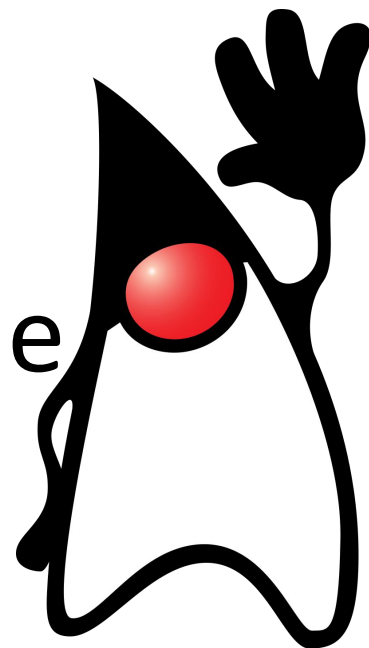




# Queue - PriorityQueue



Boris Molina Igor  
Taller de programación II  
[bmolintai@ing.ucsc.cl](mailto:bmolintai@ing.ucsc.cl)

# Imports

```
import java.util.Scanner;  
import java.util.List;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Arrays;  
import java.util.Stack;  
import java.util.Queue;  
import java.util.PriorityQueue;  
import java.util.LinkedList;  
import java.util.Map;  
import java.util.HashMap;  
import java.util.Set;
```

ó

```
import java.util.*;
```

# Queue

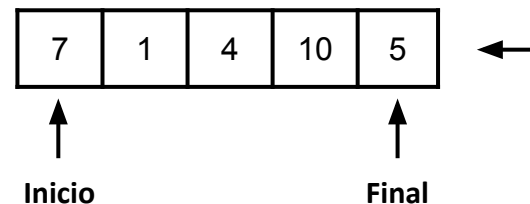
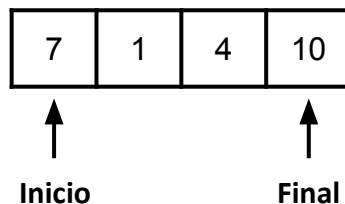
- Add
- Offer
- Element
- Peek
- Remove
- Poll

API Java Queue: <https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>

# Queue métodos

```
Queue<Integer> q = new LinkedList<>();
q.add(7);
q.add(1);
q.add(4);
q.add(10);
```

Tipo	Método	Descripción
boolean	<code>q.add(Object o);</code>	Inserta un objeto al final de la cola. Con Exception.
<code>q.add(5); //Inserta un objeto a la cola, en este caso un 5 [7, 1, 4, 10, 5]</code>		
boolean	<code>q.offer(Object o);</code>	Inserta un objeto al final de la cola. Sin Exception.
<code>q.offer(5); //Inserta un objeto a la cola, en este caso un 5 [7, 1, 4, 10, 5]</code>		

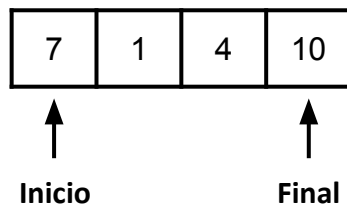


```
q.add(5);
ó
q.offer(5);
```

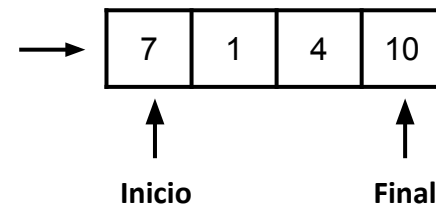
# Queue métodos

```
Queue<Integer> q = new LinkedList<>();
q.add(7);
q.add(1);
q.add(4);
q.add(10);
```

Tipo	Método	Descripción
Object	<code>q.element();</code>	Retorna el objeto al inicio de la cola. <b>Con Exception.</b>
<code>q.element(); //Retorna el primer objeto de la cola, en este caso un 7</code>		
Object	<code>q.peek();</code>	Retorna el objeto al inicio de la cola. <b>Sin Exception.</b>
<code>q.peek(); //Retorna el primer objeto de la cola, en este caso un 7</code>		



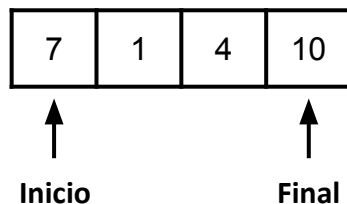
```
q.element();
ó
q.peek();
```



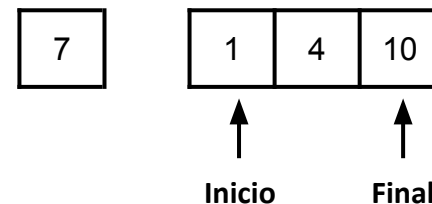
# Queue métodos

```
Queue<Integer> q = new LinkedList<>();
q.add(7);
q.add(1);
q.add(4);
q.add(10);
```

Tipo	Método	Descripción
Object	<code>q.remove();</code>	Retorna el objeto al inicio de la cola y lo elimina. <b>Con Exception.</b>
	<code>q.remove(); //Retorna el primer objeto de la cola y lo elimina, en este caso un 7</code>	
Object	<code>q.poll();</code>	Retorna el objeto al inicio de la cola y lo elimina. <b>Sin Exception.</b>
	<code>q.poll(); //Retorna el primer objeto de la cola y lo elimina, en este caso un 7</code>	



```
q.remove();
ó
q.poll();
```



# PriorityQueue

- Add
- Peek
- Poll

API Java PriorityQueue:

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

# PriorityQueue

- PriorityQueue con prioridad Ascendente

```
PriorityQueue<Integer> pq = new PriorityQueue<>();
```

- PriorityQueue con prioridad Descendente

```
PriorityQueue<Integer> pq = new PriorityQueue<>(Collections.reverseOrder());
```



# Map - HashMap - Set

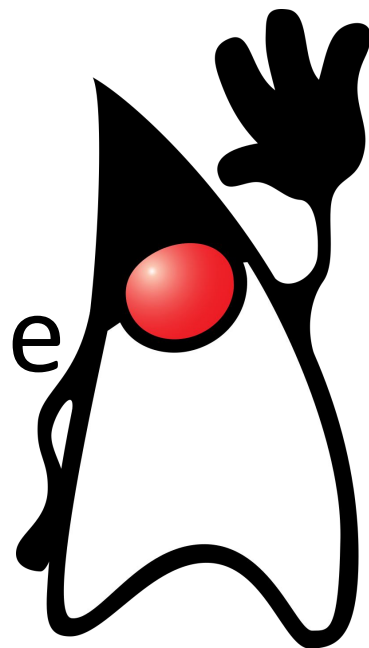
API Java HashMap: <https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>

API Java Map: <https://docs.oracle.com/javase/8/docs/api/java/util/Map.html>

API Java Set: <https://docs.oracle.com/javase/8/docs/api/java/util/Set.html>



# Queue - PriorityQueue



Boris Molina Igor  
Taller de programación II  
[bmolnai@ing.ucsc.cl](mailto:bmolnai@ing.ucsc.cl)