

## Informe Trabajo Práctico N°2

### Problema 1: Sala de Emergencias.

#### Enunciado:

En una sala de emergencias, los pacientes son clasificados según su **nivel de riesgo**: 1 (crítico), 2 (moderado) y 3 (bajo). El objetivo es implementar una estructura de datos que permita que siempre se atienda primero al paciente con mayor prioridad clínica. En caso de empate, debe atenderse al que haya ingresado antes.

#### Solución:

Para la resolución de este problema se implementó un montículo binario, estructura de datos que permite almacenar elementos de manera ordenada según colas de prioridad.

En nuestro caso cada paciente se almacenará como una tupla:

(nivel\_de\_riesgo, hora\_ingreso, paciente)

El **nivel de riesgo** es la clave principal de prioridad (pacientes críticos primero, luego moderados y, por último, de riesgo bajo).

La **hora de ingreso** fue un atributo agregado a la clase Paciente, como criterio secundario en caso de que dos pacientes tengan el mismo riesgo.

El **objeto paciente** contiene los datos personales.

De esta manera, el sistema garantiza que siempre se atienda primero al paciente más crítico y, en caso de empate, al que haya ingresado antes.

#### Análisis de complejidad:

El montículo binario permite mantener siempre al paciente de mayor prioridad en la raíz del árbol.

**insertar()** en el montículo tiene un orden de complejidad  $O(\log n)$ , ya que puede ser necesario mover el paciente recién ingresado hacia arriba para mantener el orden.

**extraer\_min()** tiene complejidad  $O(\log n)$ , porque se saca la raíz y se reorganiza el árbol hacia abajo.

Consultar al paciente que sigue es una operación  $O(1)$ , ya que la raíz siempre contiene al paciente de mayor prioridad.