



Universidad Tecnológica
del Norte de Guanajuato
Organismo Público Descentralizado del Gobierno del Estado
"Educación y progreso para la vida"

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

Colas

Estructura de Datos

2 of 25 Open navigator ▾

IEC

Personal 1234567890@outlook.com La información está actualizada

- Perfil 1 Sesión en marcha
- Perfil 2 Sesión en marcha
- Perfil 3 Sesión en ejecución
- Perfil 4 Sesión en ejecución
- Configurar un nuevo perfil personal
- Otras perfiles >

8:46 p.m.
24/3/2023

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

3 of 25 Open navigator ▾

IEC

Personal 1234567890@outlook.com La información está actualizada

- Perfil 1 Sesión en marcha
- Perfil 2 Sesión en marcha
- Perfil 3 Sesión en ejecución
- Perfil 4 Sesión en ejecución
- Configurar un nuevo perfil personal
- Otras perfiles >

8:46 p.m.
24/3/2023

4 of 25 Open navigator ▲

Personal
1234567890@ejercitacion.com
La información está actualizada

- 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
- Otras perfiles >

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

Open navigator ▲

4 of 25 Open navigator ▲

Personal
1234567890@ejercitacion.com
La información está actualizada

- 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
- Otras perfiles >

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

Next

5 of 25

Open navigator ▲



Q. Búsqueda



4:51 p.m.
24/3/2020

Question 2 / 3

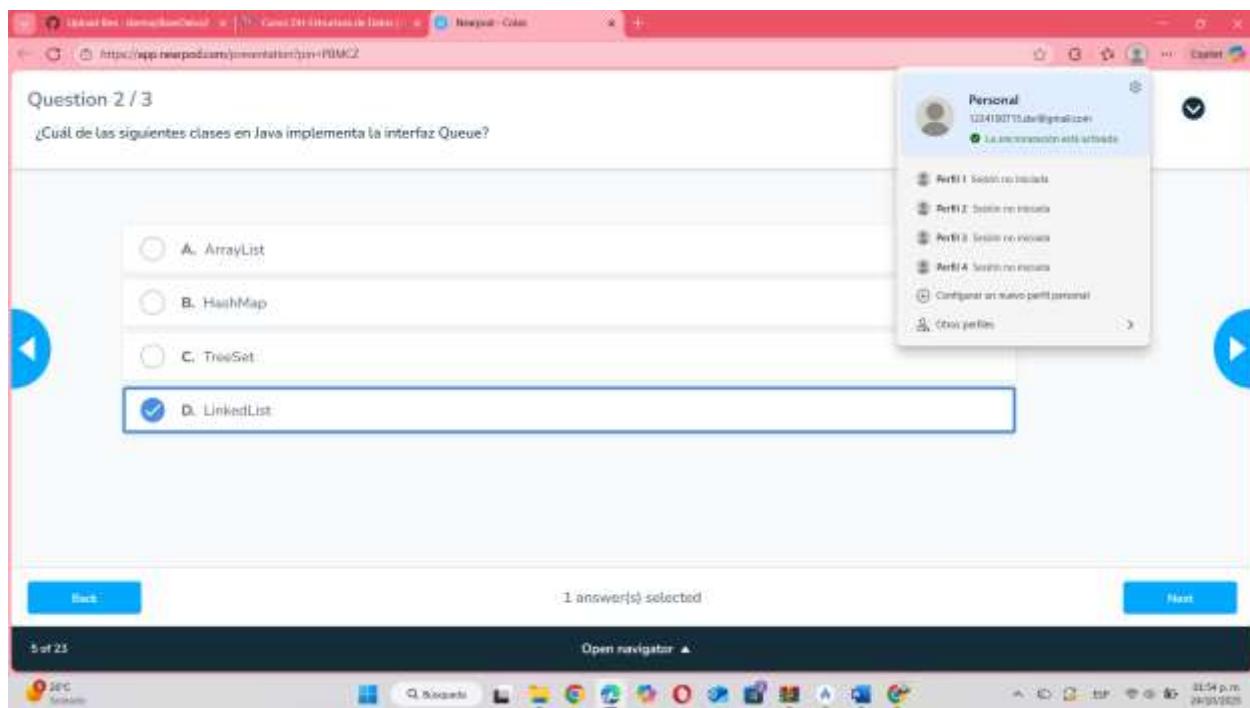
¿Cuál de las siguientes clases en Java implementa la interfaz Queue?

- A. ArrayList
- B. HashMap
- C. TreeSet
- D. LinkedList

Back Next

1 answer(s) selected

5 of 25 Open navigator



Question 3 / 3

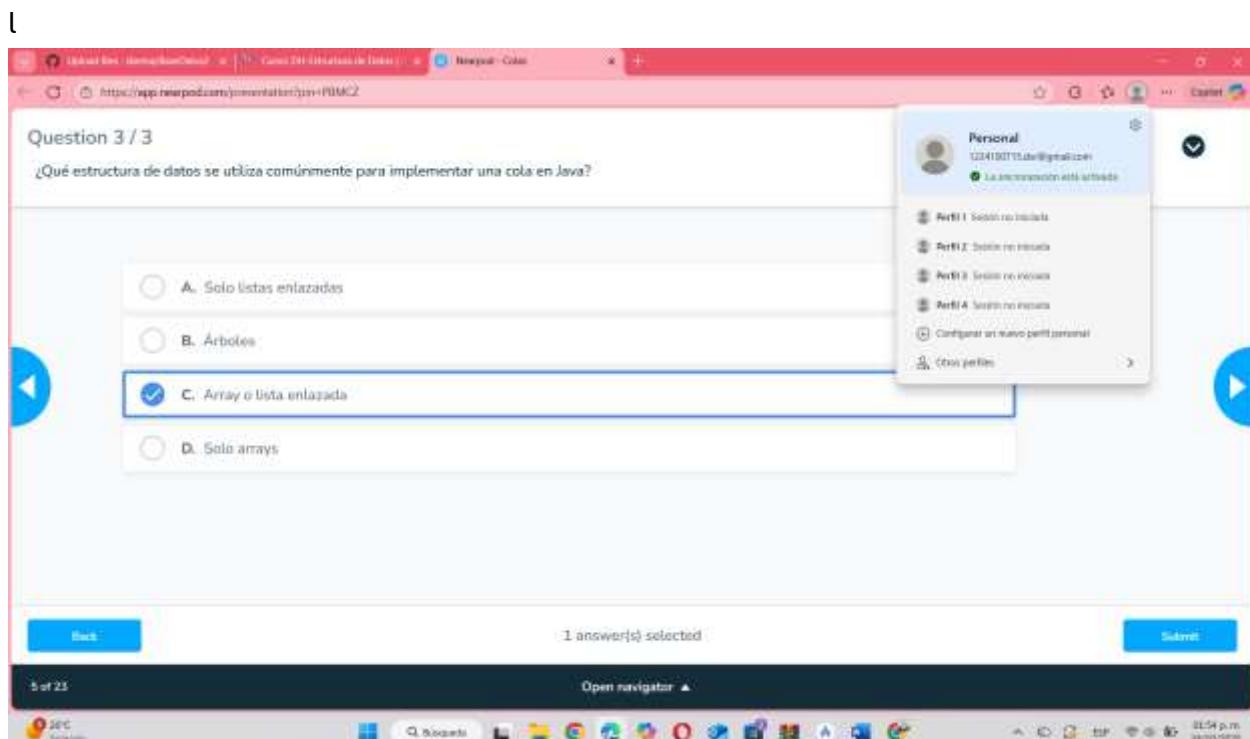
¿Qué estructura de datos se utiliza comúnmente para implementar una cola en Java?

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

Back Next

1 answer(s) selected

5 of 25 Open navigator



YOU SCORED 3 OUT OF 3

100% CORRECT
0% INCORRECT
0% NO ANSWER

My Answers

¿Qué es una cola en Java?

Una estructura de datos que sigue el principio FIFO. ✓
Una colección de elementos.

5 of 23 Open navigator ▲

Operaciones Colas

Operación	Descripción de la Acción	Síntesis
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

6 of 23 Open navigator ▲

7 of 25 Open navigator ▲

Personal
1234567890@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 >

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.
refilar	Elimina el elemento en el frente de la cola.

Open navigator ▲

8:57 p.m.
24/03/2025

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

Next

8 of 25 Open navigator ▲

Personal
1234567890@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 >

Open navigator ▲

8:59 p.m.
24/03/2025

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().

Personal
1234567890@ejercitacion.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

Otro perfil >

Back Next

8 of 25 Open navigator ▾

Open navigation bar

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().

Personal
1234567890@ejercitacion.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

Otro perfil >

Back Next

8 of 25 Open navigator ▾

Open navigation bar

Question 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.

1 answer(s) selected

8 of 23 Open navigator

Personal
1234567890@outlook.com
La autocorrección está activada

Perfil 1. Sesión en marcha
Perfil 2. Sesión en marcha
Perfil 3. Sesión en marcha
Perfil 4. Sesión en marcha
Configurar un nuevo perfil personal
Otro perfil >

YOU SCORED 4 OUT OF 4



100 CORRECT
0 INCORRECT
0 NO ANSWER

My Answers

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

Duplicar y dividir.

Encolar y desencolar.

8 of 23 Open navigator

Personal
1234567890@outlook.com
La autocorrección está activada

Perfil 1. Sesión en marcha
Perfil 2. Sesión en marcha
Perfil 3. Sesión en marcha
Perfil 4. Sesión en marcha
Configurar un nuevo perfil personal
Otro perfil >

3 of 25 Open navigator ▲

Personal
123456789@ejemplo.com
La información está actualizada

- 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
- Otro perfil ▾

TAD COLA

El **TAD Cola** se puede implementar con arrays y con listas enlazadas. La implementación con un array lineal es muy inefficiente; 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.

FINAL

FRENTE

Open navigator ▲

JEC

8:00 p.m.
26/03/2025

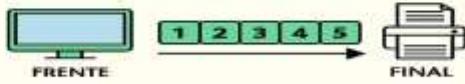
10 of 25 Open navigator ▲

Personal
123456789@ejemplo.com
La información está actualizada

- 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
- Otro perfil ▾

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).



Open navigator ▲

JEC

8:00 p.m.
26/03/2025

11 of 23

Open navigator ▲

Personal
1234567890@ejercito.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
- Otro perfil >

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 **restrictiones** de **entrada o de salida** por algún extremo. Una bícola es, realmente, una extensión de una cola. La implementación natural del TAD Bicola es con una clase derivada de la clase Cola.

The diagram illustrates a double-ended queue (Bícola) represented by four teal-colored boxes containing the numbers 10, 20, 30, and 40. Two yellow arrows labeled "INSERTAR" point upwards from above the first and third boxes. Two blue arrows labeled "RETIRAR" point downwards from below the second and fourth boxes. Below the queue, the word "FRENTE" is written next to a computer monitor icon, indicating the front of the queue.

11 of 23

Open navigator ▲

Personal
1234567890@ejercito.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
- Otro perfil >

12 of 23

Open navigator ▲

Personal
1234567890@ejercito.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
- Otro perfil >

Implementación de una Cola en Java

12 of 23

Open navigator ▲

Personal
1234567890@ejercito.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
- Otro perfil >

13 of 23 Open navigator ▲

Personal 1234567890@ejercitacion.com La sincronización está actualizada

Perfil 1 Sesión iniciada Perfil 2 Sesión iniciada Perfil 3 Sesión en ejecución Perfil 4 Sesión en ejecución Configurar un nuevo perfil personal Otras perfiles

Paso 1 Crear nodo

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

Open navigator ▲

13 of 23 Open navigator ▲

Personal 1234567890@ejercitacion.com La sincronización está actualizada

Perfil 1 Sesión iniciada Perfil 2 Sesión iniciada Perfil 3 Sesión en ejecución Perfil 4 Sesión en ejecución Configurar un nuevo perfil personal Otras perfiles

14 of 23 Open navigator ▲

Personal 1234567890@ejercitacion.com La sincronización está actualizada

Perfil 1 Sesión iniciada Perfil 2 Sesión iniciada Perfil 3 Sesión en ejecución Perfil 4 Sesión en ejecución Configurar un nuevo perfil personal Otras perfiles

Crear la Cola

```
public class Cola<T> {  
    private Nodo<T> cabeza; // Puntero al Frente (por donde se agrega)  
    private Nodo<T> cola; // Puntero al Final (por donde se saca)  
    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
```

Open navigator ▲

Verificar cola vacía

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

15 of 23 Open navigator ▾

Personal
1234567890@outlook.com
La sincronización está activada

- Perfil 1: Sincronización activada
- Perfil 2: Sincronización activada
- Perfil 3: Sincronización activada
- Perfil 4: Sincronización activada
- Configurar un nuevo perfil personal
- Otro perfil >

8:11 p.m.
24/05/2023

Encolar

```
public void insertar(T elemento) {  
    Nodo<T> nuevоНodo = new Nodo<T>(elemento);  
  
    if (colaVacia()) { // Caso 1: La cola está vacía  
        cabeza = nuevоНodo;  
        cola = nuevоНodo;  
    }  
  
    else { // Caso 2: La cola NO está vacía  
        // Si el nodo actual que era el 'final' ahora apunta al nuevo nodo  
        this.cola.setSigiente(nuevоНodo);  
        // Al ponerse 'cola' (final) se muevo al nuevo nodo  
        this.cola = nuevоНodo;  
    }  
    tamano++; // Incrementar Tamano  
    System.out.println("-> Insertado: " + elemento);  
}
```

16 of 23 Open navigator ▾

Personal
1234567890@outlook.com
La sincronización está activada

- Perfil 1: Sincronización activada
- Perfil 2: Sincronización activada
- Perfil 3: Sincronización activada
- Perfil 4: Sincronización activada
- Configurar un nuevo perfil personal
- Otro perfil >

8:12 p.m.
24/05/2023

17 of 23 Open navigator ▲

Personal
1234567890abc@gmail.com
La sincronización está activada

PerfIL Sólo en escritorio
PerfIL2 Sólo en escritorio
PerfIL3 Sólo en escritorio
PerfIL4 Sólo en escritorio
Configurar un nuevo perfil personal
Otras perfiles >

```
public T quitar() {
    if (colaVacia()) {
        System.out.println("Error: La cola está vacía.");
    }

    T datoQuitado = this.cabeza.getData(); // Guardamos el dato a devolver

    this.cabeza = this.cabeza.getSiguiente(); // Cambia siguiente nodo

    if (this.cabeza == null) // No hay elementos
        this.cola = null;
    else
        tamano--;
    return datoQuitado;
}
```

18 of 23 Open navigator ▲

Personal
1234567890abc@gmail.com
La sincronización está activada

Configurar un nuevo perfil personal
Otras perfiles >

```
/*
 * 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.getData();
}
```

19 of 23 Open navigator ▲

Personal 1234567890123456@gmail.com ✓ La sincronización está activada

Configurar un nuevo perfil personal O Otros perfiles >

Método Size

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

02:16 p.m.
26/03/2020

20 of 23 Open navigator ▲

Personal 1234567890123456@gmail.com ✓ La sincronización está activada

Configurar un nuevo perfil personal O 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

02:16 p.m.
26/03/2020

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);  
}  
}
```

Repositorio - Apache NetBeans IDE 28

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects X Services Files

Source History

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

Output - Run (Prueba)

```
Building repasoII.jar -> repasoII.jar  
--- resources/META-INF/resources (Default-resources) @ RepasoII ---  
skip non-existing resourceDirectory C:\Users\123\OneDrive\Documents\NetBeansProjects\RepassoII\src\main\resources  
--- compiler/META-INF/resources (Default-compiler) @ RepasoII ---  
Nothing to compile - all classes are up to date.  
  
--- execResources (Default-exec) @ RepasoII ---  
Tarea de la cola: 3  
Tarea al frente: Preparar café  
Tarea ejecutada: Preparar café  
Tarea solicitada: Revisar emails  
Estado final de la cola: [Iniciar proyecto]  
  
BUILD SUCCESS  
  
Total time: 0.000 s  
Finished at: 2025-10-24T15:05:23+00:00
```

Desafío

- Crear un menú para realizar las operaciones anteriores

21 of 23 Open navigator ▾

Open navigation bar

Personal

1234567890@gmail.com

La información está actualizada

Configurar un nuevo perfil personal

Otro perfil >

http://prepagosd.com/presentacion/10/PRMCF

SEGUNDO EJERCICIO DE PRUEBA 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. Eliminar mascota");
            System.out.println("3. Mostrar mascotas");
            System.out.println("4. Salir");
            System.out.print("Opción: ");
            opcion = sc.nextInt();

            switch (opcion) {
                case 1:
                    String nombre = sc.next();
                    colaMascotas.add(nombre);
                    System.out.println("Mascota agregada exitosamente.");
                    break;
                case 2:
                    if (!colaMascotas.isEmpty()) {
                        String nombre = colaMascotas.remove();
                        System.out.println("Mascota eliminada: " + nombre);
                    } else {
                        System.out.println("La cola está vacía.");
                    }
                    break;
                case 3:
                    if (!colaMascotas.isEmpty()) {
                        System.out.println("Mascotas en la cola:");
                        for (String mascota : colaMascotas) {
                            System.out.println(mascota);
                        }
                    } else {
                        System.out.println("La cola está vacía.");
                    }
                    break;
                case 4:
                    System.out.println("Saliendo del programa.");
                    break;
                default:
                    System.out.println("Opción no válida. Por favor, ingrese un número entre 1 y 4.");
            }
        } while (opcion != 4);
    }
}
```

```
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();  
}  
}
```