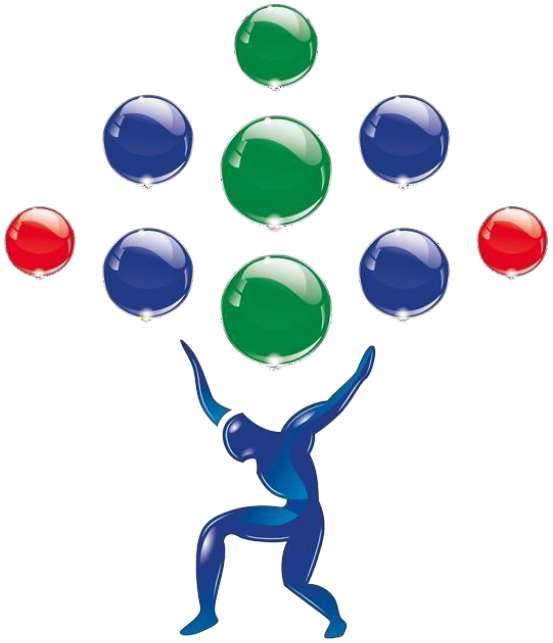


UNIVERSIDAD PRIVADA
DOMINGO SAVIO

SISTEMA DE INVENTARIO DE PRODUCTOS CON PYTHON Y TKINTER

DOCENTE: JIMMY NATANIEL REQUENA LLORENTTY

MATERIA: PROGRAMACIÓN II

CARRERA: ING. REDES Y TELECOMUNICACIONES –
INGENIERÍA DE SISTEMAS

GRUPO: DELAY TODAY

ESTUDIANTES: JHOEL IVAN MACIAS MAMANI
FLAVIO CESAR ROJAS VARGAS

INDICE

INTRODUCCIÓN

En la era digital en la que vivimos, saber gestionar los recursos de forma eficiente se ha vuelto clave para el éxito de cualquier emprendimiento, sin importar su tamaño. Uno de los aspectos más importantes en este sentido es el control de inventarios, especialmente en sectores como el alimenticio, donde factores como la frescura y la disponibilidad de productos pueden marcar una gran diferencia.

Pensando en esta necesidad, este proyecto propone el desarrollo de una aplicación de escritorio que permita llevar un control sencillo, práctico y funcional de un inventario de frutas. Para su implementación se utilizó el lenguaje de programación Python junto con la biblioteca Tkinter, la cual permite crear interfaces gráficas de forma accesible para el usuario.

La aplicación está diseñada para que cualquier persona pueda registrar frutas, editar sus cantidades, eliminarlas o reducirlas conforme se consumen, todo de manera intuitiva. Además, los datos se guardan automáticamente en un archivo JSON, asegurando que la información no se pierda al cerrar el programa.

Más allá del desarrollo técnico, este proyecto representa una oportunidad para aplicar conocimientos en programación, fortalecer habilidades en resolución de problemas y trabajar con orden y precisión. La gestión de inventarios es una necesidad real para muchos negocios, y esta herramienta, aunque sencilla, puede servir como una solución práctica para quienes buscan mejorar su organización.

JUSTIFICACIÓN

¿Por qué se va a realizar este proyecto?

La gestión de inventarios es un pilar fundamental para el buen funcionamiento de cualquier negocio, especialmente en el sector alimentario, donde la frescura y disponibilidad de productos son clave para satisfacer a los clientes. En muchos casos, pequeños comerciantes como fruteros o mercados locales aún gestionan sus inventarios de forma manual, lo que puede dar lugar a errores, pérdidas económicas y una atención al cliente deficiente.

Este proyecto se plantea como una solución concreta a esas problemáticas, brindando una herramienta digital que facilite el control de inventarios. De esta forma, los comerciantes podrán enfocarse en mejorar su servicio y hacer crecer su negocio, contando con el respaldo de una aplicación práctica y fácil de usar.

¿Para qué se va a realizar este proyecto?

El principal propósito de este proyecto es desarrollar una aplicación de escritorio que permita gestionar un inventario de frutas de manera eficiente. Los objetivos específicos incluyen:

- Facilitar el control de inventarios: Ofrecer una herramienta intuitiva que permita agregar, editar, eliminar y disminuir la cantidad de frutas con rapidez y sin complicaciones.
- Aumentar la eficiencia: Reducir el tiempo y los errores asociados a la gestión manual, mejorando la precisión en el seguimiento de existencias.
- Apoyar la sostenibilidad local: Brindar soporte tecnológico a pequeños comerciantes, contribuyendo al fortalecimiento de la economía local y promoviendo el consumo de productos frescos.

- Fomentar el desarrollo técnico: Permitir que estudiantes y desarrolladores apliquen sus conocimientos en programación y diseño de software en un proyecto real, promoviendo el aprendizaje activo, la resolución de problemas y el trabajo colaborativo.

¿Cómo se va a realizar este proyecto?

El proyecto se desarrollará en distintas etapas para asegurar una solución funcional y adaptada a las necesidades de los usuarios:

1. **Investigación y análisis:**

Se realizará un estudio de las necesidades de los posibles usuarios, mediante entrevistas, encuestas u observación, para comprender los retos actuales en la gestión de inventarios.

2. **Diseño de la aplicación:**

Se diseñará una interfaz gráfica amigable utilizando la biblioteca Tkinter de Python, priorizando la accesibilidad y la facilidad de uso para personas con distintos niveles de experiencia tecnológica.

3. **Desarrollo del software:**

Se implementarán funciones como agregar, editar, eliminar y disminuir frutas en el inventario. También se incorporará un sistema de almacenamiento persistente utilizando archivos JSON, que permitirá guardar la información de forma segura.

4. **Pruebas y validación:**

La aplicación será probada rigurosamente para asegurar su correcto funcionamiento. Se invitará a usuarios reales a probar la herramienta y proporcionar comentarios útiles para su mejora.

Prueba con assert

```python
def agregar_fruta_logica(dic, fruta, cantidad):
    fruta = fruta.strip().capitalize()
    if not fruta:
        raise ValueError("Nombre de fruta vacío")
    if cantidad < 0:
        raise ValueError("Cantidad negativa")
    dic[fruta] = dic.get(fruta, 0) + cantidad
    return dic

def editar_fruta_logica(dic, fruta, nueva_cantidad):
    fruta = fruta.strip().capitalize()
    if fruta not in dic:
        raise KeyError("Fruta no encontrada")
    if nueva_cantidad < 0:
        raise ValueError("Cantidad negativa")
    dic[fruta] = nueva_cantidad
    return dic

def eliminar_fruta_logica(dic, fruta):
    fruta = fruta.strip().capitalize()
    if fruta in dic:
        del dic[fruta]
    return dic

def disminuir_fruta_logica(dic, fruta, cantidad):
    fruta = fruta.strip().capitalize()
    if fruta not in dic:
        raise KeyError("Fruta no encontrada")
    if cantidad < 0:
        raise ValueError("Cantidad negativa")
    if cantidad > dic[fruta]:
        raise ValueError("Cantidad mayor a la disponible")
    dic[fruta] -= cantidad
    return dic
```

Estas funciones manejan directamente los datos del inventario. Permiten agregar, editar, eliminar y disminuir frutas usando un diccionario. Se pueden usar tanto en pruebas como dentro de una interfaz gráfica (Tkinter).

```
    try:
        agregar_fruta_logica(inv, "", 5)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por nombre vacío"

    try:
        disminuir_fruta_logica(inv, "naranja", 2)
    except KeyError:
        pass
    else:
        assert False, "Debe lanzar KeyError por fruta no registrada"

    try:
        agregar_fruta_logica(inv, "pera", -3)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por cantidad negativa"

    try:
        disminuir_fruta_logica({"Pera": 1}, "pera", 5)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por cantidad mayor a disponible"

    print("✅ Todas las pruebas pasaron correctamente.")

# Ejecutar las pruebas
if __name__ == "__main__":
    pruebas_inventario()
```

Esta función ejecuta una serie de pruebas automáticas para comprobar que todas las funciones del inventario funcionan correctamente en distintos casos. Usa **'assert'** para verificar resultados esperados y capturar errores.

```
    try:
        agregar_fruta_logica(inv, "", 5)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por nombre vacío"

    try:
        disminuir_fruta_logica(inv, "naranja", 2)
    except KeyError:
        pass
    else:
        assert False, "Debe lanzar KeyError por fruta no registrada"

    try:
        agregar_fruta_logica(inv, "pera", -3)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por cantidad negativa"

    try:
        disminuir_fruta_logica({"Pera": 1}, "pera", 5)
    except ValueError:
        pass
    else:
        assert False, "Debe lanzar ValueError por cantidad mayor a disponible"

    print("✅ Todas las pruebas pasaron correctamente.")

# Ejecutar las pruebas
if __name__ == "__main__":
    pruebas_inventario()
```

Esta parte ejecuta las pruebas automáticamente si el archivo se corre directamente. Es útil para que las pruebas no se ejecuten si este módulo se importa desde otro.

OBJETIVOS

Objetivo General

- Desarrollar un código de inventarios de frutas en Python.

Objetivos Específicos

- Diseñar una interfaz gráfica amigable e intuitiva

- Implementar funciones básicas de gestión de inventario.
- Integrar un sistema de almacenamiento persistente.

MARCO CONCEPTUAL

Inventario

tangibles que posteriormente serán comercializados. Se trata de uno de los rubros que requieren especial interés de los administradores de las compañías dedicadas a la comercialización o producción, ya que en este tipo de empresas los inventarios forman parte de la administración integral de recursos debido a su importancia en la planeación y control de las actividades de negocio que lleva a obtener una rentabilidad adecuada.

Sistema de gestión de inventario

Los sistemas de gestión de inventario son herramientas informáticas que permiten llevar un control preciso de los productos almacenados, facilitando el seguimiento de existencias en tiempo real. Estos sistemas automatizan muchas tareas que antes se realizaban manualmente, como el registro de nuevas existencias, la actualización de cantidades y la generación de informes. Esto no solo mejora la precisión, sino que también ahorra tiempo y reduce el riesgo de errores, permitiendo una mejor toma de decisiones y la optimización de los recursos. **Python** a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses

"Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

Tkinter

Tkinter es una biblioteca de Python que permite la creación de interfaces gráficas de usuario (GUI). Es una herramienta sencilla y accesible para crear aplicaciones de escritorio con ventanas, botones, cuadros de texto y otros elementos gráficos. Tkinter forma parte de la biblioteca estándar de Python, lo que significa que no requiere

instalaciones adicionales para su uso. Es ideal para aplicaciones que no necesitan interfaces complejas, pero que requieren interacciones simples y efectivas con el usuario.

JSON (JavaScript Object Notation)

JSON es un formato de intercambio de datos ligero, basado en texto, que es fácil de leer y escribir tanto para humanos como para máquinas. Es utilizado ampliamente en aplicaciones web y sistemas de almacenamiento, ya que permite representar datos de manera estructurada (como objetos y arrays). En el contexto de la programación, JSON se utiliza comúnmente para almacenar configuraciones, bases de datos ligeras o cualquier tipo de información que necesite persistir entre ejecuciones de una aplicación. En este proyecto, JSON se usa para guardar el inventario de frutas, asegurando que los datos no se pierdan cuando se cierre la aplicación.

Validación de datos

La validación de datos es el proceso de comprobar que los datos introducidos por un usuario sean correctos y cumplan con ciertos criterios antes de ser procesados por un sistema. Este proceso es fundamental para evitar errores o información incorrecta que pueda afectar el funcionamiento de la aplicación o el sistema en general. En la gestión de inventarios, por ejemplo, se debe validar que las cantidades ingresadas sean números y que los campos de texto no estén vacíos, garantizando así la precisión y coherencia de los datos almacenados.

Persistencia de datos

La persistencia de datos se refiere a la capacidad de un sistema para guardar información de manera que no se pierda cuando la aplicación o el sistema se cierre. A diferencia de los datos que se almacenan temporalmente en la memoria del sistema, los datos persistentes se guardan en archivos o bases de datos, permitiendo que sean

accesibles incluso después de reiniciar el sistema. En este proyecto, la persistencia de datos se logra mediante el uso de archivos JSON, lo que permite que los datos del inventario se conserven entre diferentes sesiones de la aplicación.

Programación

La programación es el proceso de crear un conjunto de instrucciones que una computadora puede entender y ejecutar. A través de la programación, los desarrolladores crean software que permite que las máquinas realicen tareas específicas de acuerdo con las necesidades de los usuarios. Existen diversos lenguajes de programación, cada uno con sus propias características y aplicaciones, como Python, Java, C++, entre otros. La programación no solo involucra escribir código, sino también planificar y diseñar algoritmos eficientes para resolver problemas. En el caso de este proyecto, la programación en Python permite desarrollar la aplicación de gestión de inventarios, implementando las funcionalidades necesarias de manera estructurada y lógica para garantizar que la aplicación funcione de forma eficiente y sin errores.

DESARROLLO

1 CONFIGURACIÓN INICIAL

```
ARCHIVO_DATOS = "inventario.json"
inventario = {}
```

- Se define el archivo donde se guardará el inventario (inventario.json).
- Se inicializa el diccionario inventario vacío.

2 FUNCIONES DE PERSISTENCIA

```
def guardar_inventario():
```

- guardar_inventario: Guarda el inventario en formato JSON, ordenado alfabéticamente.

```
def cargar_inventario():
```

- cargar_inventario: Carga el archivo JSON si existe, o crea un inventario vacío si no lo encuentra.

3 FUNCIONES DE GESTIÓN

```
def agregar_fruta():
```

- Obtiene el nombre y cantidad desde las entradas.
- Si la fruta ya existe, suma la cantidad.
- Guarda, actualiza la lista y limpia los campos.

```
def eliminar_fruta():
```

- Elimina la fruta si existe en el inventario.
- Guarda, actualiza la lista y limpia.

```
def editar_fruta():
```

- Cambia la cantidad de una fruta ya registrada.

```
def disminuir_fruta():
```

- Resta una cantidad a la fruta indicada, si existe y la cantidad es válida.
- Elimina automáticamente si la cantidad llega a 0.

```
def buscar_fruta(*args):
```

- Muestra solo las frutas cuyo nombre coincide con la búsqueda.

```
def seleccionar_fruta(event):
```

- Al hacer clic en una fruta de la lista, rellena los campos de entrada con su nombre y cantidad.

```
def actualizar_lista():
```

- Muestra todo el inventario ordenado en el Listbox.

```
def limpiar_campos():
```

- Borra las entradas de texto.

4 MODO OSCURO / CLARO

```
def toggle_modo_oscuro():
```

- Cambia los colores de fondo, texto y botones según se active el modo_oscuro (variable booleana).
- Reconfigura los estilos visuales de widgets (TLabel, TEntry, Listbox, etc.).

5 INICIALIZACIÓN

```
cargar_inventario()
```

- Se carga el inventario guardado previamente (si existe).

6 CREACIÓN DE VENTANA PRINCIPAL

```
ventana = tk.Tk()
```

- Se crea y configura la ventana principal con tamaño, título y colores base.

7 ESTILOS VISUALES

```
estilo = ttk.Style()
```

- Se definen los estilos para etiquetas (TLabel), entradas (TEntry) y botones (TButton).

8 SECCIÓN DE ENTRADA DE DATOS

```
frame_entrada = ttk.Frame(ventana)
```

- Contiene las entradas para ingresar el nombre y cantidad de frutas.

9 BOTONES FUNCIONALES

```
frame_botones = ttk.Frame(ventana)
```

- Cuatro botones: **Agregar, Editar, Eliminar y Disminuir**, cada uno con color personalizado y su respectiva función.

10 BÚSQUEDA

```
frame_busqueda = ttk.Frame(ventana)
```

- Entrada que filtra frutas mientras se escribe.

11 INVENTARIO VISUAL (LISTBOX)

```
lista_frutas = tk.Listbox(ventana, width=50, height=15, font=("Courier New",
10),
```

- Muestra todas las frutas y cantidades. Permite seleccionar una fruta para editarla.

12 MODO OSCURO

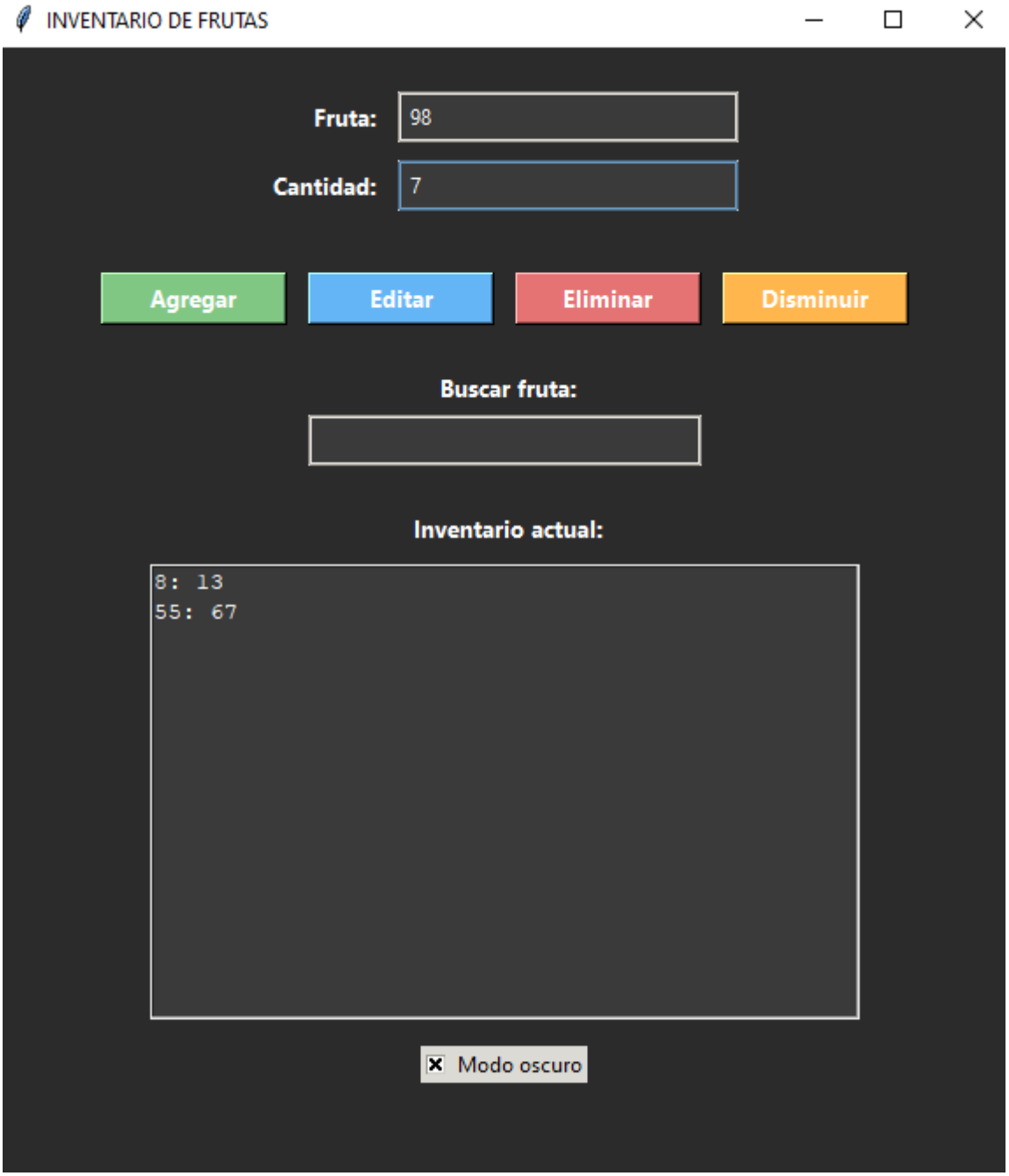
```
chk_oscuro = ttk.Checkbutton(ventana, text=" Modo oscuro", variable=modo_oscuro,
command=toggle_modo_oscuro)
```

- Un checkbox activa o desactiva el modo oscuro llamando a toggle_modo_oscuro.

13 EJECUTAR INTERFAZ

```
actualizar_lista()
ventana.mainloop()
```

- Se muestra la lista actual y se lanza el bucle principal para mantener la ventana abierta.


INVENTARIO DE FRUTAS
Fruta:
98
Cantidad:
7
Agregar
Editar
Eliminar
Disminuir
Buscar fruta:
Inventario actual:
8: 13
55: 67
Modo oscuro

ANEXO

- **Generación del ejecutable (.exe) con PyInstaller**

Gracias a ChatGPT se utilizó el siguiente comando para convertir el programa Python en un archivo ejecutable (.exe), útil para compartirlo sin necesidad de tener Python instalado:

pyinstaller --onefile --windowed inventario_frutas.py

sugerimos a la ia para que nos genere los pasos para instalar PyInstaller con:

pip install pyinstaller

- **Personalización del diseño con colores y temas**

ChatGPT ayudó a implementar una interfaz amigable mediante:

Colores cálidos en modo claro: fondo celeste claro, botones verdes, azules y naranjas suaves.

Modo oscuro con tonos grises oscuros y texto blanco para mejor visibilidad. Esto se logró creando un diccionario llamado colores, por ejemplo:

```
colores = {
    "bg": "#d0ebff",
    "fg": "#000000",
    "entry_bg": "#e7f5ff",
    "list_bg": "#e7f5ff"
}
```

- **Estructura de ventanas y componentes gráficos (Tkinter)**

El programa se organizó de forma clara gracias a la guía de ChatGPT en los siguientes aspectos:

Ventana principal (`tk.Tk`) con título y tamaño fijo.

Entradas (`ttk.Entry`) para fruta y cantidad.

Botones (`tk.Button`) con colores personalizados y comandos separados: agregar, editar, eliminar, disminuir.

Listbox para mostrar el inventario actual, con selección de frutas.

Campo de búsqueda con filtrado dinámico al escribir.

Checkbutton para activar o desactivar el modo oscuro.

Cada componente fue creado siguiendo sugerencias de estilo moderno, legibilidad y buena organización mediante `ttk.Frame`.

El uso de ChatGPT no reemplazó el trabajo del programador, sino que aceleró el desarrollo, ofreció ideas claras y ayudó a entender mejor la programación en Python con interfaz gráfica. Sirvió como asistente técnico, explicando conceptos, sugiriendo soluciones y facilitando el aprendizaje continuo. (OpenAI, 2025)

CONCLUSIÓN

Gracias a este proyecto, hemos podido aplicar todo lo aprendido con respecto a la programación 2, y es una prueba que una idea puede llegar a ser algo funcional, sencillo de manejar y que a su vez ayuda en la vida diaria de los micros, pequeños y medianos comerciantes. Lo se parece un simple inventario es la base de poder operar eficientemente, y con esto, demostramos que podemos ofrecer una herramienta simple pero poderosa a la vez para ayudar a muchos.

RECOMENDACIÓN

Se recomienda ampliar las funcionalidades del sistema incorporando herramientas como búsqueda rápida, alertas de stock bajo y generación de reportes. También sería beneficioso mejorar la interfaz gráfica para hacerla más moderna y fácil de usar, y considerar el uso de una base de datos en lugar de archivos JSON para un manejo más robusto de la información. Finalmente, esta aplicación puede extenderse a otros tipos de negocios, lo que la convierte en una base útil para futuros proyectos más completos.

Link GitHub del código del proyecto:

https://github.com/Jhoel-MM/proyecto_prog2

link presentación del proyecto:

https://www.canva.com/design/DAGrxOB2kHk/t5YNOFtx3qE43J-dGlu-lQ/edit?utm_content=DAGrxOB2kHk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

BIBLIOGRAFÍA

distancias, F. d. (24 de 06 de 2025). *CONCEPTOS BÁSICOS DE LOS INVENTARIOS*. Obtenido de http://virtual.umng.edu.co/distancia/ecosistema/ovas/administracion_empresas/contabilidad_general/unidad_4/DM.pdf

Duque, R. G. (25 de 06 de 2025). *Python para todos*. Obtenido de https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf

González, E. B. (20 de 06 de 2018). *Sistema para la planificación y control de inventarios*. Obtenido de https://repositorio.uci.cu/jspui/bitstream/123456789/9927/1/TD_09238_18.pdf

SafetyCulture. (25 de 06 de 2025). *SafetyCulture*. Obtenido de https://safetyculture.com/es/temas/manejo-de-inventario/control-de-inventarios/

OpenAI. (01 de 07 de 2025). *ChatGPT*. Obtenido de OpenAI: https://chat.openai.com/