Proyecto Final – Sistema de gestión de bibliotecas

El objetivo de este proyecto es diseñar e implementar en Python un **Sistema de Gestión de Bibliotecas (SGB)**. Este sistema debe manejar diversas estructuras de datos y la aplicación de algoritmos para clasificar, buscar, y resolver problemas de asignación de recursos. Con la elaboración de este proyecto ustedes deben demostrar la comprensión de **todos los temas** vistos en la actividad académica de **"Técnicas de Programación"**.

El proyecto base tiene un valor del **100% de la nota**. La sustentación no otorga puntos adicionales, sino que **valida** que ustedes realizaron el proyecto, permitiendo **mantener la nota obtenida en la revisión de código y cumplimiento de requerimientos**. Si la pareja o estudiante no logra resolver las modificaciones solicitadas en la sustentación o no demuestran pleno entendimiento del código, la nota de este será de cero. El proyecto debe implementar la siguiente lógica de manera completamente funcional:

Adquisición y Estructuras de Datos

1. **Adquisición de Datos:** El sistema debe cargar su inventario inicial leyendo un archivo (CSV o JSON) que contiene **al menos cinco atributos** por libro: ISBN, Título, Autor, Peso (en Kg), y Valor (en pesos colombianos). Si consideran necesario agregar más atributos a este archivo inicial pueden hacerlo.

2. **Manejo de Listas:** Se deben mantener dos listas maestras de objetos Libro: el **Inventario General** (una lista desordenada, reflejando el orden de carga) y el **Inventario Ordenado** (una lista siempre mantenida en orden ascendente por ISBN).

3. **Pilas (Historial):** Implementar la gestión del **Historial de Préstamos** por usuario como una Pila (LIFO). Al prestar un libro, se apilan el ISBN y la fecha de préstamo. (El historial debe ser almacenado en un archivo y puede ser cargado posteriormente)

4. **Colas (Reservas):** Implementar la **Lista de Espera** para libros agotados como una Cola (FIFO). Solo se puede encolar un usuario para reserva si el libro tiene stock cero. (Esta solicitud de reservas también deben ser almacenadas en un archivo que puede ser cargado posteriormente)

Algoritmos de Ordenamiento

1. **Ordenamiento por Inserción:** Este algoritmo debe usarse para mantener el **Inventario Ordenado** cada vez que se **agrega un nuevo libro** al sistema. Esto asegura que la lista para la Búsqueda Binaria esté siempre lista.

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

2. **Ordenamiento por Mezcla (Merge Sort):** Este algoritmo debe usarse para generar un **Reporte Global** de inventario, ordenado por el atributo **Valor (COP)**. El reporte generado también debe poder almacenarse en un archivo.

Algoritmos de Búsqueda

1. **Búsqueda Lineal:** Implementar la búsqueda por **Título o Autor** sobre el **Inventario General** (la lista desordenada).

2. **Búsqueda Binaria (Crítica):** Implementar la búsqueda por **ISBN** sobre el **Inventario Ordenado**. **Esta función es crítica**; su resultado (posición o no encontrado) debe ser utilizado obligatoriamente para **verificar si un libro devuelto tiene reservas pendientes** en la Cola de Espera. Si esto es así debe asignarse a la persona que ha solicitado la reserva según la prioridad.

Módulo de estantería: Algoritmos de Resolución de Problemas (Fuerza Bruta y Backtracking)

Se debe ofrecer un módulo de estantería donde se permite asignar la ubicación de cada uno de los libros en diferentes estanterías disponibles.

1. **Fuerza Bruta (Estantería Deficiente):** Implementar un algoritmo que encuentre y liste **todas las combinaciones posibles de cuatro libros** que, al sumar su peso en Kg, superen un umbral de "riesgo" de 8 Kg (Que es lo máximo que soporta un estante de libros). El algoritmo debe explorar exhaustivamente todas las combinaciones.

2. **Backtracking (Estantería Óptima):** Implementar un algoritmo que encuentre la combinación de libros que **maximice el valor total (COP)** sin exceder la capacidad máxima de peso (8 Kg) de un estante. El algoritmo debe demostrar la exploración y su ejecución.

Recursión (Pila y Cola)

1. **Recursión de Pila:** Implementar una función recursiva que calcule el **Valor Total** de todos los libros de un autor específico.

2. **Recursión de Cola:** Implementar una función recursiva que calcule el **Peso Promedio** de la colección de un autor, demostrando la lógica de la recursión de cola por consola.

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

Estructura del proyecto (Obligatorio)

1. **Programación Orientada a Objetos (POO):** Todo el sistema debe estar estructurado en **Clases** (ej. Clase Libro, Clase Usuario, Clase GestorInventario). Las estructuras de datos (Pila y Cola) deben ser implementadas usando **Clases** o **métodos de clases**.

2. **Modularidad:** El código debe ser **modular**. Las diferentes secciones del proyecto (ej. AlgoritmosBusqueda.py, EstructurasDatos.py, ProblemasResueltos.py) deben estar separadas en **carpetas diferentes** y ser importadas correctamente al archivo principal.

3. **Documentación:** El código debe estar completamente **documentado** (siguiendo estándares vistos en clase). Cada clase, método, y algoritmo (Merge Sort, Backtracking, Recursión) debe tener una explicación clara de su propósito, parámetros y retorno.

Otras consideraciones finales

Aunque el documento no lo indique explícitamente, se debe tener una gestión de usuarios, libros, estantes, préstamos y reservas (Es decir, la posibilidad de crear, buscar, modificar, eliminar y listar cada uno de los registros de estas categorías).

Entregables

1. Documento con análisis y estrategias de solución para cada componente del proyecto.

2. Código fuente en Python completamente funcional y documentado.

3. Archivos de datos inicial (mínimo 20 libros de base)

4. Informe que incluya: descripción general del sistema, estructuras de datos utilizadas, algoritmos implementados, resultados de ejecución y justificación de decisiones.

5. Video o presentación mostrando el funcionamiento del sistema y explicación del diseño.

Nota importante: la documentación deberá estar en inglés. Los videos narrados en inglés tendrán valoración adicional.

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia