

EXAMEN DE CASOS APLICADOS – ALGORITMOS Y ESTRUCTURAS DE DATOS

Duración: 3 horas

Lenguaje permitido: C++ o Python

Entrega: código funcional

Caso 1: Sistema de emergencia hospitalaria

Un hospital regional implementará un sistema para administrar la atención de pacientes según su gravedad. Cada paciente tiene: nombre, DNI, nivel de emergencia (1-5), y tiempo estimado de atención.

Requisitos:

1. Diseña un sistema que permita atender a los pacientes priorizando los casos más graves.
2. Permitir ver en tiempo real la lista de pacientes pendientes.
3. Mostrar cuántos pacientes fueron atendidos y su tiempo total de atención.

Preguntas:

- ¿Qué estructura(s) de datos usarías y por qué?
- ¿Qué algoritmo aplicarías para mantener el orden dinámico de prioridades?
- Implementa el código que simula la atención con entrada de datos.

Caso 2: Análisis de rutas en una ciudad altiplánica

Una ciudad altiplánica con calles empinadas y rutas turísticas necesita un sistema que calcule si existe una ruta posible entre dos ubicaciones usando caminos existentes.

Se te proporciona un grafo no dirigido con nombres de puntos y las conexiones entre ellos.

Requisitos:

1. Implementar la entrada de datos para el grafo.
2. Verificar si existe camino entre dos puntos utilizando **DFS o BFS**.
3. Mostrar todas las rutas posibles de un punto a otro.

Preguntas:

- ¿Cuál estructura es más conveniente para representar el grafo?
- ¿Qué algoritmo conviene usar si se busca eficiencia?
- ¿Cómo optimizarías el recorrido?

Caso 3: Sistema de historial de comandos en una aplicación de diseño

Una aplicación de diseño gráfico permite a los usuarios deshacer y rehacer sus acciones (dibujar, mover, eliminar objetos).

Requisitos:

1. Modela este sistema usando estructuras de datos.
2. Simula el uso de los comandos deshacer y rehacer.
3. Incluye límite máximo de 10 acciones almacenadas para optimizar memoria.

Preguntas:

- ¿Qué estructura de datos es ideal para este comportamiento?
- Justifica cómo se maneja la reversión de acciones con tus estructuras.
- Simula con un menú sencillo: [1] Dibujar, [2] Mover, [3] Eliminar, [4] Deshacer, [5] Rehacer.

Evaluación (0 – 20 puntos)

Criterio	Puntaje Máximo
Análisis y elección de estructuras	6 puntos
Desarrollo y claridad del código	6 puntos
Funcionamiento correcto del programa	4 puntos
Explicación técnica y documentación	4 puntos
Total	20 puntos

Nota: Los estudiantes deberán desarrollar las soluciones de cada caso utilizando **interfaces gráficas simples** que permitan visualizar el comportamiento de las estructuras de datos implementadas (listas, pilas, colas, árboles o grafos). El uso de gráficos puede ser básico (por ejemplo, impresión visual por consola en forma de tablas o diagramas), pero debe representar claramente el flujo de los datos, las inserciones, eliminaciones, y/o recorridos. Se evaluará tanto la **funcionalidad del algoritmo** como la **capacidad de representar gráficamente la lógica estructural del problema**.