

Módulo 4 - Aprendizaje esperado 6 - Ejercicio grupal

Nombres: Cristian Aranda - Matías Lagos - Catalina Villegas

Cristian:

A modo personal, este ejercicio principalmente me sirvió para aprender la manipulación de archivos con sus distintos modos de abrir archivos ("r", "w", "a" y "r+" principalmente). Veo este aprendizaje como algo crucial para mi formación a futuro, para poder generar programas completos. Al utilizar archivos externos al script, se siente más cercano lo que podría generar con un programa a futuro a través de la programación en Python.

Además, el ejercicio me ayudó a entender de mejor forma el manejo de excepciones de forma aplicada y la utilización de mejor forma de funciones como strip(), split() y la aplicación de bucles para procesar la información que se puede recibir desde un archivo de texto y poder utilizarla en otras funciones del programa. Con un ejercicio de esta naturaleza, en el cual se está utilizando un archivo de texto como inventario, el cual puede ser tratado como una lista, se me hizo más simple el poder visualizar una aplicación real del código. Si bien aún me cuesta el poder abstraer los requerimientos y la forma en la cual aplicar el código, siento que voy mejorando cada vez más.

Catalina:

El objetivo de este ejercicio era reemplazar la gestión manual en hojas de cálculo por un programa que leyera y escribiera los datos directamente en un archivo de texto llamado "inventario.txt". Bajo esta perspectiva, considero que este ejercicio me ayudó a comprender de manera práctica, más allá de lo teórico, los diferentes modos de apertura de archivos en Python ("r", "w", "a", "r+"), realizando la lectura, escritura y modificación de datos estructurados en texto plano.

El código está organizado con funciones para leer_inventario, registrar_producto, agregar_producto, modificar_producto, modificar_nombre, eliminar_producto, leer_producto y obtener_atributos a través de las bibliotecas os y time. Cada método usa la estructura with open (...) para abrir y cerrar los archivos automáticamente, además de incluir el manejo de excepciones a través de try/except, lo que evita que el programa se interrumpa si el archivo no existe o hay un error inesperado. Todo esto siendo llamado a través del menú interactivo que permite la interacción con el usuario a través de la consola.

En síntesis, este trabajo me ayudó a entender de manera práctica cómo trabajar de manera correcta en la manipulación de archivos a través del intercambio de puntos de vista y opiniones con mis compañeros de equipo.

Matías:

Fui responsable de la función consultar_producto(), que busca un artículo específico en el inventario por su nombre o ID. También desarrollé la lógica para modificar_producto(), que actualiza los datos de un artículo, y eliminar_producto(), que remueve un producto del inventario. Ambas funciones leían todo el inventario, realizaban la operación necesaria en memoria y luego reescribían el archivo de texto.

Entendí que para modificar o eliminar información en un archivo, primero es necesario cargar todos los datos a la memoria del programa. Allí, los datos se manipulan en estructuras como listas y diccionarios antes de reescribir completamente el archivo con la información actualizada. Para lograr esto, utilicé los modos de apertura de archivos de Python, específicamente el modo de lectura ('r') para obtener los datos y el modo de escritura ('w') para guardar los cambios. Además, apliqué algoritmos de búsqueda con bucles for para iterar sobre los productos y encontrar un elemento específico por su nombre o ID.