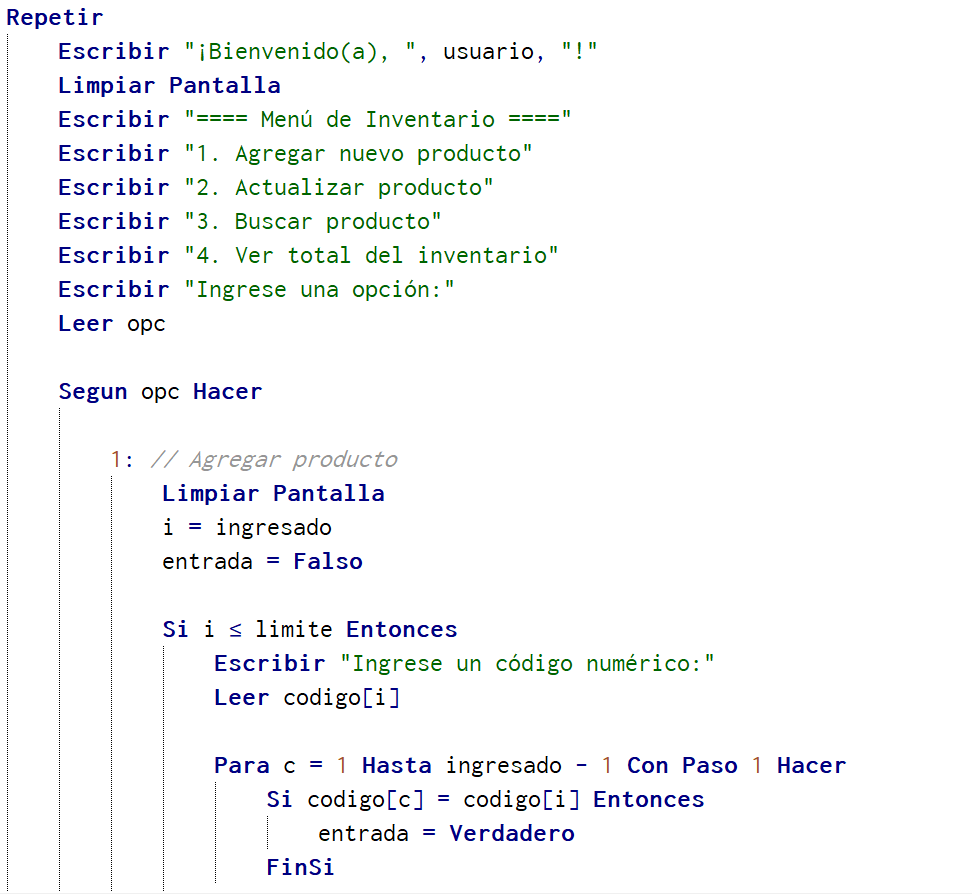
Esto provoca un desorden en su registro del día a día, así que yendo frecuentemente a estos lugares, se pudo ver que tipo de productos se venden, como los registran, el proceso de factura, y todo eso se fue anotando, y sumando para llegar a la versión actual de nuestro programa.

# Diseño del algoritmo

Este algoritmo crea un sistema de inventario básico, el cual, al inicio pide credenciales al usuario, y si no se ingresa la contraseña correcta, se finaliza el proceso. Si se ingresan correctamente los datos, el algoritmo prosigue con su función y nos muestra un menú con 4 opciones las cuales serían agregar, actualizar, buscar, ver total, siguiendo un ciclo de Repetir mientras el usuario así lo decida.

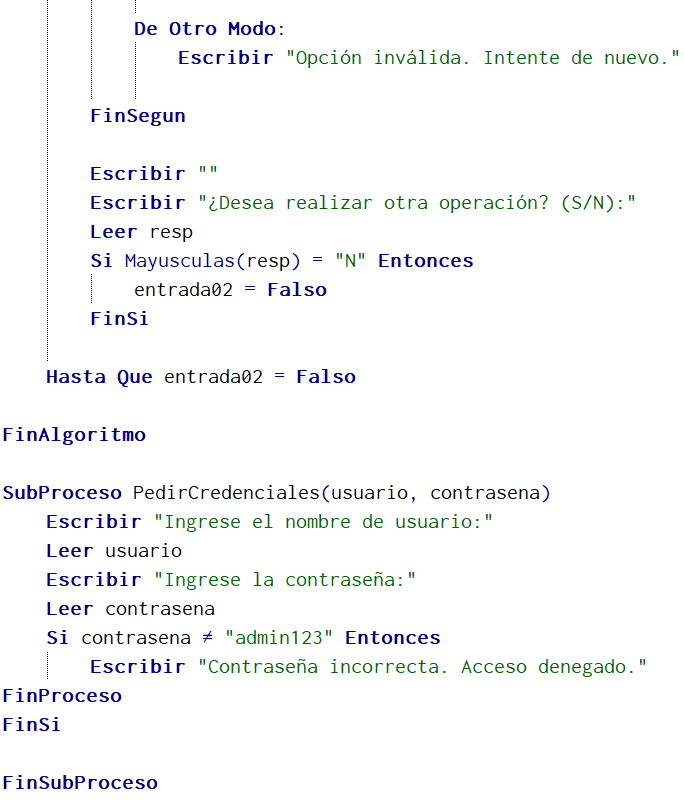
Cada módulo se genera a partir de una estructura de Dimensión, con un rango límite de 1000 productos para todo el inventario, donde no se pueden repetir productos y cada uno debe tener un código único.

Ejemplo de diseño algorítmico (pseudocódigo para menú principal):

****



En la siguiente imagen, podemos ver un bloque de uno de los arreglos correspondientes al menú. Estos responden a los datos de entrada que ingrese el usuario.



Podemos observar acá como termina el segun, y el programa pide al usuario que elija si va a seguir haciendo operaciones con el programas, y después como se muestra en la imagen está el subproceso que pide al usuario sus credenciales para ingresar.

# Codificación.

El archivo Main.py actúa como punto de entrada del sistema. El resto del código está modularizado y documentado con comentarios que explican el propósito de cada función y bloque.

**Fragmento del código principal:**

#Módulo principal

from Productos import AgregarProducto, ActualizarProducto, EliminaProducto, VerInventario, VenderProducto, ValorTotal

from Factura import GenerarFactura

from Usuario import INICIO, agregar\_usuario

from Menu import MostrarMenu

while True:

print("1. Iniciar sesión")

print("2. Registrar nuevo usuario")

print("3. Salir")

opcion = input("Seleccione una opción: ")

if opcion == "1":

if INICIO():

while True:

MostrarMenu()

Opcion = input("Seleccione una opcion: ")

if Opcion == "1":

AgregarProducto()

elif Opcion == "2":

ActualizarProducto()

elif Opcion == "3":

VerInventario()

elif Opcion == "4":

EliminaProducto()

elif Opcion == "5":

ValorTotal()

elif Opcion == "6":

GenerarFactura()

elif Opcion == "7":

VenderProducto()

elif Opcion == "8":

print("-Saliendo del programa-")

break

else:

print("Opción no válida. Por favor, intente de nuevo.")

break

elif opcion == "2":

usuario = input("Nuevo usuario: ")

clave = input("Nueva contraseña: ")

agregar\_usuario(usuario, clave)

print("Usuario registrado correctamente.\n")

elif opcion == "3":

print("Saliendo...")

break

else:

print("Opción no válida.")

Como se puede ver en este fragmento de todo el programa en Main.py se importan desde otros archivos las funciones como lo son las funciones referentes a agregar los productos, nombre, cantidad, precio unitario, la capacidad de poder actualizar la información que ya se había guardado acerca de un producto, poder ver el estado actual del inventario, de liberar espacio borrando productos que ya no sean necesarios para el inventario.

Como es un inventario para una tienda, es necesario poder generar un valor total, junto con la opción de generar facturas, este programa cumple con todas esas funciones, y como extra, para inicializar el programa es necesario introducir tus credenciales de usuario, para mantener un orden y más de una persona pueda usarlo en el mismo dispositivo.

Todo esto se realizó haciendo uso de repetidores, que mientras se cumplan ciertas condiciones, el programa va a realizar una función, por ejemplo, si no se ingresan las credenciales correctas, el programa va a ir acumulando los intentos fallidos, hasta que termina cerrándose automáticamente. En el menú del inventario podemos ver usos de condicionales como if, elif, y uso de break, todo esto, para que funcione en base a los datos que se ingresan, y el usuario pueda decidir cuándo cerrar el programa.

# Documentación

**Documentación interna:**

Se ha utilizado el sistema de comentarios en Python (#) para describir cada función en los módulos.

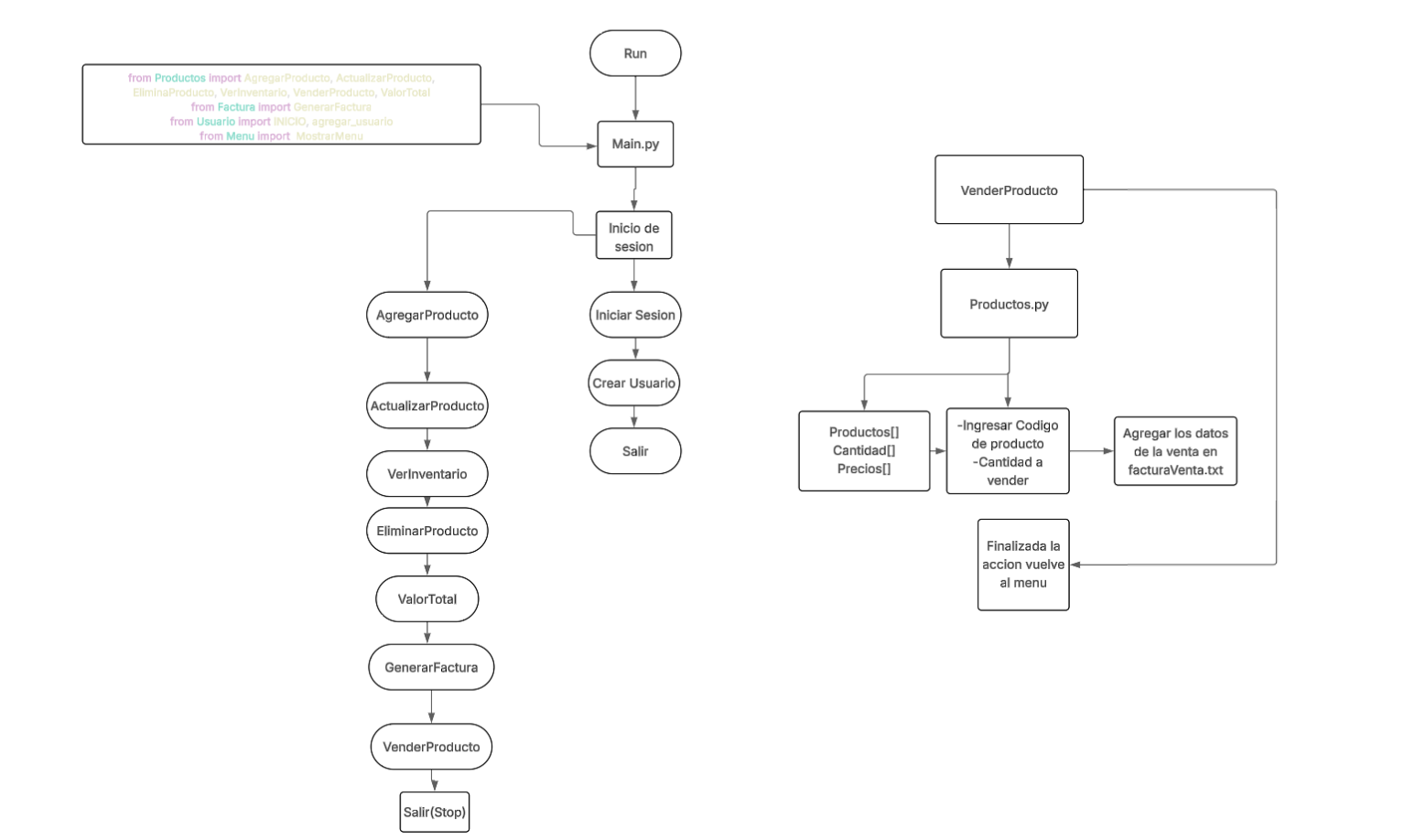
**Documentación externa:**

Este documento sirve como guía teórica del desarrollo y funcionamiento del sistema mediante la explicación de las funciones y variables establecidas en el programa.

**Manual de usuario (resumido)**

1. Ejecutar el archivo Main.py en consola.
2. Elegir “Iniciar sesión” o “Registrar usuario”.
3. Acceder al menú principal.
4. Realizar operaciones de inventario, ventas y facturación según el menú.

# Diagrama de estructura



# Conclusión

El desarrollo de esta aplicación representó una valiosa oportunidad para poner en práctica los conocimientos adquiridos a lo largo del curso, especialmente en lo que respecta a la programación estructurada, el diseño modular mediante funciones, y el manejo de archivos para el almacenamiento persistente de datos. A través de este proyecto, se logró simular un entorno real en el que una gasolinera requiere una solución informática eficiente para gestionar su inventario de productos, tales como lubricantes, aditivos, repuestos y otros artículos de consumo frecuente.

Uno de los principales logros fue la capacidad de diseñar una solución funcional, clara y escalable, capaz de adaptarse a diversas situaciones operativas del negocio. El programa permite agregar, actualizar, eliminar productos, generar facturas y calcular valores totales de venta, todo a través de un menú intuitivo que facilita su uso incluso por personal con

conocimientos técnicos limitados.

Además, su diseño modular no solo mejora la legibilidad y mantenibilidad del código, sino que también lo convierte en una base sólida para futuras mejoras. Entre las posibles extensiones se encuentra la incorporación de una interfaz gráfica de usuario (GUI) que optimice la interacción, así como la integración con bases de datos relacionales que permitan un manejo más robusto y seguro de la información.

En términos de impacto, este tipo de herramienta puede traducirse en una gestión más precisa del inventario, una mejor atención al cliente y una toma de decisiones basada en datos actualizados. De esta manera, el proyecto trasciende el ámbito académico, brindando una solución real a una necesidad operativa concreta.

En resumen, el proyecto no solo cumple con los objetivos propuestos, sino que también demuestra la aplicabilidad del conocimiento técnico en contextos reales, fomentando así una visión profesional y proactiva del desarrollo de software.

# Recomendaciones

* Crear una interfaz gráfica  
   Para que el sistema sea más fácil de usar, se recomienda agregar una interfaz gráfica con ventanas y botones, en lugar de utilizar solo la consola. Esto haría que personas sin conocimientos técnicos puedan manejarlo de forma más cómoda. Herramientas como Tkinter permiten hacer esto en Python de manera sencilla.
* Reemplazar los archivos de texto por una base de datos  
   Actualmente el sistema guarda los datos en archivos .txt. lo cual funciona, pero no es lo más eficiente a largo plazo. Usar una base de datos como SQLite permitiría guardar y organizar mejor la información, hacer búsquedas más rápidas y mantener los datos más seguros.
* Adaptar el sistema para diferentes dispositivos o hacerlo web  
   Sería útil hacer que el programa funcione bien en cualquier sistema operativo (Windows, Linux, etc.). Además, en el futuro se podría pensar en convertirlo en una aplicación web, para que se pueda usar desde cualquier computadora con acceso a internet, sin necesidad de instalar nada.

# Referencias

Joyanes, L. (2008). \*Fundamentos de Programación. Algoritmos, estructura de datos y objetos\* (4ta ed.). McGraw-Hill/Interamericana.

Joyanes, L., & Zahonero, I. (2005). \*Programación en C: Metodología, algoritmos y estructura de datos\* (2da ed.). McGraw-Hill/Interamericana.

Maner, W. (2020, octubre 27). \*Prototipado\*. <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>