



Renaissance nearpod

Nombre completo

Jesus Talat Otero Hernandez

Nombre (optional)

Jesus Otero

Únete a la lección →

Colas

Estructura de Datos

Estructura de Datos **COLAS**

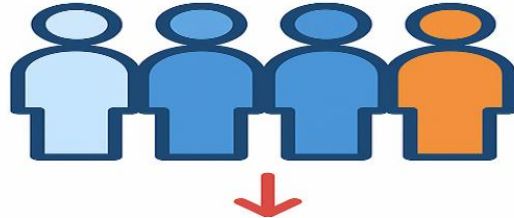


OBJETIVOS

- Comprender el concepto de Cola (TAD)
- Distinguir entre Pilas y Colas
- Implementar Colas en Java (array y lista enlazada)
- Aplicar colas a problemas reales

UNA COLA

es una estructura de datos lineal que sigue el principio **FIFO** (First In, First Out):



El primero en entrar es el primero en salir.

Ejemplo:



fila en un supermercado



cola de impresión



atención a clientes

Pregunta 2 / 3

¿Cuál de las siguientes clases en Java implementa la interfaz Queue?

☐ A. ArrayList

☐ B. HashMap

☐ C. TreeSet

☒ D. LinkedList

Pregunta 1 / 3

¿Qué es una cola en Java?

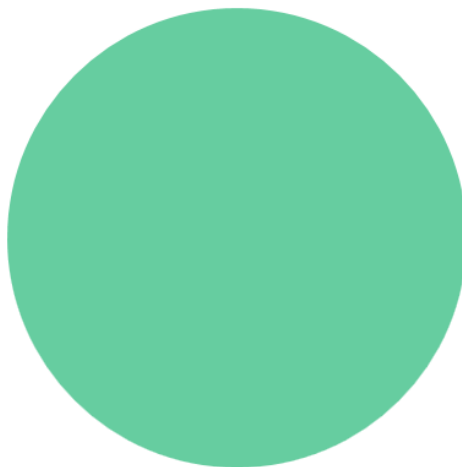
- ☒ A. Una estructura de datos que sigue el principio FIFO.
- ☐ B. Una colección de elementos desordenados.
- ☐ C. Una estructura de datos que sigue el principio LIFO.
- ☐ D. Una estructura de datos que permite acceso aleatorio.

Pregunta 3 / 3

¿Qué estructura de datos se utiliza comúnmente para implementar una cola en Java?

- ☐ A. Solo listas enlazadas
- ☐ B. Árboles
- ☒ C. Array o lista enlazada
- ☐ D. Solo arrays

ACERTASTE 3 DE 3













100 CORRECTA(S)

0 INCORRECTA(S)

0 SIN RESPUESTA

Operaciones Colas

Operación	Descripción de la Acción	Sinónimos Comunes
CrearCola	 Inicializa la estructura, dejándola vacía y lista para usar.	Inicializar
Insertar	 Añade un nuevo elemento al Final de la cola.	Enqueue , Añadir
Quitar	 Retira y devuelve el elemento que está en el Frente de la cola.	Dequeue , Eliminar
Frente	 Devuelve el elemento del Frente , pero no lo elimina .	Peek

Operación	Descripción
 crearCola	Inicializa una cola vacía.
 colaVacia	Verifica si la cola está vacía.
 colaLlena	Verifica si la cola ha alcanzado su capacidad máxima (en colas limitadas).
 insertar	Añade un nuevo elemento al final de la cola.
 frente	Consulta el elemento en el frente sin eliminarlo.
 retirar	Elimina el elemento en el frente de la cola.

Pregunta 1 / 4

¿Cuál es una operación común que se puede realizar en una Cola?

- ☐ A. Duplicar y dividir.
- ☒ B. Encolar y desencolar.
- ☐ C. Buscar y ordenar.
- ☐ D. Agregar y eliminar aleatoriamente.

Pregunta 2 / 4

¿Qué método se utiliza para agregar un elemento a una Cola en Java?

- ☒ A. add() o offer().
- ☐ B. get() o set().
- ☐ C. remove() o poll().
- ☐ D. insert() o append().

Pregunta 3 / 4

¿Qué método se utiliza para eliminar un elemento de una Cola en Java?

- ☐ A. insert() o append().
- ☐ B. peek() o top().
- ☐ C. add() o offer().
- ☒ D. remove() o poll().

Pregunta 4 / 4

¿Qué sucede si intentas desencolar de una Cola vacía usando remove()?

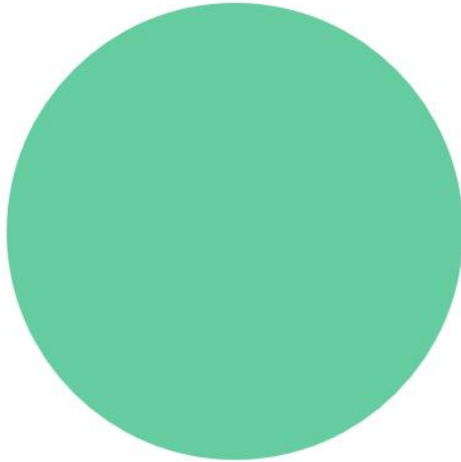
☒ A. Lanza una excepción NoSuchElementException.

☐ B. Devuelve un valor predeterminado.

☐ C. No hace nada.

☐ D. Devuelve null.

ACERTASTE 4 DE 4



100 CORRECTA(S)

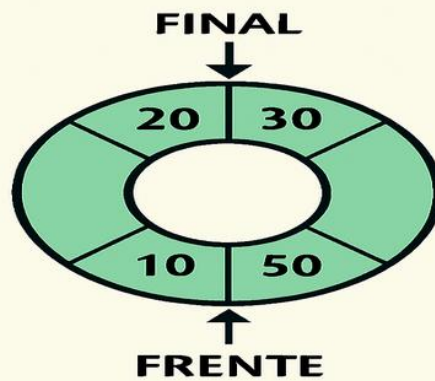
0 INCORRECTA(S)

0 SIN RESPUESTA

[Mis respuestas](#)

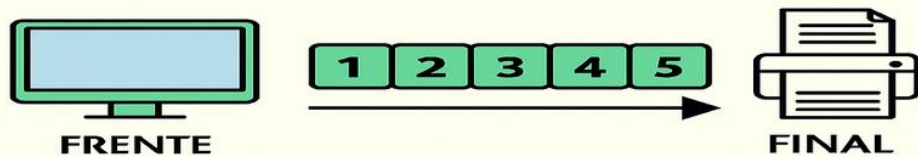
TAD COLA

El **TAD Cola** se puede implementar con **arrays** y con **listas enlazadas**. La implementación con un array lineal es muy ineficiente; se ha de considerar el array como una estructura circular y aplicar la teoría de **los restos** para avanzar el frente y el final de la cola.



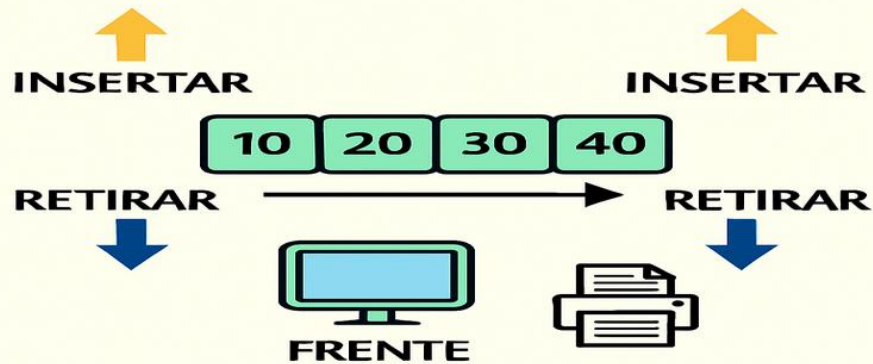
APLICACIONES DE LAS COLAS

Numerosos modelos de sistemas del mundo real son de tipo **cola**: la cola de impresión en un **servidor de impresoras**, los programas de simulación, las **colas de prioridades** de organización de viajes. Una cola es la estructura típica que se suele utilizar como almacenamiento de datos, cuando se envían datos desde un componente rápido de una computadora a un componente lento (por ejemplo, a una impresora).



BÍCOLAS

Las **bícolas** son **colas dobles**, las operaciones básicas de **insertar** y **retirar elementos** se pueden realizar por los dos extremos. A veces se ponen **restricciones de entrada** o de **salida** por algún extremo. Una bicola es, realmente, una extensión de una cola. La implementación natural del **TAD Bícola** es con una clase derivada de la clase **Cola**.



Implementación de una Cola en Java

Paso 1 Crear nodo

```
public class Nodo<T> {  
    private T dato;  
    private Nodo siguiente;  
  
    public Nodo(T data) {  
        dato = data;  
        siguiente = null;  
    }  
  
    //Implementar los Getters y Setters  
    //Implementar toString  
}
```

Crear la Cola

```
public class Cola<T> {  
    private Nodo<T> cabeza; // Puntero al Frente (por donde se quita)  
    private Nodo<T> cola; // Puntero al Final (por donde se inserta)  
    private int tamano; // Para llevar la cuenta del tamaño  
  
    public Cola() {  
        this.cabeza = null; // Inicializa una cola vacía  
        this.cola = null;  
        this.tamano = 0;  
    }  
  
    //Implementar los Getters y Setters
```

Verificar cola vacía

```
public boolean colaVacía() {  
    return cabeza == null;  
}
```

Encolar

```
public void insertar(T elemento) {  
    Nodo<T> nuevoNodo = new Nodo<>(elemento);  
  
    if (colaVacía()) { // Caso 1: La cola está vacía  
        cabeza = nuevoNodo;  
        cola = nuevoNodo;  
    }  
    else { // Caso 2: La cola NO está vacía  
        // El nodo actual que era el 'final' ahora apunta al nuevo nodo  
        this.cola.setSiguiente(nuevoNodo);  
        // El puntero 'cola' (final) se mueve al nuevo nodo  
        this.cola = nuevoNodo;  
    }  
    tamaño++; //Incrementar Tamaño  
    System.out.println("-> Insertado: " + elemento);  
}
```

Método desencolar

```
public T quitar() {  
    if (colaVacia()) {  
        System.out.println("Error: La cola está vacía.");  
    }  
  
    T datoQuitado = this.cabeza.getDato(); // Guardamos el dato a devolver  
  
    this.cabeza = this.cabeza.getSiguiente(); //Cabeza siguiente nodo  
  
    if (this.cabeza == null) { //Actualiza no hay elementos  
        this.colas = null;  
    }  
    tamaño--;  
    return datoQuitado;  
}
```

Peek

```
/*  
 * Operación: Frente (Peek)  
 * Devuelve el elemento del FRENTE sin retirarlo.  
 */  
public T frente() {  
    if (colaVacia()) {  
        System.out.println("Error: La cola está vacía");  
    }  
    return this.cabeza.getDato();  
}
```

Método Size

```
// Método auxiliar para ver el tamaño  
// (aunque no sea una operación base del TAD)  
public int getTamano() {  
    return tamano;  
}
```

Prueba

- Crear un objeto cola
- Agregar las tareas:

Tarea 1: Preparar café

Tarea 2: Revisar emails

Tarea 3: Iniciar proyecto

- Solicitar el tamaño
- Consultar Frente
- Quitar tarea 1 y despliega tarea ejecutada
- Quitar tarea 2 y despliega tarea solicitada
- Muestra el estado final de la Cola

Desafío

- Crear un menú para realizar las operaciones anteriores