Universidad de las Fuerzas Armadas

Nombre: *Jefferson David Yépez Morán*
NRC: *14543*
Docente: *Ing. Nancy Betancourt*
Asignatura: *Estructura de datos*
Fecha: *2024/06/15*

Documentación actividad 3:

- ***Descripción cola estática:*** *Consiste en un arreglo el cual tiene un tamaño fijo y el cual no se puede extender más durante la ejecución, en el cual se realizan las operaciones de ingresar datos y eliminar datos en la cola.*
- ***Descripción del caso de cola estática:*** *Un bibliotecario administra una cola de solicitudes de libros el cual tiene un orden, cada cliente es atendido según en el orden en el que realizaron la solicitud. Como la biblioteca cuenta con una cantidad de libros limitada, se emplea memoria estática para gestionar esta cola de solicitudes.*

```
int opcion;
cout << "En este ejercicio, el bibliotecario administra una cola de solicitudes de libros siguiendo el orden en que son recibidas. Cada cliente es ate

do {
    cout << "\n1. Agregar solicitud de libro\n";
    cout << "2. Atender solicitud de libro\n";
    cout << "3. Mostrar solicitudes en la cola\n";
    cout << "4. Salir\n";
    cout << "Ingrese una opción: ";
    cin >> opcion;
    switch (opcion) {
        case 1: {
            int id;
            cout << "Ingrese el ID del libro: ";
            cin >> id;
            insertarElementosCola(cola,id);
            break;
        }
        case 2:
            eliminarElementoCola(cola);
            break;
        case 3:
            imprimirCola(cola);
            break;
        case 4:
            cout << "Salir\n";
            break;
        default:
            cout << "Opción no válida\n";
    }
} while (opcion != 4);
```

- ***Funcionalidades:***

  1.- insertarElementosCola: *Ingresa elementos a la cola y los va añadiendo según el orden ingresado, colocándolos uno luego del otro hasta llegar al tamaño límite.*

  2.- eliminarElementoCola: *Elimina elementos de la cola, eliminando en orden de ingreso, en este caso desde el primer ingresado hasta el último.*

  3.- imprimirCola: *Imprime la cola desde el primero hasta el ultimo.*

deber4estatico.cpp

```cpp
#include <iostream>
#include <stdlib.h>

using namespace std;

int frente = -1;
int fin = -1;
int tamano = 5;
int contadorEliminados = 0;

void insertarElementosCola(int cola[], int numero){
    if(frente == -1 || fin == -1){
        frente = 0;
        fin = 0;
    }if(fin > tamano){
        cout<<"La cola esta llena!"<<endl;
    }else{
        cola[fin] = numero;
        fin++;
    }
}

void eliminarElementoCola(int cola[]){
     if (frente == -1 || frente > fin) {
        cout << "La cola está vacia" << endl;
    } else {
        cout << "Elemento removido: " << cola[frente] << endl;
        frente++;
}
}

void imprimirCola(int cola[]){
    int auxiliar = frente;
    for(int i = 0; i < tamano; i++){
        cout<<cola[auxiliar++]<<" ,";
    }
}
```

- ***Interfaz de usuario:***

```
En este ejercicio, el bibliotecario administra una cola de solicitudes de libros siguiendo el orden en que son recibidas. Cada cliente es atendido en la secuencia
en la que realizaron su solicitud. Dado que la biblioteca cuenta con un l├¡mite en la cantidad de libros disponibles, se emplear├í memoria est├ática para gestionar
 esta cola de solicitudes.

1. Agregar solicitud de libro
2. Atender solicitud de libro
3. Mostrar solicitudes en la cola
4. Salir
Ingrese una opci├│n: 1
Ingrese el ID del libro: 1
```

- ***Descripción cola dinámica:*** *Consiste en una lista simplemente enlazada en la cual usaremos dos punteros, “Frente” y “Fin” los cuales ayudarán a que la lista se comporte como una cola para poder implementar las operaciones de ingreso de datos y eliminación de datos en la cola.*
- ***Descripción de caso de cola dinámica:*** *Llevaremos a cabo la simulación de una fila de un supermercado, donde se atenderá a los clientes según el orden de su llegada. Se utilizarán listas dinámicas ya que no se conoce de antemano cuántos clientes llegarán a la fila para pagar sus compras.*

```cpp
int main(){
    int opcion;
    cout<< "En el presente ejercicio, llevaremos a cabo la simulación de una fila de un supermercado, donde se atenderá a los clientes según el orden de

    do {
        cout << "\n1. Agregar cliente a la fila de compras\n";
        cout << "2. Atender cliente \n";
        cout << "3. Mostrar fila de supermercado\n";
        cout << "4. Salir\n";
        cout << "Ingrese una opción: ";
        cin >> opcion;
        switch (opcion) {
            case 1: {
                int id;
                cout << "Ingrese el ID del libro: ";
                cin >> id;
                insertarElmentosCola(id);
                break;
            }
            case 2:
                eliminarElementoCola();
                break;
            case 3:
                imprimirCola();
                break;
            case 4:
                cout << "Salir\n";
                break;
            default:
                cout << "Opción no válida\n";
        }
    } while (opcion != 4);
```

- ***Funcionalidades:***

  1.- insertarElmentosCola: *Ingresa elementos a la cola y los va añadiendo según el orden ingresado, colocándolos uno luego del otro hasta que lo deseemos.*

  2.- imprimirCola: *Imprime la cola desde el primero hasta el ultimo.*

  3.- eliminarElementoCola: *Elimina elementos de la cola, eliminando en orden de ingreso, en este caso desde el primer ingresado hasta el último.*

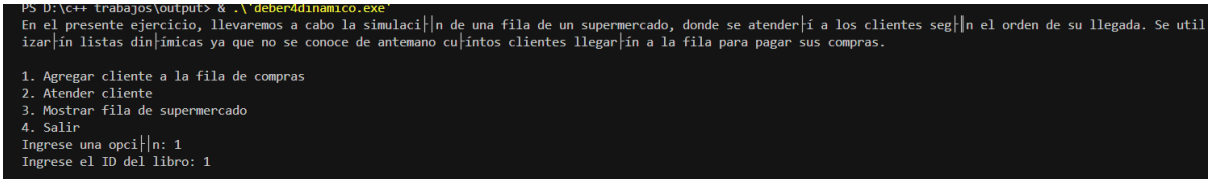
deber4estatico.cpp | deber4dinamico.cpp

deber4dinamico.cpp > eliminarElementoCola()

```cpp
#include <iostream>
#include <stdlib.h>

using namespace std;

typedef struct nodo
{
    int dato;
    nodo *siguiente;
}nodo;

nodo *frente = NULL;
nodo *fin = NULL;

void insertarElmentosCola(int numero){
    nodo *newNodo = new nodo();
    newNodo->dato = numero;
    newNodo->siguiente = NULL;
    if(fin == NULL){
        frente = newNodo;
        fin = newNodo;
    }else{
        fin->siguiente = newNodo;
        fin = newNodo;
    }
}

void imprimirCola(){
    if(frente == NULL){
        cout<<"La cola esta vacia"<<endl;
    }else{
        nodo *actual = frente;
        while (actual != NULL){
            cout<< actual->dato<<" ,";
            actual = actual->siguiente;
        }
    }
}
```

```
}

void eliminarElementoCola(){
    if (frente == NULL) {
        cout << "La cola está vacía\n";
        return;
    }
    nodo* actual = frente;
    frente = frente->siguiente;
    if (frente == NULL) {
        fin = NULL;
    }
    cout << "Elemento removido: " << actual->dato << endl;
    delete actual;
}
```

- ***Interfaz de usuario:***



- ***Análisis comparativo:***

  Las colas estáticas y dinámicas son estructuras de datos que siguen el principio FIFO (First In, First Out), pero se diferencian en cómo manejan el almacenamiento y la gestión de los elementos. A continuación, se presentan las semejanzas y diferencias entre ambas:

- ***Semejanzas:***

  1.- Orden de acceso: *Ambas colas operan siguiendo el principio FIFO, donde el primer elemento en entrar es el primero en salir.*

  2.- Operaciones básicas: *Las operaciones principales (inserción y eliminación) son las mismas en ambas colas. Se utilizan las funciones de enqueue (para agregar un elemento) y dequeue (para eliminar un elemento).*

  3.- Aplicaciones: *Ambas colas pueden utilizarse en aplicaciones similares, como la gestión de tareas en sistemas operativos, manejo de colas de impresión, y en algoritmos de búsqueda en anchura.*

- ***Diferencias:***

  1.- Almacenamiento:

  Cola estática: *Utiliza un array de tamaño fijo. La capacidad de la cola está limitada por el tamaño del array definido al momento de la creación.*

  Cola dinámica: *Utiliza nodos enlazados (listas enlazadas). La capacidad de la cola puede crecer y disminuir dinámicamente según sea necesario.*

  2.- Gestión de memoria:

Cola estática: *La memoria se asigna de manera continua y fija. No puede crecer más allá del tamaño predefinido del array.*

Cola dinámica: *La memoria se asigna de manera dinámica. Cada nuevo elemento se almacena en un nodo que se asigna en el momento de la inserción, permitiendo que la cola crezca según se necesite.*

3.- Complejidad de implementación:

Cola estática: *Suele ser más sencilla de implementar debido a su estructura de almacenamiento fija.*

Cola dinámica: *Es más compleja de implementar porque requiere la gestión de punteros para los nodos.*

4.- Uso de espacio:

Cola estática: *Puede ser ineficiente en el uso de espacio si el tamaño del array no se utiliza por completo.*

Cola dinámica: *Tiende a ser más eficiente en el uso del espacio, ya que sólo utiliza la memoria necesaria para los elementos presentes en la cola.*

5.- Tiempo de inserción y eliminación:

Cola estática: *Las operaciones de inserción y eliminación son generalmente O(1), pero puede haber excepciones en la gestión de casos como la cola circular.*

Cola dinámica: *Las operaciones de inserción y eliminación también son O(1), ya que se realizan mediante la manipulación de punteros.*

- ***Ventajas y Desventajas:***

Ventajas de una Cola Estática:

1.- Simplicidad: *Su implementación es más sencilla debido a su estructura fija.*

2.- Acceso rápido: *Operaciones como enqueue y dequeue pueden ser muy rápidas debido a la contigüidad de la memoria.*

3.- Menor sobrecarga de memoria: *No hay necesidad de almacenamiento adicional para punteros, lo cual puede reducir el uso de memoria.*

Desventajas de una Cola Estática

1.- Tamaño fijo: *No puede crecer más allá del tamaño predefinido, lo que puede llevar a desperdicio de memoria si el tamaño es sobreestimado, o a falta de espacio si es subestimado.*

2.- Fragmentación interna: *Puede haber espacio no utilizado si la cola no está llena.*

3.- Requiere gestión adicional: *Para implementar una cola circular, se necesita lógica adicional para gestionar el envolvimiento de los índices.*

Ventajas de una Cola Dinámica

1.- Flexibilidad: *Puede crecer y reducirse dinámicamente según las necesidades, evitando limitaciones de tamaño fijo.*

2.- Uso eficiente de memoria: *Solo utiliza la cantidad de memoria necesaria para los elementos presentes en la cola.*

3.- No hay necesidad de tamaño predefinido: *No es necesario conocer de antemano el número máximo de elementos.*

Desventajas de una Cola Dinámica

1.- Complejidad de implementación: *Requiere una implementación más compleja debido a la gestión de punteros.*

2.-Mayor sobrecarga de memoria: *Cada nodo necesita almacenamiento adicional para los punteros.*

3.- Posible fragmentación de memoria: *Puede haber fragmentación de memoria si se asignan y liberan muchos nodos dinámicamente.*