

Informe del Problema 1

En el contexto del problema se nos solicitó simular una sala de emergencias donde llegaban una cierta cantidad de pacientes con un determinado nivel de riesgo que podían ser bajo, moderado y crítico. Debimos implementar algún tipo de estructura la cual nos permitiera que los pacientes con mayor riesgo sean atendidos en primer lugar y que en caso de tener el mismo nivel de criticidad, sea el paciente que llegó antes el que sea atendido.

Para llevar a cabo este ejercicio, la cátedra nos proporcionó el código de la sala de emergencias y la clase paciente, por lo que tuvimos que pensar en dónde deberíamos almacenar estos pacientes. Dicha estructura de datos debería ser capaz de almacenarlos, ordenarlos según el criterio de criticidad (el cual era un dato numérico) y poder agregar o quitar pacientes. En base a esto pensamos que nuestra mejor opción era utilizar un montículo de mínimos ya que nos aseguraba que la raíz del mismo sea un paciente con la menor criticidad posible, pero minutos más tarde nos dimos cuenta que para el contexto del problema no tenía sentido que un montículo gestione todo este labor, para esta tarea era más adecuado una “cola de prioridad” por lo que creamos dicha estructura cuyo funcionamiento interno sea la del montículo de mínimo que habíamos implementado.

De esta forma conseguimos que los pacientes atendidos sean los de menor criticidad, pero nos faltaba considerar el caso donde haya muchos pacientes con el mismo nivel de riesgo, por lo que a nuestra clase paciente le agregamos un nuevo atributo el cual lo llamamos prioridad donde esperamos que hayan datos de tipo entero. La forma en que lo implementamos fue en que en nuestra sala de espera cuando llega un nuevo paciente recorramos la cola de prioridad (para ello hicimos iterable nuestra cola de prioridad y montículo) y en cada nodo de nuestro montículo le sumamos un 1 (uno) a su atributo prioridad pidiéndole a nuestro montículo que no solo compare los valores de criticidad para su ordenamiento, sino también que tenga en cuenta el atributo prioridad de cada paciente.

Una vez considerado este caso ya teníamos una estructura de datos la cual era capaz de ordenar a los pacientes de acuerdo al criterio solicitado