

## PRÁCTICA AED Estructuras de Datos lineales

### Aspectos a tener en cuenta:

- La fecha límite de entrega es el 19/12/2024 a las 23:59.
- No se permite la entrega mediante correo electrónico después de la fecha límite.
- La práctica se realizará individualmente.
- Se deberá de entregar un archivo PDF con la descripción de la práctica y los códigos fuente (.h y .cpp) con la resolución de la práctica.

### Objetivo:

Diseñar e implementar un proyecto en C++ que utilice estructuras de datos y algoritmos para implementar la funcionalidad de un sistema software.

### Descripción de la práctica:

La práctica consta de dos partes:

1. Simulación del sistema de un hospital. 6 puntos
2. Simulación de un sistema que elige el alumno. 4 puntos

## Parte 1

Simular un sistema de Hospital que consta de las siguientes clases:

- Medicamento
- Enfermedad
- Doctor
- Paciente
- Hospital

Se proporcionarán datos de objetos de estas clases mediante un fichero CSV que habrá que leer.

Se proporciona el esqueleto del proyecto CMake, la cabecera de la librería (.h) y un main y tests unitarios para probar la funcionalidad.

Se deben implementar las siguientes funcionalidades:

- `std::list<Medicamento> reponerStock_k(int k);`

Devuelve una lista de medicamentos donde se especifica la cantidad que falta para tener k unidades de cada medicamento.

- `std::list<Medicamento> reponerStock_niveles(std::list<Medicamento> nivelesAdecuados);`

Devuelve una lista de medicamentos donde se especifica la cantidad que falta para tener las unidades especificadas en la lista de entrada de cada medicamento.

- `std::pair<Doctor, Paciente> treatPaciente();`

Devuelve el par Doctor y paciente siendo el doctor el primero disponible y el paciente con mayor prioridad, actualizando los estados internos de ambos, utilizando las funciones miembro correspondientes.

- `int pagarDoctores();`

Actualiza el dinero del doctor de acuerdo con las enfermedades que ha tratado, actualizando la lista interna de Doctor. **La implementación debe ser recursiva.**

## Parte 2

Cada estudiante debe elegir o diseñar un sistema que cumpla estas características para la parte 2:

### 1. Descripción del sistema (0.5 puntos) en el PDF

Se debe describir el sistema, qué elementos tiene, porqué tienen sentido las estructuras de datos elegidas para cada propósito y qué algoritmos se van a utilizar.

**Se deberá comunicar al profesor durante las sesiones de proyecto en el aula para obtener el visto bueno del sistema que se quiere implementar.**

### 2. Diseño de AED: estructuras de datos y algoritmos (1 punto) en el PDF

Encontrar **todas las estructuras de datos y los algoritmos que se apoyen en ellas del sistema** y diseñarlas teniendo en cuenta los argumentos necesarios para poder realizarla.

Deberá tener **al menos 3 estructuras de datos y 3 algoritmos** para poder aprobar la práctica. Cuantas más haya y de mayor complejidad mejor será la calificación (suponiendo calidad de todo lo demás).

### **3. Implementación (2 puntos)**

Se evalúa el código, mediante ejecución y revisión de código, analizando su calidad (complejidad, buenas prácticas

**Si al ejecutar cualquier secuencia de comandos definida en el sistema en el apartado 2 se produce un error (crash) la calificación de este apartado será automáticamente 0.**

El código no debe aparecer bajo ningún concepto en el pdf, ni como capturas de pantalla, ni como texto.

### **4. Captura de trazas de ejecución (0.5 puntos) en el PDF**

Captura en texto o imágenes de las trazas correctas de ejecución del sistema, con diferentes casuísticas, explicando lo que sucede y por qué tiene sentido.

## Evaluación

Se evalúa cada parte según los puntos obtenidos en cada una de ellas.

**Esta práctica supone un 25% de la nota final de la asignatura, es necesario obtener al menos un 4 en esta práctica para poder aprobar la asignatura.**

La nota de las prácticas se guarda si está aprobada para la convocatoria extraordinaria en caso de no aprobar el examen de teoría.

## Extra (OPCIONAL)

Se valorará con hasta 2 puntos adicionales si toda la parte 2 está implementada utilizando recursión (en lugar de iteración).

Se valorará con hasta 1 punto adicional si se entregan 5 tests unitarios para la parte 2 del proyecto.

Se valorará con hasta 1 punto adicional si se implementan las estructuras de datos en lugar de utilizar las de la librería estándar.