ESCUELA POLITÉCNICA NACIONAL

Carrera Software

Nombre: Fernando Eliceo Huilca Villagómez Fecha: 05/19/2024 Grupo:GR2

Profesora: Dra. Mayra CARRION

ESTRUCTURAS DE DATOS TIPO BICOLA

1. **PROBLEMA 1:** TRABAJAR LA IMPLEMENTACION DE LOS ALGORITMOS DE OPERACIONES BASICAS DE LA ESTRUCTURA DE DATOS TIPO BICOLA

   a. **ALGORITMOS OPERACIONES BÁSICAS EDA BICOLA**

```
Inicio AlgoritmoEliminarFrente
  si (frente == -1)
    Escribir "La biCola está vacía."
  sino
    biCola[frente] = null
    si (frente == fin)
      frente = -1
      fin = -1
    sino si (frente == tamaño - 1)
      frente = 0
    sino
      frente = frente + 1
Fin AlgoritmoEliminarFrente
```

```
Inicio AlgoritmoEliminarFin
  si (fin == -1)
    Escribir "La biCola está vacía."
  sino
    biCola[fin] = null
    si (frente == fin)
      frente = -1
      fin = -1
    sino si (fin == 0)
      fin = tamaño - 1
    sino
      fin = fin - 1
Fin AlgoritmoEliminarFin
```

```
Inicio AlgoritmoInsertarFrente
   si ((frente - 1 + tamaño) % tamaño == fin)
      Escribir "La biCola está llena."
   sino
      si (frente == -1)
         frente = 0
         fin = 0
      sino si (frente == 0)
         frente = tamaño - 1
      sino
         frente = frente - 1
      biCola[frente] = elemento
Fin AlgoritmoInsertarFrente
```

```
Inicio AlgoritmoInsertarFin
   si ((fin + 1) % tamaño == frente)
      Escribir "La biCola está llena."
   sino
      si (frente == -1)
         frente = 0
      fin = (fin + 1) % tamaño
      biCola[fin] = elemento
Fin AlgoritmoInsertarFin
```

b. IMPLEMENTACION

En la clase principal Main, declaración de variables y entrada de datos:

```java
// Autor: Fernando Huilca
public class Main {

    public static void main(String[] args) {
        // Declaración de variables para el tamaño de la biCola y la opción del menú
        int tamaño, opcion;

        // Solicitar al usuario el tamaño de la biCola
        tamaño = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el tamaño de la BiCola:"));
        BiCola biCola = new BiCola(tamaño);

        // Bucle para mostrar el menú y realizar las operaciones en la biCola
        do {
```

Bucle del menú principal:

```
// Bucle para mostrar el menú y realizar las operaciones en la biCola
do {
    opcion = Integer.parseInt(JOptionPane.showInputDialog(
                    "\nBiCola con Entrada Restringida." +
                    "\n1. Eliminar fin." +
                    "\n2. Eliminar frente." +
                    "\n3. Insertar fin." +
                    "\nBiCola con Salida Restringida." +
                    "\n4. Eliminar frente." +
                    "\n5. Insertar frente." +
                    "\n6. Insertar fin." +
                    //"\n_____" +
                    "\n7. Ver BiCola." +
                    "\n8. Salir."));

    // Ejecutar la acción correspondiente a la opción seleccionada
    switch (opcion) {
```

El constructor BiCola(int tamaño) se utiliza para inicializar el array biCola con el tamaño proporcionado tamaño y establecer los índices frente y fin en -1

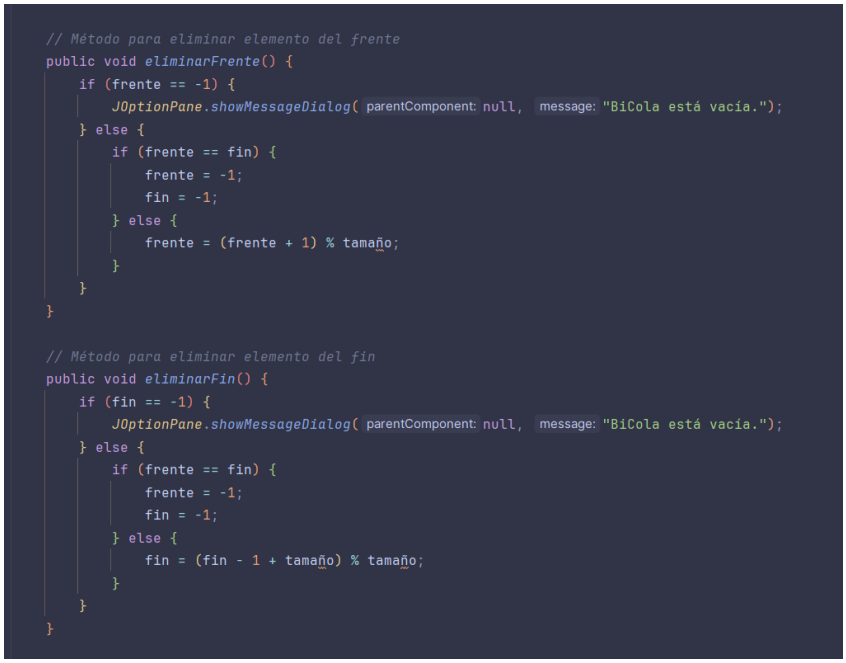
```
import javax.swing.*;

// Clase BiCola que representa una doble cola (deque)
class BiCola {
    private String[] biCola;
    private int frente, fin, tamaño;

    // Constructor para inicializar la biCola
    public BiCola(int tamaño) {
        this.tamaño = tamaño;
        biCola = new String[tamaño];
        frente = -1;
        fin = -1;
    }
```

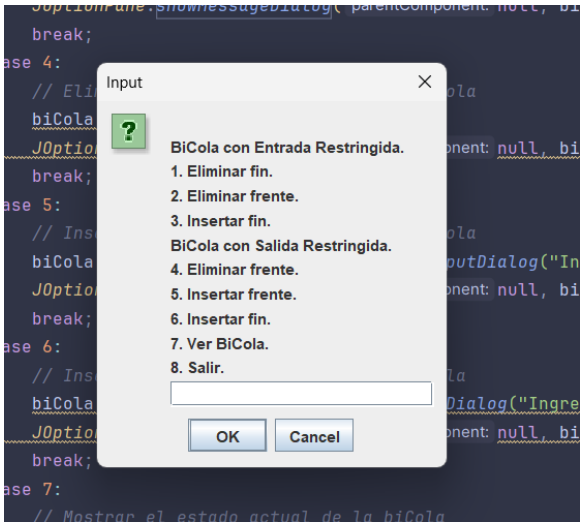
El código proporcionado incluye métodos para insertar elementos al final y al frente de una cola en Java. Se realizan comprobaciones de espacio disponible y se muestra un mensaje de error si la cola está llena. Las operaciones se realizan de forma circular utilizando índices modulares.

El código proporcionado incluye métodos para eliminar elementos del frente y del final de una cola en Java. Se realizan comprobaciones para verificar si la cola está vacía antes de eliminar elementos. Se utilizan operaciones aritméticas modulares para gestionar los índices de la cola de forma circular. Además, se muestra un mensaje indicando que la "BiCola está vacía" en caso de que no se pueda eliminar ningún elemento.

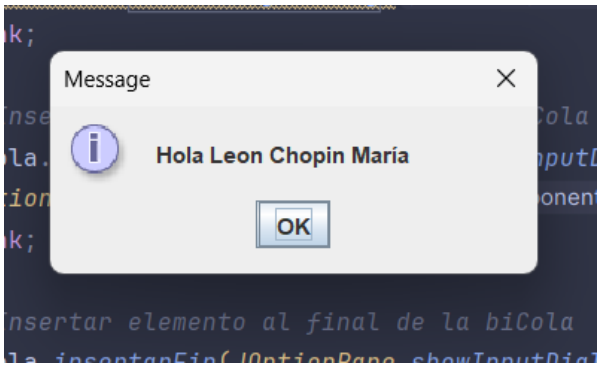


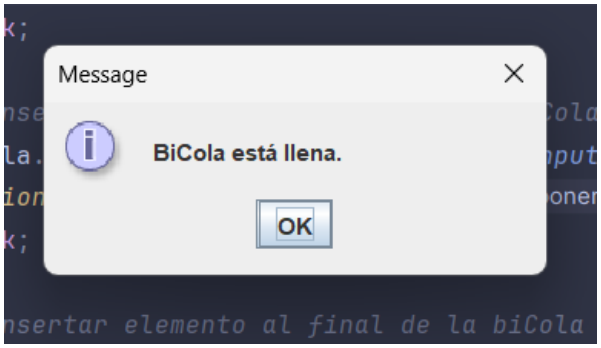
c. RESULTADOS

Se muestra la ventana emergente, luego de seleccionar el tamaño de la bicola, con las opciones que corresponde a una bicola dividida en Entrada restringida y salida restringida.



Se muestra un ejemplo de cola llena de variables de tipo String en un tamaño de cuatro elementos, y un mensaje de error en caso de querer insertar más elementos:





Se muestra como se puede eliminar tanto por el fin como por el frente :