

### Hoja de trabajo No. 8

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

**Realizarse:** en parejas.

**Competencias a desarrollar:**

- Utiliza el ADT Priority Queue para resolver el problema planteado.
- Implementa el ADT Priority Queue empleando un heap.
- Lleva un control efectivo de las versiones del sistema elaborado en la que colaboran ambos integrantes del grupo.
- Planifica de manera efectiva las tareas a realizar para resolver el problema planteado.
- Gestiona efectivamente el tiempo dedicado a la realización de la hoja de trabajo.

**Problema a resolver:**



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 muestra la lista en el orden en que deben ser atendidos:

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.
- Implemente la interface PriorityQueue haciendo uso del Java Collection Framework para el PriorityQueue:

<http://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>

Los métodos no necesariamente se llaman como se especificaron en la interface PriorityQueue.

### Informe de Gestión de Tiempo

- Elabore un informe de gestión de tiempo en el que debe incluir:
  - División en pequeñas tareas las actividades a realizar.
  - Para cada una de las tareas debe especificar:
    - Nombre de la tarea, descripción de la tarea, responsable de la tarea, fecha planificada para iniciarla, fecha estimada de terminación.
  - Un formulario LOGT por cada uno de los miembros del equipo. El formulario y las instrucciones para completarlo lo puede encontrar en los Anexos.
  - Una reflexión del desempeño del equipo en la realización de la hoja de trabajo. Cuáles fueron las principales dificultades y las fortalezas.

### Material a entregar en blackboard

- Archivo .pdf con los resultados de las tarea e.
- Archivos .java con el código elaborado.
- Vínculo del repositorio en github.

### Calificación:

Aspecto	Puntos
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el Blackboard. Hay contribuciones de TODOS los miembros del grupo.	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.	25
Programa de atención de emergencias que use la implementación proporcionada por el Java Collection Framework.	25
Informe de gestión del tiempo. Están presentes los formularios de todos los integrantes del grupo correctamente llenos, se hizo una planificación de las actividades a realizar para terminar la hoja de trabajo. Se hizo una reflexión del desempeño del equipo de desarrollo	10
<b>TOTAL:</b>	<b>100</b>

### Referencias:

- Interfaz Comparable: <https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html>
- PriorityQueue: <http://docs.oracle.com/javase/8/docs/api/java/util/PriorityQueue.html>
- Bailey, Duane A. (2007). Java Structures. Data Structures in Java for the Principled Programmer. Capítulo 13.