

Presentación Final: Sistema de Gestión de Inventario

Módulo 3: Fundamentos de Programación en Python

1. Introducción y Propósito

Este proyecto surge de la necesidad de automatizar procesos internos de gestión de activos. El sistema desarrollado permite administrar un catálogo de productos, controlar existencias y generar reportes financieros básicos de manera automatizada, eliminando el error humano en cálculos manuales.

2. Tecnologías y Herramientas

Para garantizar la eficiencia y escalabilidad, se utilizó el siguiente stack:

- **Lenguaje:** Python 3.x (Estructuras de datos nativas).
 - **Entorno de Desarrollo:** Visual Studio Code.
 - **Control de Versiones:** Git (Local) y GitHub (Remoto).
 - **Formato de Intercambio:** CSV (Comma Separated Values) para pruebas de datos.
-

3. Arquitectura del Sistema (Modularidad)

Siguiendo las mejores prácticas, el código no es un bloque plano, sino que está dividido en funciones con responsabilidades únicas:

- `registrar_producto()`: Maneja la entrada de datos y limpieza.
 - `ver_inventario()`: Procesa la salida visual usando **f-strings**.
 - `calcular_totales()`: Realiza la lógica matemática sobre las estructuras.
-

4. Estructuras de Datos (El Corazón del Código)

Se seleccionaron las estructuras según su eficiencia:

1. **Diccionarios:** Ideales para representar "objetos" (Producto -> Precio).
 2. **Listas:** Utilizadas para el almacenamiento dinámico y ordenado del inventario.
 3. **Sets:** Implementados en las validaciones para asegurar que no existan nombres de productos duplicados.
-

5. Validación y Control de Flujo

El sistema es robusto frente a errores de usuario:

- **Ciclos `while`:** Mantienen el sistema activo en un menú interactivo.
 - **Sentencias `if-elif-else`:** Dirigen el flujo según la opción elegida.
 - **Bloques `try-except`:** Evitan que el programa colapse si el usuario ingresa texto en lugar de números (precios/stock).
-

6. Conclusión

El proyecto demuestra una base sólida en lógica de programación y manejo de datos.