

Practica 04 Colas -

Ejercicio 1: Operaciones basicas de una cola.

```
class cola {  
    private int frente;  
    private int fin;  
    private String [] datos;  
  
    public Cola (int capacidad) {  
        datos = new String [capacidad];  
        frente = 0;  
        fin = -1;  
    }  
  
    public boolean estaVacia () {  
        return fin < frente;  
    }  
  
    public boolean estaLlena () {  
        return fin == datos.length - 1;  
    }  
  
    public void enqueue (String elemento) {  
        if (estaLlena ()) {  
            System.out.println ("Cola Llena");  
        } else {  
            fin = fin + 1;  
            datos [fin] = elemento;  
        }  
    }  
  
    public String dequeue () {  
        if (estaVacia ()) {  
            return null;  
        }  
    }  
}
```

```

} else {
    String temp = datos[frente];
    frente = frente + 1;
    return temp;
}
}
}

```

Simulación

Operación	frente	fin	Cola
Inicial	0	-1	[]
Enqueue	0	0	[A]
Enqueue	0	1	[A, B]
Enqueue	0	2	[A, B, C]
Enqueue	1	2	[B, C]
Enqueue	1	3	[B, C, D]
Enqueue	1	4	[B, C, D, E]
Enqueue	2	4	[C, D, E]

a) Estado Final: [C, D, E]

b) Elemento en frente: C (posición 2)

c) Elemento en fin: E (posición 4)

Practica 04 Colas

Ejercicio 2: Cola circular

```
class ColaCircular {  
    private frente;  
    private fin;  
    private String[] datos;  
    public ColaCircular(int capacidad) {  
        datos = new String [capacidad];  
        frente = 0;  
        fin = 0;  
    }  
    public boolean estaVacia () {  
        return frente == fin;  
    }  
    public boolean estaLlena () {  
        return (fin + 1) % datos.length == frente;  
    }  
    public void enqueue (String elemento) {  
        if (estaLlena ()) {  
            System.out.println ("Cola llena");  
            return;  
        }  
        datos [fin] = elemento;  
        fin = (fin + 1) % datos.length;  
    }  
}
```



```

public String dequeue () {
    if (estaVacia ()) return null;
    String temp = datos [frente];
    frente = (frente + 1) % datos.length;
    return temp;
}
}

```

- Simulación - capacidad 5 -

Inicial : frente=0, fin=0, []

Dequeue () frente=1, fin=0, [] vacía no hace nada

Enqueue (A) frente=1, fin=2, [_, A]

Enqueue (B) frente=1, fin=3, [_, A, B]

a) Nueva posición de frente : 1

b) Nueva posición de fin : 3

c) Si hacemos otro enqueue, se agrega en la posición 3 y fin pasara a 4.

Practica 04 Colas.

Ejercicio 3: Verificar uso correcto de colas
(teórico + razonamiento).

a) Personas esperando turno en una ventanilla:

Si es adecuado, porque siguen el principio FIFO, la primera persona en llegar es la primera en ser atendida, lo cual es exactamente el comportamiento de una cola.

b) Impresión de Documentos:

Si es adecuado, porque los documentos se imprimen en el orden en que fueron enviados a la impresora o sea el principio FIFO.

c) Desechar acciones: (UNDO):

No es adecuada, porque deshacer acciones requiere acceder a la última acción realizada o sea LIFO, lo que le corresponde a una pila, no a una cola.

d) Reservación de asientos:

Si es adecuado, porque las reservaciones se procesan en orden de llegada o sea FIFO, la primera persona en solicitar una reservación es la primera en ser procesada.

Miguel Lozano Arvizu GTID141

Practica 04 Colas

Ejercicio 4. Simulación de colas de proceso.

```
Class Proceso {  
    private String nombre;  
    public Proceso (String n) {  
        nombre = n;  
    }  
    public String getNombre () {  
        return nombre;  
    }  
    public void setNombre (String nombre) {  
        this.nombre = nombre;  
    }  
}
```

```
class ColaProcesos {  
    private int frente;  
    private int fin;  
    private Proceso [] procesos;  
    public ColaProcesos (int capacidad) {  
        procesos = new Proceso [capacidad];  
        frente = 0;  
        fin = -1;  
    }  
}
```

```

public void enqueue (Proceso p) {
    if (fin == procesos.length - 1) {
        System.out.println ("Cola llena");
        return;
    }
    fin = fin + 1;
    procesos [fin] = p;
}

```

```

public Proceso dequeue () {
    if (fin < frente) {
        return null;
    }
    Proceso temp = procesos [frente];
    frente = frente + 1;
    return temp;
}

```

Pregunta: El tercer proceso en ejecutarse es el P3.

- Simulación -

t = 0: Enqueue (P1) → Cola: [P1]

t = 1: Enqueue (P2) → Cola: [P1, P2]

t = 2: Enqueue (P3) → Cola: [P1, P2, P3]

t = 3: Dequeue (P1) → Cola: [P2, P3] (P1 Ejecutándose)

t = 4: Enqueue (P4), Dequeue (P2) → Cola: [P3, P4]
(P2 Ejecutándose)

Practica 04 Colas

Ejercicio 5 - BFS usando una cola

```
class ColaInt {
    private int [] datos;
    private int frente;
    private int fin;

    public ColaInt (int capacidad) {
        datos = new int [capacidad];
        frente = 0;
        fin = -1;
    }

    public void enqueue (int x) {
        fin = fin + 1;
        datos [fin] = x;
    }

    public int dequeue () {
        int temp = datos [frente];
        frente = frente + 1;
        return temp;
    }

    public boolean estaVacia () {
        return fin < frente;
    }
}
```


- Simulación BFS (A-B-C-D):

Grafo: 0-1-2-3

Iteración 1: Encolar \rightarrow Cola: [0]

Iteración 2: Desencolar: (0), Encolar: (1) \rightarrow Cola [1], Visitado: 0

Iteración 3: Desencolar: (1), Encolar: (2) \rightarrow Cola [2], Visitado: 0, 1

Iteración 4: Desencolar: (2), Encolar: (3) \rightarrow Cola [3], Visitado: 0, 1, 2

Iteración 5: Desencolar: (3) \rightarrow Cola [], Visitado: 0, 1, 2, 3

- Orden de visita: 0, 1, 2, 3, (A, B, C, D).

Practica 04 Colas.

Ejercicio 6 - Cola en un Banco.

```
class BancoCola {  
    private int frente;  
    private int fin;  
    private String[] clientes = new String [10];  
  
    public BancoCola() {  
        frente = 0;  
        fin = -1;  
    }  
  
    public void enqueue (String c) {  
        fin = fin + 1;  
        clientes [fin] = c;  
    }  
  
    public String dequeue () {  
        String temp = clientes [frente];  
        frente = frente + 1;  
        return temp;  
    }  
}
```

- Simulación -

1. Enqueue (Carla) \rightarrow frente = 0, fin = 0, Cola: [Carla]
2. Dequeue () \rightarrow frente = 1, fin = 0, Cola: []
3. Enqueue (Pedro) \rightarrow frente = 1, fin = 1, Cola: [Pedro]
4. Enqueue (Sofía) \rightarrow frente = 1, fin = 2, Cola: [Pedro, Sofía]
5. Dequeue () \rightarrow frente = 2, fin = 2, Cola: [Sofía]
6. Dequeue () \rightarrow frente = 3, fin = 2, Cola: []

a) Cola final: [] (vacía)

b) Quien queda al frente: Nadie (vacío)

c) Total personas atendidos: 3 (Carla, Pedro, Sofía)