

Hoja de trabajo No. 8

Realizar: Un sistema de colas con prioridad para atender pacientes en la emergencia de un hospital.

Realizarse: INDIVIDUAL

Objetivos:

- Utilización del ADT Priority Queue.
- Implementación del Priority Queue empleando un **heap**.

Programa a realizar:



Usted fue contratado para desarrollar el sistema de atención de pacientes que llegan a la sección de Emergencias de un hospital. Al llegar un paciente, se llena su ficha de ingreso con:
Nombre del paciente, descripción del síntoma, código de emergencia.

El código de emergencia es una letra de la **A** a la **E**, que indica la prioridad con que debe ser atendido el paciente. La prioridad A es la que más rápido debe ser atendida y la E es la que puede atenderse más lentamente, ya que no presenta una condición mortal o de gran peligro para el paciente.

Usted decide hacer el sistema basado en una Priority Queue en la que se ingresan las fichas del paciente y se retira de esa cola al paciente que tenga la prioridad de atención más rápida. Las fichas del paciente vendrán en un archivo de texto (debe llamarse **pacientes.txt**) con los campos separados por comas. Por ejemplo:

Juan Perez, fractura de pierna, C
Maria Ramirez, apendicitis, A
Lorenzo Toledo, chikunguya, E
Carmen Sarmientos, dolores de parto, B

El sistema permite indicar cual paciente debe ser atendido. Así, que cuando el doctor pide el siguiente paciente, se retira de la cola y se muestra su nombre, síntoma y prioridad. El orden en que serán retirados los pacientes, uno a la vez:

Maria Ramirez, apendicitis, A
Carmen Sarmientos, dolores de parto, B
Juan Perez, fractura de pierna, C
Lorenzo Toledo, chikunguya, E



Tareas:

- Codifique la clase Paciente que contendrá la ficha del paciente. Notar que esta clase debe implementar la interfaz comparable. De esta forma se sabrá cual paciente debe ser atendido antes, dependiendo de su código de emergencia.
- Implemente la clase VectorHeap<E> extends Comparable<E>> implements PriorityQueue<E> para manejar una cola con prioridad basada en un Heap.
- Haga pruebas unitarias para los métodos que insertan y retiran un elemento del VectorHeap.
- Haga otra versión del programa, pero ahora use el Java Collection Framework para el PriorityQueue:
<http://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html>
Los métodos no necesariamente se llaman como se especificaron en la interface PriorityQueue.

Debe subir a Canvas todos los productos elaborados y los enlaces a su repositorio de github (o equivalente).

Calificación:

Aspecto	Puntos
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el Canvas.	10
Implementación de clase VectorHeap (que implementa un PriorityQueue)	10
Documentación Javadoc de VectorHeap	10
Pruebas unitarias de su implementación de VectorHeap	10
Programa de atención de emergencias que use su implementación de VectorHeap.	30
Programa de atención de emergencias que use la implementación proporcionada por el Java Collection Framework.	30
TOTAL:	100