

1. Operaciones básicas de una Cola

```
class Cola {
```

```
    private int frente = 0;
```

```
    private int fin = -1;
```

```
    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;
```

```
}
```

01 DIC 25

Elsy Jocelyn Gómez Juárez

```
Public void enqueue (String elemento) {  
    if (estallena ()) {  
    } else {  
        fin = fin + 1;  
        datos [fin] = elemento;  
    }  
}
```

```
Public String dequeue () {  
    if (estavacia ()) {  
        return null;  
    } else {  
        String temp = datos [frente];  
        frente = frente + 1;  
        return temp;  
    }  
}
```

Actividad

Simulación en Papel de las Operaciones

Representación

frente → Primera Posición Válida

fin → Última Posición Válida

Operación 1: Enqueue (A)

Cola:

[A, -, -, -, -]

frente = 0

fin = 0

Operación 2: Enqueue (B)

Cola:

[A, B, -, -, -]

frente = 0

fin = 1

Elsy Jazelyn Godínez Pérez

G1110141

01 Oct 25

Operación 3 Enqueue (C)

Cola:

[A, B, C, -, -]

Frente = 0

f_{in} = 2

Operación 4 Dequeue (C)

Se elimina A

Cola queda

[A, B, C, -, -]

Frente = 1

f_{in} = 2

Operación 5 Enqueue (E)

Cola:

[A, B, C, D, E]

Frente = 1

f_{in} = 3

Operación 6 Enqueue (E)

Cola:

[A, B, C, D, E]

Frente = 1

f_{in} = 4

Operación 7 Dequeue (C)

Se elimina B

Cola final

[A, C, D, E]

Frente = 2

f_{in} = 4

Respuestas:

a) Estado final de la Cola (de frente a f_{in})

Desde frente = 2 hasta f_{in} = 4

C, D, E

b) Elemento frente

Frente = 2

C

01 DIC 25

Elsy Josselyn Godínez Juárez

C) Elemento en fin

fin = 4

E

2. Cola Circular

```
class ColaCircular {  
    private int frente;  
    private int 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;
```

Elyz Josselyn Galíñez Juárez

return temp;

}

}

Actividad Simulación en Papel (capacidad = 5)

Capacidad = 5 → Índices: 0, 1, 2, 3, 4

Estado inicial

frente = 0 , fin = 0

Cola vacía

1: Dequeue()

La Cola está vacía → no cambia nada

frente = 0

fin = 0

2: Enqueue (A)

- Guardamos en posición fin = 0 abi

- Luego: fin = (0 + 1) % 5 = 1

Estado:

Índice 0 1 2 3 4

valor A

frente = 0

fin = 1

3: Enqueue (B)

- Guardamos en posición fin = 1

- Luego: fin = (1 + 1) % 5 = 2

Estado

Índice 0 1 2 3 4

valor A

frente = 0

fin = 2

a) Nueva Posición

0

b) Nueva Posición de fin

2

c) Que pasa si hacemos otro enqueue?

Se inserta sin problema la cola no se llena

porque solo tiene 2 elementos y la

capacidad es 5

$$fin = (2 + 1) \% 5 = 3$$

01 Dic 25

Elsy Joeselyn Godínez Juárez

3. Verificar Uso Correcto de Colas

a) Personas esperando turno en una ventanilla

Si es adecuado (caso PERFECTO para una cola) !!

Razón:

La Atención es primero en llegar, primero en ser atendido (FIFO)

La primera persona que llega debe ser la primera en salir

Es exactamente el comportamiento de una Cola.

b) Impresión de documentos

Si es adecuado (caso ABSURDO)

Razón

Las impresoras manejan trabajos en orden de llegada

el primer documento → Se imprime primero

Aunque algunas impresoras permiten prioridades, el comportamiento basico sigue FIFO, igual a una cola

c) Deshacer acciones (undo)

No es adecuado para una Cola

Razón

¿Cómo funciona? La última acción realizada → es la primera que se deshace

Eso es comportamiento LIFO, propio de una pila (stack) No de una Cola

- Cola : Primero en entrar, primero en salir

- Pila : último en entrar, primero en salir → ideal para undo

d) Reservaciones de asientos

No es adecuado para un Cola

Razón: las reservaciones no se alinean en orden de llegada

- el tiempo específico que el usuario elige

- disponibilidad por número

- errores no seriales

No se libera o asigna un asiento "en orden FIFO sino por indices o claves. Esto se modela mejor con":

- arreglos (asientos numerados)
- tablas hash
- lista indexadas

Pero no corresponde a una cola

4. Simulación de Cola de Procesos

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

```
class Cola Proceso  
(cola Circular)  
class ColaProceso {  
    private int frente;  
    private int fin;  
    private Proceso [] procesos;
```

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


Public void Enqueue(Proceso p) {

```
// Completar validacion de Pila Llena
fIn = (fIn + 1) % Procesos.length;
Procesos[fIn] = p;
```

}

Public Proces dequeue () {

```
// Completar validacion de Cola Vacia
Proceso temp = Procesos[frente];
frente = (frente + 1) % procesos.length;
return temp;
```

}

2. Simulación de Cola de Proceso ($t = 0 \text{ a } 4$)
 Llegadas

Proceso	Tiempo
P1	0
P2	1
P3	2
P4	4

Dibujar Cola en cada tiempo.

$t = 0$

Llega P1:

F \rightarrow [P1]

$t = 1$

Llegada P2

F1 \rightarrow [P1, P2]

$t = 2$

Llega P3

ädeku®

Elsy Josselyn Gómez Juarez

G11D141

01 DIC 25

F → [P₁, P₂, P₃]

t = 3

18 min llegadas

F → [P₁, P₂, P₃]

t = 4

Llega P₄:

F → [P₁, P₂, P₃, P₄]

¿ Cuál proceso es el tercero en ejecutarse?

Orden FIFO

1. P₁

2. P₂

3. P₃ ← tercero

5. BFS Usando una Cola (conceptual + Código)

Class ColaInt {

Private int []

Private int frente, fin;

Public ColaInt(int capacidad) {

datos = new int [capacidad];

frente = 0;

fin = -1;

}

Public void enqueue (int x) {

fin = (fin + 1) % datos.length;

datos [fin] = x;

}

01 dic 25

Elsy Josselyn Godínez Juárez

Public int deQueue()

```
int temp = datos[frente];
frente = (frente + 1) % datos.length;
return temp;
```

}

Public boolean notVacia()

```
return frente == (fin + 1) % datos.length;
```

}

}

BFS del grafo A - B - C - D

A = 0, B = 1, C = 2, D = 3

Grafo es una cadena

0 - 1 - 2 - 3

Simulación Poco a Poco

Estado Inicial

Cola = [0]

visitados = {0}

Salida = 0

Interacción 1: sacar 0

Vecino 1

Cola: [1]

visitados: {0, 1}

Salida 0 1

Interacción 2: sacar 1

Vecino 2

Cola: [2]

visitados: {0, 1, 2}

Salida: 0 1 2

Interacion 3: sacar 2

Vecino 3

Cola: [3]

visitados: {0, 1, 2, 3}

Interacion 4: sacar 3

No tiene vecinos nuevos

Cola: []

Ordena final de visita

0, 1, 2, 3

(A, B, C, D)

4. Cola en un banco

Completa el código

$f_{in} = f_{in} + 1;$

frente = frente + 1;

Simulación

1: Enqueue (Carla)

F → [Carla]

2: Dequeue (C)

Sale Carla → Cola vacia

F → []

3: Enqueue (Pedro)

F → [Pedro]

4: Enqueue (Sofia)

F → [Pedro, Sofia]

Dos Enqueue ()

1er dequeue → Sale Pedro

F → [Sofia]

• 2do dequeue → Sale Sofia

F → []

Respuestas finales del banco

a) Cola final: [] (vacia)

b) Quien queda al frente: Nadie, esta vacia

c) Total de Personas: 0

6. Cola en un banco

```
Class BancoCola {  
    private int frente, fin;  
    private String [] clientes = new String [10];
```

```
Private void enqueue (String c) {
```

```
    fin = fin + 1;
```

```
    Clientes [fin] = c;
```

```
}
```

```
Public String dequeue () {
```

```
    String temp = Clientes [frente];
```

```
    frente = frente + 1;
```

```
    return temp;
```

```
}
```

```
}
```

Simulacion en Papel Paso a Paso

[Estado Inicial]

frente = 0

fin = -1

Cola = []

1: Enqueva (Carla)

fin = 0

Cola :

[Carla]

↑ frente

↑ fin

2. Dequeva C)

Sale Carla

frente = 1

[Carla]

↑ frente

fin = 0

(en Cola queda vacia aunque los indices hayan avanzado)

3 : Enqueva (Pedro)

fin = 1

[Carla, Pedro]

↑ fin

↑ frente

4: Enqueva (Sofia)

[Carla, Pedro, Sofia]

↑ fin

↑ frente

Dos Dequeues()

primer dequeue \rightarrow sale Pedro

frente = 2

[Carla, Pedro, Sofia \rightarrow Sale Pedro

frente 2

[Carla, Pedro, Sofia]

\uparrow frente

\uparrow fin

Segundo dequeue \rightarrow sale Sofia

frente = 3

[Carla, Pedro, Sofia]

\uparrow frente

\uparrow fin

La cola queda vacia nuevamente

Respuestas

a) Cola final: vacia

b) Quien quedo al frente: Nadien, la cola esta vacia

c) Total de personas en la cola: 3