

Tema 2.3 TAD lineales

TAD Cola

Estructura de Datos y Algoritmos (EDA)

Contenidos

2.1. ¿Qué es un TAD Lineal?

2.2. TAD Pila

2.3. TAD Cola

2.4. TAD Lista

2.4.1 Implementación con una Lista Simplemente Enlazada

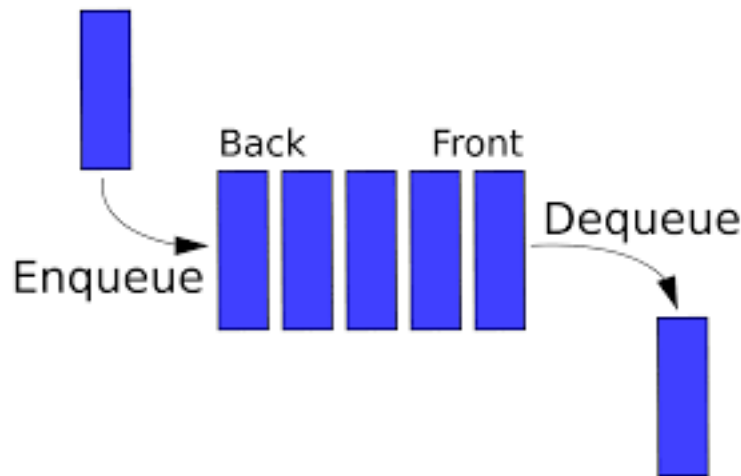
2.4.2 Implementación con una Lista Doblemente Enlazada

Objetivos

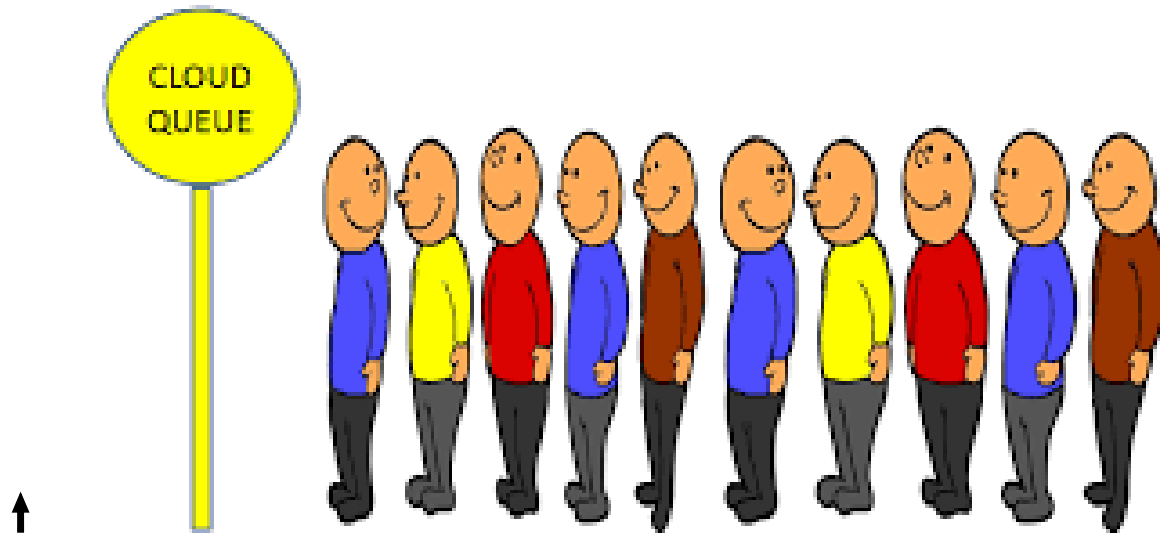
- ▶ Al final de la clase, los estudiantes deben ser capaces de:
 - ▶ Entender el principio FIFO (first-in, first-out)
 - ▶ Explica las operaciones de una Cola.
 - ▶ Especificar formalmente una Cola
 - ▶ Implementar una Cola usando una estructura dinámica

TAD Cola: Introducción

- ▶ Estructura de datos lineal basada en el principio **FIFO (first-in, first-out)**
- ▶ Eliminamos (**dequeue**) el primer elemento de la Cola.
- ▶ Insertamos (**enqueue**) un nuevo elemento al final de la Cola



TAD Cola



TAD Cola. La Cola en impresión



Estado de impresión de documentos (mis trabajos)					
Archivo Trabajo Ver					
Trabajo	Documento	Impresora	Tamaño	Hora de envío	Estado
2547	MARIBEL	HP-Laserj...	123k	hace un minuto	Pendiente
2546	DATOS EXC...	HP-Laserj...	145k	hace un minuto	Pendiente
2545	cuestionari...	HP-Laserj...	210k	hace un minuto	Pendiente
2544	Diario clas...	HP-Laserj...	11k	hace un minuto	Pendiente
2543	GUÍA BÁSIC...	HP-Laserj...	2476k	hace un minuto	Pendiente
2542	A9ROXbjSdh	HP-Laserj...	4325k	hace 2 minutos	Procesando

Operaciones

- ▶ **enqueue** (Object e): añade el elemento e al final de la cola
- ▶ **dequeue**(): borra y devuelve el primer elemento de la cola
- ▶ **front**(): devuelve el primer elemento de la cola
- ▶ **isEmpty**(): devuelve true si la cola está vacía; en otro caso, false
- ▶ **getSize**(): devuelve el número de elementos de la cola

Ejemplo

<i>Operation</i>	<i>Output</i>	<i>Q</i>
enqueue(5)	—	(5)
enqueue(3)	—	(5, 3)
dequeue()	5	(3)
enqueue(7)	—	(3, 7)
dequeue()	3	(7)
front()	7	(7)
dequeue()	7	()
dequeue()	"error"	()
isEmpty()	true	()
enqueue(9)	—	(9)
enqueue(7)	—	(9, 7)
size()	2	(9, 7)
enqueue(3)	—	(9, 7, 3)
enqueue(5)	—	(9, 7, 3, 5)
dequeue()	9	(7, 3, 5)

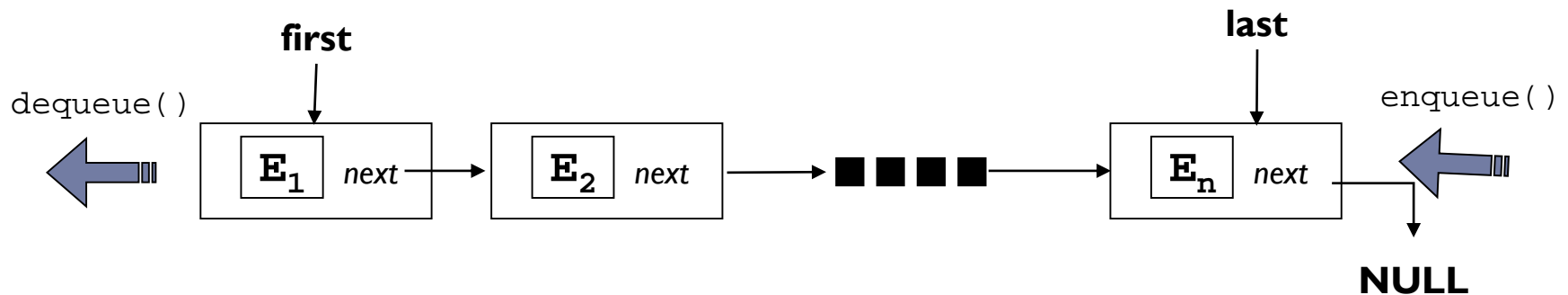
Tad Cola

```
public interface IQueue {  
  
    public boolean isEmpty();  
    public void enqueue(String elem);  
    public String dequeue();  
    public String front();  
    public int getSize();  
  
}
```

En estas diapositivas, usamos una cola de objetos String, pero recuerda que puede definir una cola de cualquier tipo de datos

Implementación de un TAD Cola usando una estructura dinámica

- Una cola puede ser representado como una secuencia de nodos
- Solo permitimos eliminar el primer elemento en la cola
- Solo permitimos insertar al final de la cola



Implementación de un TAD Cola usando una estructura dinámica

```
public class SQueue implements IQueue {  
  
    private SNode first;  
    private SNode last;  
    int size;  
  
    public boolean isEmpty() {  
        return first == null;  
    }  
}
```

front() method

```
public String front() {  
    if (isEmpty()) {  
        System.out.println("Queue is empty!");  
        return null;  
    }  
    return first.elem;  
}
```

enqueue() method

```
public void enqueue(String elem) {  
    SNode node = new SNode(elem);  
    if (isEmpty()) {  
        first = node;  
    } else {  
        last.next = node;  
    }  
    last = node;  
    size++;  
}
```

dequeue() method

```
public String dequeue() {  
    if (isEmpty()) {  
        System.out.println("Queue is empty!");  
        return null;  
    }  
    String firstElem = first.elem;  
    first = first.next;  
    if (first == null) {  
        last = null;  
    }  
    size--;  
    return firstElem;  
}
```