



TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

Licenciatura en Ingeniería en Tecnologías de la Información e Innovación Digital (DSM)

Desarrollo de software Multiplataforma

Estructura de datos

Neardpod: Colas

UNIDAD II

Grupo: GTID141

Alumna: Serna Rodríguez Sara Lizbeth

1224100715

COLAS

A screenshot of a presentation slide titled "Colas Estructura de Datos". The slide has a dark background with the title in large, light-colored serif font. On the right side, there is a user profile menu for "Personal" with the email "123456789@igmail.com" and a status "La sincronización está activada". Below the profile, there are five items labeled "Perfil 1" through "Perfil 5", each with a status "Sesión no iniciada". At the bottom of the menu, there are options "Configurar un nuevo perfil personal" and "Otros perfiles". The slide is part of a presentation, as indicated by the "2 of 25" and "Open navigator" text at the bottom.

A screenshot of a presentation slide titled "Estructura de Datos COLAS". The slide features a diagram on the left showing three stylized human figures in blue and orange, with a red arrow pointing down to a blue rectangular box. To the right of the diagram, there is a blue icon of a person inside a square. Below this, the word "OBJETIVOS" is followed by a bulleted list: "Comprender el concepto de Cola (TAD)", "Distinguir entre Pilas y Colas", "Implementar Colas en Java (array y lista enlazada)", and "Aplicar colas a problemas reales". The slide is part of a presentation, as indicated by the "3 of 25" and "Open navigator" text at the bottom.

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

4 of 25 Open navigator

Question 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.

1 answer(s) selected

5 of 25 Open navigator

l

YOU SCORED 3 OUT OF 3

100 CORRECT
0 INCORRECT
0 NO ANSWER

Personal
1234190715.lee@gmail.com
La sincronización está activada

Perfil 1 Sesión no iniciada
Perfil 2 Sesión no iniciada
Perfil 3 Sesión no iniciada
Perfil 4 Sesión no iniciada
Configurar un nuevo perfil personal
Otros perfiles

My Answers

¿Qué es una cola en Java?

Una estructura de datos que sigue el principio FIFO. ✓

Una colección de elementos

5 of 25

Open navigator

22°C
Sábado

31:54 p.m.
24/05/2025

Operaciones Colas

Operación	Descripción de la Acción	Sinónimos
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

Personal
1234190715.lee@gmail.com
La sincronización está activada

Perfil 1 Sesión no iniciada
Perfil 2 Sesión no iniciada
Perfil 3 Sesión no iniciada
Perfil 4 Sesión no iniciada
Configurar un nuevo perfil personal
Otros perfiles

6 of 25

Open navigator

22°C
Sábado

31:57 p.m.
24/05/2025

Personal 1234190715.lee@gmail.com La sincronización está activada

Operación	Descripción
crearCola	Inicializa una cola vacía.
colaVacía	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.

7 of 25 Open navigator

25°C Sevilla

Question 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.

1 answer(s) selected

8 of 25 Open navigator

25°C Sevilla

Question 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().

1 answer(s) selected

8 of 25

Open navigator

Personal

1234190715.lee@gmail.com

La sincronización está activada

Perfil 1 Sesión no iniciada

Perfil 2 Sesión no iniciada

Perfil 3 Sesión no iniciada

Perfil 4 Sesión no iniciada

Configurar un nuevo perfil personal

Otros perfiles

20°C

8:08 p.m.

24/04/2025

Question 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().

1 answer(s) selected

8 of 25

Open navigator

Personal

1234190715.lee@gmail.com

La sincronización está activada

Perfil 1 Sesión no iniciada

Perfil 2 Sesión no iniciada

Perfil 3 Sesión no iniciada

Perfil 4 Sesión no iniciada

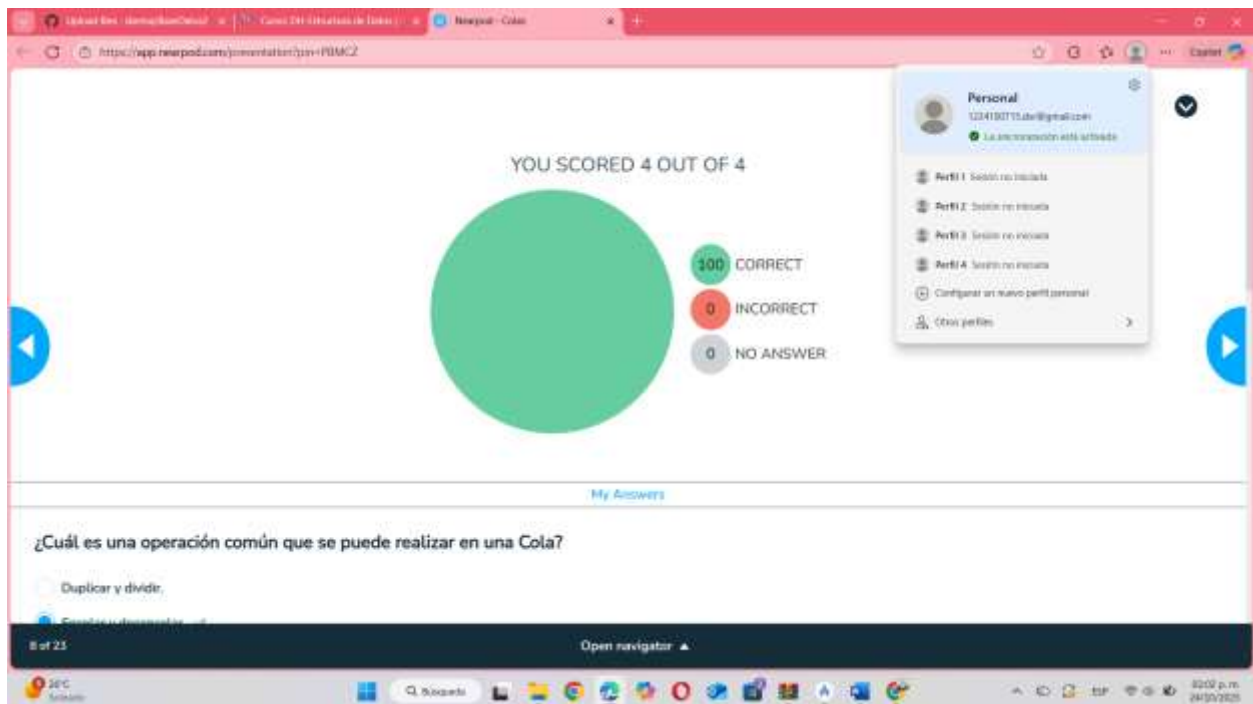
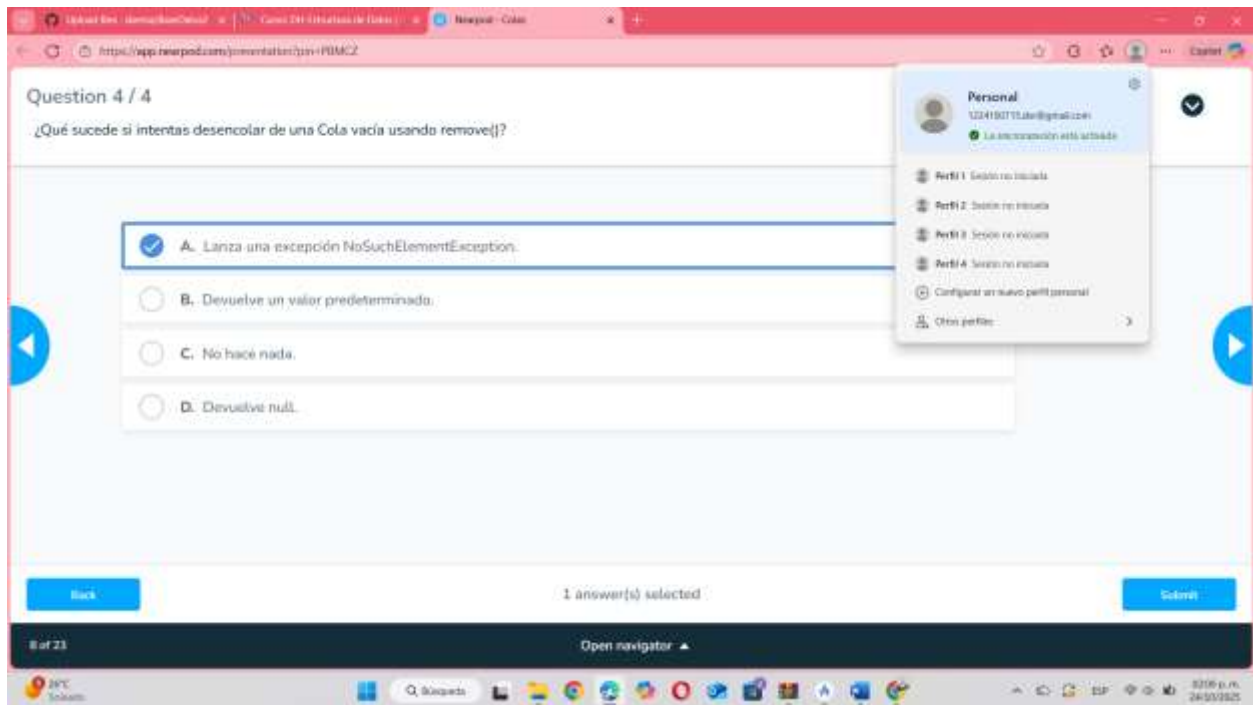
Configurar un nuevo perfil personal

Otros perfiles

20°C

8:08 p.m.

24/04/2025



Personal1234190715.lee@gmail.comLa sincronización está activadaPerfil 1 Sesión no iniciadaPerfil 2 Sesión no iniciadaPerfil 3 Sesión no iniciadaPerfil 4 Sesión no iniciadaConfigurar un nuevo perfil personalOtros perfiles

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.

3 of 21

Open navigator

25°C Torrevieja

Personal1234190715.lee@gmail.comLa sincronización está activadaPerfil 1 Sesión no iniciadaPerfil 2 Sesión no iniciadaPerfil 3 Sesión no iniciadaPerfil 4 Sesión no iniciadaConfigurar un nuevo perfil personalOtros perfiles

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).

10 of 23

Open navigator

25°C Torrevieja

Personal 1234190715.lee@gmail.com La implementación está activada

- Perfil 1 Sesión no iniciada
- Perfil 2 Sesión no iniciada
- Perfil 3 Sesión no iniciada
- Perfil 4 Sesión no iniciada
- Configurar un nuevo perfil personal
- Otros perfiles

BÍCOLAS

Las **bicolas** 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 Bicola es con una clase derivada de la clase Cola.

INSERTAR 10 20 30 40 RETIRAR FRENTE

11 of 23 Open navegador

20°C Santiago

8:07 p.m. (M) 3/3/2025

Personal 1234190715.lee@gmail.com La implementación está activada

- Perfil 1 Sesión no iniciada
- Perfil 2 Sesión no iniciada
- Perfil 3 Sesión no iniciada
- Perfil 4 Sesión no iniciada
- Configurar un nuevo perfil personal
- Otros perfiles

Implementación de una Cola en Java

12 of 23 Open navegador

20°C Santiago

8:07 p.m. (M) 3/3/2025

13 of 23

Paso 1 Crear nodo

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

Open navigator

28°C
Sábado

8:08 p.m.
24/05/2025

14 of 23

Crear la Cola

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

Open navigator

28°C
Sábado

8:11 p.m.
24/05/2025

Personal

12345678901234567890@gmail.com

La sincronización está activada

- Perfil 1 Sesión no iniciada
- Perfil 2 Sesión no iniciada
- Perfil 3 Sesión no iniciada
- Perfil 4 Sesión no iniciada
- Configurar un nuevo perfil personal
- Otros perfiles

Verificar cola vacía

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

Personal
1234190713.lee@gmail.com
La sincronización está activada

- Perfil 1 Sesión no iniciada
- Perfil 2 Sesión no iniciada
- Perfil 3 Sesión no iniciada
- Perfil 4 Sesión no iniciada
- Configurar un nuevo perfil personal
- Otros perfiles

15 of 23

Open navigator

20°C
Sábado

32:13 p.m.
24/10/2025

Encolar

```
public void insertar(T elemento) {  
    Nodo<T> nuevoNodo = new Nodo<T>(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);  
}
```

Personal
1234190713.lee@gmail.com
La sincronización está activada

- Perfil 1 Sesión no iniciada
- Perfil 2 Sesión no iniciada
- Perfil 3 Sesión no iniciada
- Perfil 4 Sesión no iniciada
- Configurar un nuevo perfil personal
- Otros perfiles

16 of 23

Open navigator

20°C
Sábado

32:12 p.m.
24/10/2025

Personal 1234190715.lee@gmail.com La sincronización está activada

Perfil 1 Sesión no iniciada
Perfil 2 Sesión no iniciada
Perfil 3 Sesión no iniciada
Perfil 4 Sesión no iniciada
Configurar un nuevo perfil personal
Otros perfiles

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.cola = null;  
    }  
  
    tamaño--;  
    return datoQuitado;  
}
```

17 of 23 Open navigator

22°C
Sábado

Personal 1234190715.lee@gmail.com La sincronización está activada

Configurar un nuevo perfil personal
Otros perfiles

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();  
}
```

18 of 23 Open navigator

22.03°C
Sábado

Personal 1234190715.abc@gmail.com La sincronización está activada
Configurar un nuevo perfil personal
Otros perfiles

Método Size

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

19 of 23 Open navigator

CAD/MEN 10.25% 32:15 p.m. 24/10/2020

Personal 1234190715.abc@gmail.com La sincronización está activada
Configurar un nuevo perfil personal
Otros perfiles

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

20 of 23 Open navigator

CAD/MEN 10.25% 32:16 p.m. 24/10/2020

EJERCICIO DE PRUEBA EN NETBEANS:

```
import java.util.LinkedList;

import java.util.Queue;

/**
 *
 * @author 12241
 * @Autor Sara Lizbeth Serna Rodriguez
 * Grupo: GTID0141
 * EJERCICIO DE PRUEBA COLA -----
 * 24/10/25
 *
 */
public class Prueba {

    public static void main(String[] args) {

        //cola
        Queue<String> colaTareas = new LinkedList<>();

        // Agregar las tareas
        colaTareas.add("Preparar café");
        colaTareas.add("Revisar emails");
        colaTareas.add("Iniciar proyecto");

        // Solicitar el tamaño
        System.out.println("Tamaño de la cola: " + colaTareas.size());
```



```
// Consultar Frente
System.out.println("Tarea al frente: " + colaTareas.peek());

//Quitar tarea 1 y despliega tarea ejecutada
String tareaEjecutada = colaTareas.poll();
System.out.println("Tarea ejecutada: " + tareaEjecutada);

// Quitar tarea 2 y despliega tarea solicitada
String tareaSolicitada = colaTareas.poll();
System.out.println("Tarea solicitada: " + tareaSolicitada);

// Muestra el estado final de la Cola
System.out.println("Estado final de la cola: " + colaTareas);

}
}
```

```
13 public class Prueba {
14
15
16
17
18
19
20
21
22
23 //cola
24 Queue<String> colaTareas = new LinkedList<>();
25 // Agregar las tareas
26 colaTareas.add("Preparar cafe");
27 colaTareas.add("Revisar email");
28 colaTareas.add("Iniciar proyecto");
29
30 // Inicializar el tamaño
31 System.out.println("Tamaño de la cola: " + colaTareas.size());
32 }
```

Output - Run (Prueba)

```
Running repasoII.jar on Windows
from pom.xml
----- [ jar ]-----
--- resources:1.3.1:resources (default-resources) @ RepasoII ---
skip non existing resourceDirectory C:\Users\12241\OneDrive\Documents\NetBeansProjects\RepasoII\src\main\resources
--- compiler:1.3.1:compile (default-compile) @ RepasoII ---
Nothing to compile - all classes are up to date.
--- exec:1.3.1:exec (default-cli) @ RepasoII ---
Tamaño de la cola: 3
Tarea al frente: Preparar cafe
Tarea ejecutada: Preparar cafe
Tarea solicitada: Revisar email
Estado final de la cola: [Iniciar proyecto]
-----
BUILD SUCCESS
-----
Total time: 0.888 s
Finished at: 2025-10-24T15:05:23-06:00
```

Desafío

- Crear un menú para realizar las operaciones anteriores

31 of 23

Open navigator

SEGUNDO EJERCICIO DE PUREBA EN ENTBEANS

```
package Colas;

import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

/**
 *
 * @author 12241
 * @Autor Sara Lizbeth Serna Rodriguez
 * Grupo: GTID0141
 * SEGUNDO EJERCICIO DE PRUEBA COLA -----
 * 24/10/25
 */

public class DesafioCola {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Queue<String> colaMascotas = new LinkedList<>();

        int opcion;

        do {

            System.out.println("\n===== MENÚ DE MASCOTAS =====");
```

```
System.out.println("1. Agregar mascota");
System.out.println("2. Mostrar tamaño de la cola");
System.out.println("3. Consultar frente");
System.out.println("4. Quitar primera mascota (atendida)");
System.out.println("5. Mostrar estado final de la cola");
System.out.println("6. Salir");
System.out.print("Selecciona una opción: ");
opcion = sc.nextInt();
sc.nextLine(); // limpiar

switch (opcion) {
    case 1:
        System.out.print("Ingresa el nombre de la mascota: ");
        String mascota = sc.nextLine();
        colaMascotas.add(mascota);
        System.out.println("Mascota agregada a la cola.");
        break;

    case 2:
        System.out.println("Tamaño actual de la cola: " + colaMascotas.size());
        break;

    case 3:
        if (colaMascotas.isEmpty()) {
            System.out.println("La cola está vacía.");
        } else {
```

```
        System.out.println("Mascota al frente: " + colaMascotas.peek());
    }
    break;

case 4:
    if (colaMascotas.isEmpty()) {
        System.out.println("No hay mascotas para quitar.");
    } else {
        String atendida = colaMascotas.poll();
        System.out.println("Mascota atendida: " + atendida);
    }
    break;

case 5:
    System.out.println("Estado actual de la cola: " + colaMascotas);
    break;

case 6:
    System.out.println("Saliendo del programa...");
    break;

default:
    System.out.println("Opción no válida. Intenta nuevamente.");
    break;
}
} while (opcion != 6);
```

```
sc.close();
```

```
}
```

```
}
```