**Enunciado del ejercicio: “Gestión de Inventario con Lista Enlazada y Tkinter”**

**Objetivo**

Diseñar e implementar una aplicación de escritorio en Python que permita gestionar un inventario de productos usando:

1. Una **estructura de datos dinámica** (lista enlazada) para almacenar y manipular los ítems.
2. Una **interfaz gráfica** construida con **Tkinter** para interactuar con el usuario.

**Descripción del problema**

Se quiere una herramienta sencilla para registrar, actualizar y eliminar productos de un inventario, sin recurrir a colecciones integradas (listas o diccionarios) de Python. En su lugar, deberás construir tu propia lista enlazada.

**Requisitos funcionales**

1. **Modelo de dominio**
   * Crear la clase Producto con atributos:
     + id (entero)
     + nombre (cadena)
     + precio (flotante)
     + cantidad (entero)
   * Implementar un método \_\_str\_\_ para representar un Producto.
2. **Estructura de datos**
   * Definir la clase Nodo que almacene un Producto y apunte al siguiente nodo.
   * Implementar la clase ListaEnlazada con los métodos:
     + agregar(dato): inserta al final.
     + eliminar\_por\_id(id): quita el nodo cuyo Producto.id coincida.
     + buscar\_por\_id(id): devuelve el Producto o None.
     + listar(): devuelve un listado de todos los Producto existentes.
3. **Lógica de negocio**
   * Crear la clase Inventario que encapsule una instancia de ListaEnlazada.
   * Métodos CRUD delegados a la lista:
     + agregar\_producto(producto)
     + eliminar\_producto(id)
     + actualizar\_producto(id, nombre, precio, cantidad)
     + buscar\_producto(id)
     + listar\_productos()
4. **Interfaz gráfica (Tkinter + ttk)**
   * Ventana principal de tamaño aproximado 800×600 px.
   * **Panel de datos** con campos Entry para ID, Nombre, Precio y Cantidad, y botones:
     + **Agregar**, **Actualizar**, **Eliminar**, **Buscar** y **Limpiar**.
   * **Panel de lista** con un Treeview que muestre las columnas: ID, Nombre, Precio y Cantidad.
   * Barra de desplazamiento vertical para el Treeview.
   * Al seleccionar una fila: cargar los datos en los Entry.
   * Todas las operaciones deben mostrar mensajes de éxito o error con messagebox.

**Entregables**

1. Archivo Python .py con todo el código.