

# Problema a Resolver:

Se necesita simular el funcionamiento de una sala de emergencias médicas donde los pacientes son atendidos según su nivel de criticidad en lugar de su orden de llegada. El objetivo es gestionar eficientemente la cola de espera priorizando a los pacientes más críticos, mientras se mantiene un sistema justo para casos con igual nivel de prioridad.

## La solución implementada:

Se utilizó una Cola de Prioridad implementada con un Montículo Binario (Min-Heap) para gestionar los pacientes según su nivel de riesgo implementandolo mediante una lista, dada por la clase ListsBinario.

## El sistema consta de:

- Un generador de pacientes aleatorios con distribución probabilística de riesgos
- Un montículo binario que garantiza extracción eficiente del paciente más prioritario
- Un mecanismo de desempate basado en tiempo de llegada para pacientes con igual riesgo
- Una simulación por ciclos que modela la llegada y atención de pacientes

Esto nos da un sistema de triage automatizado que optimiza la atención médica en situaciones de emergencia.

## La complejidad temporal es:

- Inserción en cola de prioridad:  $O(\log n)$
- Extracción del paciente más prioritario:  $O(\log n)$
- Generación de pacientes:  $O(1)$  por paciente
- Simulación completa:  $O(n \log n)$  para  $n$  ciclos

## La complejidad espacial es:

- Almacenamiento del montículo:  $O(n)$
- Estructuras de datos auxiliares:  $O(1)$
- Pacientes en espera:  $O(k)$  donde  $k$  es el tamaño máximo de la cola

## Conclusiones:

El algoritmo garantiza que los pacientes con mayor urgencia (menor número de riesgo) sean atendidos primero, con una eficiencia computacional óptima gracias al uso de montículos binarios. El sistema es escalable para grandes volúmenes de pacientes y mantiene equidad mediante el contador de llegada para casos con igual nivel de riesgo. La simulación demuestra un funcionamiento realista de un servicio de emergencias médicas, cumpliendo con los requisitos de priorización crítica en atención médica.