
	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 1

## INFORME DE TRABAJO PRÁCTICO

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Estructura de datos y Algoritmos</i>				
<b>TITULO DEL TRABAJO:</b>	<i>HeapQueuePriority</i>				
<b>NÚMERO DE TRABAJO:</b>	<i>1</i>	<b>AÑO LECTIVO:</b>	<i>2023</i>	<b>NRO. SEMESTRE:</b>	<i>2</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>25/05/2023</i>	<b>HORA DE PRESENTACIÓN</b>			
<b>INTEGRANTE (s)</b> <i>Mauricio Eduardo Zegarra Puma</i> <i>Cristhian Willians Puma Larico</i>				<b>NOTA (0-20)</b>	<i>Nota colocada por el docente</i>
<b>DOCENTE(s):</b> <i>Karim Guevara Puente de la Vega</i>					

INTRODUCCIÓN
<p>Realizamos este ejercicio con el objetivo de entender y comprender con mayor detalle todo lo visto hasta ahora en clases. Conceptos tales como Técnicas y Diseños de Algoritmos, recursividad, genericidad, entre otros. Usando como metodología la herramienta de GitHub y alcanzado satisfactoriamente un mejor aprendizaje y dominio sobre los temas ya mencionados.</p>
MARCO CONCEPTUAL
<p>Definir los conceptos utilizados o información referencial que hayan tomado en como base para resolver los problemas. Todo debe estar debidamente citado y referenciado a los documentos fuentes</p>
SOLUCIONES Y PRUEBAS
<p><b>EJERCICIO 5:</b> Se construye una cola de prioridad que utilice heap como estructura de datos.</p> <p><b>LA CLASE NODE:</b></p> <pre>public class Node &lt;E, T&gt; {      private E key;     private T data;</pre>

```
public Node(E key, T data) {
    this.key = key;
    this.data = data;
}

public E getKey() {
    return key;
}

public void setKey(E key) {
    this.key = key;
}

public T getData() {
    return data;
}

public void setData(T data) {
    this.data = data;
}
}
```

**AHORA HAREMOS LA ESTRUCTURA DE NUESTRO HEAP:**

```
import java.util.ArrayList;

public class HeapQueue <E extends Comparable<E>, T>{

    private ArrayList<Node<E, T>> queue;

    public HeapQueue() {
        this.queue = new ArrayList<Node<E, T>>();
    }

    public boolean isEmpty() {
        return this.queue.size() == 0;
    }

    public void enqueue (E key, T data) {
```

```
Node<E, T> aux = new Node<E, T>(key, data);
this.queue.add(aux);

int size = this.queue.size();

if (size > 1) {
    int pos = size - 1;
    int father = (pos - 1)/2;

    while
(this.queue.get(pos).getKey().compareTo(this.queue.get(father).getKey()) > 0) {
        Node<E, T> aux2 = new Node<E,
T>(this.queue.get(father).getKey(), this.queue.get(father).getData());

        this.queue.set(father, this.queue.get(pos));
        this.queue.set(pos, aux2);

        pos = father;
        father = (pos - 1)/2;
    }
}



public Node<E, T> Front() {
    return this.queue.get(0);
}

public Node<E, T> Back() {

    Node<E, T> menor = this.queue.get(0);

    for (int i = 1; i < this.queue.size(); i++) {
        Node<E, T> actual = this.queue.get(i);

        if (menor.getKey().compareTo(actual.getKey()) > 0) {
            menor = actual;
        }
    }
}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 4</p>

```

        return menor;
    }

    public String toString() {
        String str = "";

        for (Node<E, T> aux : queue) {
            str += aux.getKey() + ", ";
        }

        return str;
    }
}

```

#### LA PRUEBA DE TEST:

```



public class test {
    public static void main(String[] args) {
        HeapQueue<Integer, Integer> cola = new HeapQueue<Integer, Integer>();

        cola.enqueue(4,5);
        cola.enqueue(1,7);
        cola.enqueue(7,6);
        cola.enqueue(5,37);
        cola.enqueue(11,8);
        cola.enqueue(6,100);
        cola.enqueue(-4,96);
        cola.enqueue(0,3);
        cola.enqueue(-1,41);
        cola.enqueue(6,59);
        cola.enqueue(3,11);
        cola.enqueue(8,1);

        System.out.println(cola);

        System.out.println("La mayor prioridad es de: " + cola.Front().getKey()
+ " y pertenece al elemento con la informacion: " + cola.Front().getData());
    }
}

```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 5

```

        System.out.println("La menor prioridad es de: " + cola.Back().getKey()
+ " y pertenece al elemento con la informacion: " + cola.Back().getData());

    }
}

```

### LECCIONES APRENDIDAS Y CONCLUSIONES

En el desarrollo de este trabajo se han adquirido conocimientos valiosos sobre estructuras de datos y uso de clases genéricas. Además, se ha utilizado con éxito técnicas como el uso importante de los árboles binarios(HEAP), lo cual ha contribuido a mejorar la eficiencia y rendimiento de los algoritmos implementados.

### REFERENCIAS Y BIBLIOGRAFÍA

LINK DEL REPOSITORIO:

[https://github.com/MauricioZegarra/EDA\\_PriorityQueueHeap/blob/main/HeapQueue.java](https://github.com/MauricioZegarra/EDA_PriorityQueueHeap/blob/main/HeapQueue.java)