

Tema Principal: El grupo deberá definir un tema.

Por ejemplo: **Análisis de Datos Médicos**

El trabajo final se dividirá en dos partes: una parte teórica-práctica y una parte de codificación. El objetivo será analizar un conjunto de datos médicos reales o simulados utilizando las estructuras de datos vistas en clase para resolver problemas de organización, consulta y análisis de datos clínicos.

Requerimientos:

- **Lenguaje:** Python
- **Problema a Resolver (Ejemplo):** "Análisis y Gestión de Datos Médicos" (pueden ser datos de pacientes, diagnósticos, tratamientos, etc.).

Parte 1: Teoría y Realización Práctica

1. Desarrollo de Clases e Interfaces (Unidad 1):

- Define una clase Paciente que encapsule la información médica relevante (nombre, edad, historial de enfermedades, medicamentos, etc.).
- Crea interfaces para la gestión de los datos médicos, permitiendo añadir, eliminar, y modificar información.

2. Estructuras Recursivas (Unidad 2):

- Implementa un algoritmo recursivo que permita buscar en el historial de tratamientos de un paciente la aparición de ciertas enfermedades o medicamentos clave.

3. Árboles Binarios (Unidad 3):

- Usa un árbol binario de búsqueda para organizar los pacientes de una clínica por su número de identificación médica (ID). Debes permitir inserciones, eliminaciones, y búsquedas eficientes.

4. Árboles Generales (Unidad 4):

- Modela el historial clínico de cada paciente como un árbol general donde cada nodo es un evento médico (consulta, diagnóstico, tratamiento) y las ramas conectan las visitas y tratamientos asociados.

5. Cola de Prioridades y Heap Binaria (Unidad 5):

- Utiliza una cola de prioridades para gestionar la urgencia de los pacientes. Los pacientes más críticos deben ser atendidos primero, basándote en la gravedad de su condición.

6. Análisis de Algoritmos (Unidad 6):

- Analiza la eficiencia de los algoritmos utilizados (inserción en árbol binario, recorrido en árbol general, búsqueda en cola de prioridad) en términos de tiempo y espacio. Asegúrate de explicar la complejidad de cada uno.

Parte 2: Codificación y Algoritmos

1. Grafos (Unidad 7):

- Modela la red de hospitales y clínicas como un grafo. Los nodos representan hospitales y las aristas las conexiones entre ellos (distancias o tiempos de transferencia de pacientes).

2. Recorridos DFS y BFS (Unidad 8):

- Implementa un algoritmo DFS y BFS para encontrar el camino más corto entre hospitales para transferir un paciente en caso de emergencia.

3. Ordenamiento Topológico (Unidad 9):

- Utiliza ordenamiento topológico para modelar la secuencia de pasos necesarios para diagnosticar una enfermedad en base a los síntomas. Algunos pasos requieren la realización de pruebas antes de avanzar a la siguiente fase del diagnóstico.

4. Problemas NP y Camino Mínimo (Unidad 10):

- Analiza el problema del camino mínimo entre varios hospitales utilizando el algoritmo de Dijkstra para encontrar la mejor ruta para una ambulancia que necesita trasladar a un paciente crítico.

Entregables:

1. **Informe Teórico-Práctico:** Explicación de las decisiones de diseño, análisis de complejidad y justificación de las estructuras de datos utilizadas para resolver los problemas.
2. **Código en Python:** Implementación de las soluciones propuestas. Se evaluará la claridad, eficiencia y adherencia a buenas prácticas de programación.
3. **Conclusión:** Reflexión sobre cómo el uso de estructuras de datos optimizadas impacta el análisis y procesamiento eficiente de grandes volúmenes de datos médicos.

Fecha de entrega:

- A convenir según el calendario académico.

Este trabajo final permitirá integrar todos los conceptos de la materia en un contexto práctico y relevante como el análisis y gestión de datos médicos.