

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRONICA
CARRERA DE SOFTWARE



TEMA

COLAS CIRCULARES

ESTUDIANTES



CÓDIGOS



ASIGNATURA

ESTRUCTURAS DE DATOS

PROFESOR



FECHA DE ENTREGA

27/05/2024

RIOBAMBA - ECUADOR

Introducción

En el vasto mundo de la informática y la programación las estructuras de datos desempeñan un papel fundamental en el almacenamiento y manipulación eficiente de información entre estas estructuras las colas destacan por su sencillez y su capacidad para gestionar conjuntos de elementos de manera ordenada inspiradas en la noción de una fila en la vida real, las colas operan bajo el principio de "First In, First Out" (FIFO), asegurando que los elementos se manejen en el mismo orden en el que fueron agregados.

En este contexto las colas circulares emergen como una variante optimizada de las colas lineales ofreciendo ventajas significativas en términos de eficiencia y rendimiento este informe tiene mucha información que queda claro que es cola circular, la importancia de las colas en general como el potencial y las ventajas específicas de las colas circulares en la programación y la informática.

Desarrollo

Definición de colas

Las colas son estructuras de datos fundamentales en informática y programación, utilizadas para almacenar y gestionar conjuntos de elementos de manera ordenada. La principal característica de una cola es su disciplina de acceso, conocida como FIFO (First In, First Out), que asegura que el primer elemento en entrar en la cola será también el primero en salir. Este comportamiento es análogo a una fila en la vida real, como la fila en una taquilla o en un banco, donde los primeros en llegar son los primeros en ser atendidos.

"Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista (Fig. 1). Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, cabeza o frente) de la lista. Las aplicaciones utilizan una cola para almacenar elementos en su orden de aparición o concurrencia." (Joyanes Aguilar, (2006)).

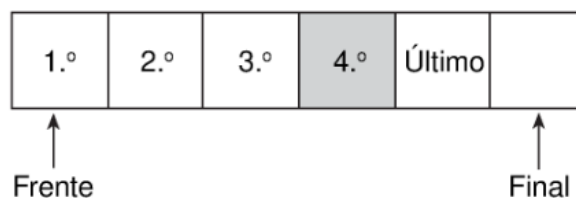


Fig. 1

Estructura de una Cola

Una cola se puede visualizar como una secuencia lineal de elementos, donde las operaciones principales son la inserción (enqueue) y la eliminación (dequeue). Los elementos se añaden al final de la cola, llamado

"cola" o "trasero", y se eliminan del inicio de la cola, conocido como "frente" o "cabeza". Este mecanismo asegura que el orden de los elementos se mantiene según el principio FIFO.

"

- *Los elementos se insertan de uno en uno: insertar*
- *Los elementos se extraen de uno en uno: extraer*
- *La posición de la cola donde se encuentra el siguiente elemento a ser extraído se denomina front (o head, inicio)*
- *La posición de la cola donde se colocará el siguiente elemento que se inserte se denomina rear (o tail, fin)*

" Ivan Cantador, (2024)

Importancia de las colas

Las colas representan una estructura de datos fundamental con una amplia gama de aplicaciones en diversos campos de la informática y otras disciplinas su importancia radica en su capacidad para gestionar eficientemente conjuntos de elementos en orden secuencial lo cual es crucial en múltiples escenarios donde el orden de procesamiento es vital. A continuación, se detallan algunas de las principales razones que subrayan la relevancia de las colas:

Gestión de Procesos en Sistemas Operativos: En los sistemas operativos las colas son esenciales para la administración de procesos los sistemas operativos modernos utilizan colas para gestionar la planificación de tareas permitiendo que los procesos se ejecuten de manera ordenada y eficiente los procesos en espera de recursos o tiempo de CPU se almacenan en colas garantizando que se atiendan en el orden correcto y optimizando el uso del procesador.

Administración de Recursos en Redes de Computadoras: Las colas desempeñan un papel crucial en la transmisión y gestión de datos en redes de computadoras los routers y switches utilizan colas para almacenar paquetes de datos temporalmente antes de encaminarlos a su destino final. Este proceso asegura que los paquetes se transmitan en el orden adecuado reduciendo la latencia y evitando la pérdida de datos las colas también ayudan a gestionar la congestión en las redes manteniendo un flujo constante y ordenado de información.

Sistemas de Atención al Cliente: En aplicaciones comerciales las colas se utilizan para gestionar a los clientes que esperan atención tanto en tiendas físicas como en centros de llamadas los sistemas de colas aseguran que los clientes sean atendidos en el orden en que llegaron mejorando la experiencia del cliente y la eficiencia operativa esto es particularmente importante en sectores donde el tiempo de espera y la satisfacción del cliente son factores críticos.

Procesamiento de Trabajos en Impresoras: En el ámbito de la impresión las colas son utilizadas para gestionar trabajos de impresión enviados a una impresora los documentos se colocan en una cola de espera y se imprimen secuencialmente esto garantiza que los trabajos se procesen en el orden correcto y facilita la administración de múltiples trabajos de impresión de manera eficiente.

Aplicaciones de colas

Las colas debido a su estructura y funcionamiento basados en el principio FIFO (First In, First Out) tienen una variedad de aplicaciones en múltiples dominios de la informática y otras disciplinas. A continuación, se describen algunas de las aplicaciones más significativas de las colas:

Sistemas Operativos

En los sistemas operativos, las colas son fundamentales para la gestión de procesos y tareas los planificadores de procesos utilizan colas para organizar los trabajos en espera de ejecución optimizando así el uso del procesador y asegurando que los procesos se ejecuten en el orden de llegada las colas de prioridad también se implementan para gestionar tareas críticas que requieren atención inmediata.

Redes de Computadoras

Las colas son esenciales en la administración de tráfico en redes de computadoras los routers y switches emplean colas para almacenar temporalmente paquetes de datos antes de reenviarlos a sus destinos esto permite gestionar eficientemente el tráfico de red minimizar la congestión y mantener la calidad del servicio (QoS) al asegurar que los paquetes se procesen en el orden adecuado.

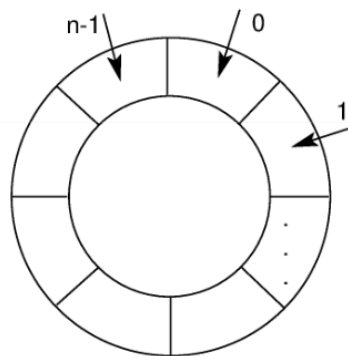
Definición de colas circulares

Las colas circulares son una variación de la estructura de datos de colas que optimiza el uso del espacio al tratar el contenedor de la cola como un espacio cíclico. A diferencia de las colas lineales donde el final de la cola es un punto fijo que se mueve linealmente en una cola circular los extremos de la cola pueden "envolverse" al inicio del contenedor una vez alcanzado el final esto permite una utilización más eficiente de la memoria y evita problemas comunes de desbordamiento en colas estáticas lineales.

"Un vector circular es un vector para el que, desde un punto de vista lógico, consideramos que la posición "siguiente" a la última es la primera (y al contrario). Estos vectores se representan gráficamente de forma circular, de manera que estas dos posiciones aparecen de forma consecutiva." (Garrido Carrillo, (2006)).

"El medio más eficiente, sin embargo, para almacenar una cola en un array, es utilizar un tipo especial de array que junte el extremo final de la cola con su extremo cabeza. Tal array se denomina array circular y permite que el array completo se utilice para almacenar elementos de la cola sin necesidad

de que ningún dato se desplace. Un array circular con n elementos se visualiza en la Fig. 2." (Joyanes Aguilar, (2005)).



Un array circular.

Fig. 2

Importancia de colas circulares

Las colas circulares son una variante eficiente de las colas lineales que optimizan el uso del espacio y mejoran el rendimiento en diversas aplicaciones su estructura cíclica permite que los índices de inicio y final se envuelvan alrededor del contenedor evitando el desperdicio de memoria y asegurando un uso continuo y completo del espacio disponible.

Optimización del Espacio: A diferencia de las colas lineales las colas circulares utilizan el espacio de manera más eficiente eliminando áreas inutilizadas en el contenedor esta característica es crucial en sistemas con recursos limitados donde la optimización de la memoria es esencial.

Rendimiento Constante: Las operaciones de inserción y eliminación en las colas circulares se realizan en tiempo constante $O(1)$ esto garantiza un rendimiento rápido y predecible especialmente importante en aplicaciones de tiempo real y sistemas de alto rendimiento.

Prevención de Errores: La naturaleza cíclica de las colas circulares previene errores de desbordamiento y subdesbordamiento al mantener los índices dentro de los límites del contenedor esto asegura la fiabilidad y estabilidad del sistema.

Aplicaciones de las colas circulares

Las colas circulares encuentran aplicaciones cruciales en diversos dominios debido a su eficiencia y estructura específica en sistemas de comunicación de datos se utilizan como buffers circulares para almacenar flujos continuos de información garantizando una transmisión fluida y sin interrupciones en entornos de tiempo real y sistemas embebidos las colas circulares son esenciales para la gestión eficiente de tareas su capacidad para realizar operaciones de inserción y eliminación en tiempo constante asegura una respuesta rápida a eventos externos, optimizando el uso de recursos y el

tiempo de CPU en sistemas distribuidos las colas de mensajes basadas en colas circulares facilitan la comunicación entre componentes de manera eficiente y desacoplada esto asegura que los mensajes se entreguen en el orden correcto y sin pérdida de datos contribuyendo a la fiabilidad y escalabilidad del sistema. Además, en aplicaciones de procesamiento de datos como edición de video y procesamiento de señales de audio las colas circulares son utilizadas para el almacenamiento temporal garantizando un flujo continuo y suave de información.

Finalmente las colas circulares también son esenciales en la implementación de otras estructuras de datos más complejas como colas de prioridad circulares y buffers de tamaño fijo esto permite optimizar el rendimiento y la gestión de datos en una variedad de aplicaciones críticas.

Comparación entre colas lineales y circulares

Las colas lineales y circulares son estructuras de datos para organizar elementos en orden de llegada. En las colas lineales, los elementos se almacenan en una secuencia lineal mientras que, en las circulares se utilizan contenedores cíclicos las colas circulares usan el espacio de manera más eficiente y permiten operaciones más rápidas pero las colas lineales son más simples y adecuadas cuando no hay restricciones de espacio

Estructura de una cola circular

Una cola circular es una estructura de datos que organiza elementos en un contenedor cíclico donde los elementos se almacenan en una secuencia circular su implementación se basa en dos índices principales: el frente (front) y el trasero (rear) que indican la posición del primer y último elemento de la cola respectivamente.

"Ejemplo de ejecución de operaciones

- 1) *cola_inicializar(q)*
- 2) *cola_insertar(q, 5)*
- 3) *cola_insertar(q, 3)*
- 4) *cola_extraer(q, e)*
- 5) *cola_extraer(q, e)*
- 6) *cola_insertar(q, 7)*
- 7) *cola_insertar(q, 2)*
- 8) *cola_insertar(q, 1)" Ivan Cantador, (2024)*

Operaciones en Colas Circulares

"Una cola permite un conjunto limitado de operaciones que añade un nuevo elemento (Qinsertar) o quita/elimina un elemento (EliminarQ). La clase Cola proporciona también frenteQ, que permite «ver» el primer elemento de la cola.",(Joyanes Aguilar, (2006))

Inserción (Enqueue)

La operación de inserción también conocida como "enqueue", consiste en agregar un nuevo elemento al final de la cola circular para ello se incrementa el índice trasero y se almacena el elemento en la posición correspondiente si el índice trasero alcanza el final del contenedor, se ajusta para envolver al principio del contenedor si es necesario. En caso de que la cola esté llena se produce un desbordamiento y no se puede agregar más elementos hasta que se elimine alguno (Fig.3).

```
void ColaCircular::enqueue(Tipo val) { // Inserción de un elemento
    if (!colaLlena()) {
        vec[final] = val; // Insertar elemento en posición final
        final = (final + 1) % EXT; // Actualizar posición final de manera circular
    }
}
```

Fig. 3

Eliminación (Dequeue)

La operación de eliminación, conocida como "dequeue", consiste en retirar el elemento que se encuentra en el frente de la cola circular. Para ello, se incrementa el índice frente y se devuelve el elemento almacenado en esa posición si el índice frente alcanza el final del contenedor, se ajusta para envolver al principio. Si la cola está vacía (frente y trasero en la misma posición) se produce un subdesbordamiento y no se puede realizar la operación de eliminación (Fig. 4)

```
3
4 Tipo ColaCircular::dequeue() { // Eliminación de un elemento
5     Tipo aux = -1; // Valor por defecto si la cola está vacía
6
7     if (!colaVacía()) {
8         aux = vec[frente]; // Almacenar el valor del elemento a eliminar
9         frente = (frente + 1) % EXT; // Actualizar posición de frente de manera circular
10    }
11    return aux; // Retornar el valor extraído
12 }
```

Fig. 4

Consulta del Elemento en el Frente (Peek):

La operación "peek" permite consultar el elemento que se encuentra en el frente de la cola circular sin eliminarlo esto proporciona acceso al próximo elemento que será procesado sin modificar la estructura de la cola (Fig. 5)

```
5
4 Tipo ColaCircular::peek() { // Consulta del elemento en el frente
5     if (!colaVacía()) {
6         return vec[frente]; // Retornar el valor del elemento en frente si la cola no está vacía
7     }
8     return -1; // Retornar -1 si la cola está vacía
9 }
```

Fig. 5

Verificación de estado (vacía)

La operación "isEmpty" se utiliza para verificar si la cola circular está vacía. Esto se logra comparando los índices de frente y trasero si son iguales la cola está vacía y devuelve verdadero de lo contrario, devuelve falso (Fig. 6)

```
50
51 bool ColaCircular::colaVacía() {           // Verificación de estado (vacía)
52     return frente == final;               // La cola está vacía si frente y final apuntan al mismo lugar
53 }
```

Fig. 6

Verificación de estado (llena)

La operación "isFull" se utiliza para determinar si la cola circular está llena esto se logra comprobando si el índice trasero incrementado en una posición (ajustado por el módulo del tamaño del contenedor) coincide con el índice de frente si esto ocurre, la cola está llena y devuelve verdadero de lo contrario devuelve falso (Fig. 7)

```
bool ColaCircular::colaLlena() {           // Verificación de estado (llena)
    return (final + 1) % EXT == frente;     // La cola está llena si la posición siguiente a final es igual a frente
}
```

Fig. 7

Estas operaciones fundamentales en colas circulares permiten una gestión eficiente de elementos en una secuencia circular garantizando un uso óptimo del espacio y un rendimiento constante en una variedad de aplicaciones.

Implementación de Colas Circulares

En el programa se implementa una cola circular utilizando la clase "Cola Circular" que está definida en el archivo de cabecera "Cola Circular.h" esta clase contiene los métodos necesarios para manejar una cola circular, como la inserción de elementos, eliminación de elementos, consulta del elemento en el frente, y verificación de si la cola está vacía o llena.

Inclusión del archivo de cabecera: Se incluye el archivo "Cola Circular.h" para poder utilizar la implementación de la cola circular en el programa (Fig. 8)

```
#include <iostream>
#include "Cola Circulares.h" // Incluimos la implementación de la cola circular
#define FIN 999999

using namespace std;
```

Fig. 8

Definición de la función main(): En la función principal main(), se crea un objeto de la clase "Cola Circular" que representa la cola circular utilizada en el programa (Fig. 9)


```

int main() {
    system("color f0");

    ColaCircular colaC; // Instanciar el objeto del TDA

    void leerCola(ColaCircular *c); // Prototipo de función ingreso del TDA
    void imprimirCola(ColaCircular c); // Prototipo de la función (la función imprimir no modifica

    leerCola(&colaC); // Llenar la cola con datos ingresados por el usuario

    imprimirCola(colaC); // Imprimir la cola

    return 0;
}

```

Fig. 9

Implementación de las funciones leerCola() e imprimirCola(): Estas funciones se encargan de interactuar con el usuario para leer valores y llenar la cola así como imprimir los elementos de la cola en orden (Fig. 10)

```

void leerCola(ColaCircular *c); // Prototipo de función ingreso del TDA
void imprimirCola(ColaCircular c); // Prototipo de la función (la función

```

Fig. 10

Llamado a las funciones: Finalmente se llaman a las funciones leerCola() e imprimirCola() para interactuar con el usuario y manipular la cola circular (Fig. 10)

```

void leerCola(ColaCircular *c) { // Función ingreso del TDA
    Tipo vl; // Variable de Lectura

    cout << endl << "Ingrese un valor a incluir en el TDA Cola. Finalice con [" << FIN << "]: ";
    cin >> vl;

    while (vl != FIN) { // Usuario aún desea ingresar valores al TDA
        c->enqueue(vl); // Llamado al método de insercción de nuevo elemento
        cout << "Ingrese otro valor o [" << FIN << "] para finalizar: ";
        cin >> vl;
    }
}

void imprimirCola(ColaCircular c) {
    cout << endl << "Elementos en la cola: ";
    while (!c.colaVacia()) { // Aún hay elementos en el TDA
        cout << c.dequeue() << " ";
    }
    cout << endl;
}

```

Fig. 11

Importancia de las colas circulares en informática

Las colas circulares son esenciales en informática debido a su eficiencia en la gestión de datos al aprovechar un contenedor cíclico evitan el desperdicio de espacio y permiten un uso óptimo del almacenamiento disponible esto ofrecen un rendimiento constante realizando operaciones de inserción y eliminación en tiempo constante independientemente del tamaño de la cola esto asegura un comportamiento predecible y rápido en diversas aplicaciones informáticas.

La prevención de desbordamientos es clave en sistemas críticos la estructura cíclica de las colas circulares evita eficazmente errores de desbordamiento garantizando la integridad de los datos su versatilidad es notable desde la gestión de flujos de datos en redes hasta la planificación de procesos en sistemas operativos las colas circulares son indispensables en una variedad de aplicaciones informáticas.

Simplifican la implementación de algoritmos complejos su uso puede reducir la complejidad de ciertas operaciones y mejorar la legibilidad y mantenibilidad del código lo que facilita el desarrollo de software y sistemas informáticos.

Ventajas y Desventajas

Las colas circulares representan una estructura de datos con características distintivas que ofrecen tanto ventajas como desventajas en su aplicación.

A continuación, se detallan de forma formal:

Ventajas

Prevención de Desbordamientos: La naturaleza cíclica de las colas circulares evita eficazmente errores de desbordamiento al envolver los índices dentro del contenedor asegurando un comportamiento robusto y confiable.

Implementación Eficiente: La implementación de colas circulares es relativamente sencilla y eficiente utilizando aritmética modular para gestionar los índices y operaciones en el contenedor cíclico.

"Las colas circulares en C++ permiten un uso eficiente de la memoria al reutilizar espacios liberados. Además, facilitan el acceso a los elementos y minimizan el desplazamiento de datos al agregar o quitar elementos." (My Blog, (2024))

Desventajas

Dificultad en la Capacidad Dinámica: La capacidad fija de las colas circulares determinada por el tamaño del contenedor puede ser una limitación en aplicaciones que requieren una capacidad dinámica y adaptable.

Complejidad Adicional: La naturaleza cíclica de las colas circulares puede introducir complejidad adicional en la implementación y gestión de los índices de frente y trasero en comparación con las colas lineales.

"La implementación de colas circulares en C++ puede complicarse debido a la necesidad de gestionar correctamente el puntero de inicio y fin, lo que puede llevar a errores de lógica y dificultades en el mantenimiento del código" (My Blog, (2024))

Casos de uso recomendados

Sistema de turnos: Imagina un sistema donde las personas llegan a una oficina y toman un número. Estos números se manejan como elementos en una cola circular donde se van asignando y atendiendo en el orden en que llegaron.

Buffer de datos en comunicaciones: Cuando recibes datos a través de una conexión de red o de algún dispositivo de entrada podrías almacenarlos temporalmente en una cola circular antes de procesarlos. Esto te permite manejar los datos en el orden en que llegaron.

Recorrido de elementos en un juego: En un juego donde hay varios objetos o personajes que deben realizar acciones en un determinado orden, puedes usar una cola circular para manejar el orden en que se ejecutan esas acciones.

Implementación de un algoritmo de planificación de procesos: En sistemas operativos, una cola circular podría utilizarse en la implementación de un algoritmo de planificación de procesos como Round Robin, donde los procesos se ejecutan en turnos predeterminados.

Manejo de eventos en un sistema embebido: En un sistema embebido donde se reciben eventos de diferentes sensores o dispositivos, una cola circular puede utilizarse para almacenar temporalmente estos eventos y procesarlos en el orden correcto.

Aplicaciones en la vida real

Las colas circulares, con su capacidad para gestionar eficientemente elementos en una secuencia cíclica, encuentran numerosas aplicaciones en diversos ámbitos de la vida real. A continuación se describen algunas de estas aplicaciones de manera formal:

Sistemas de Comunicación y Redes: En sistemas de comunicación y redes, las colas circulares se utilizan ampliamente para gestionar flujos de datos enrutados como paquetes de red. Los enrutadores y switches de red emplean colas circulares para organizar los paquetes entrantes y salientes, garantizando un flujo de datos eficiente y ordenado.

Sistemas Embebidos y Tiempo Real: En aplicaciones de sistemas embebidos y tiempo real, como controladores industriales y sistemas de automatización, las colas circulares son esenciales para la planificación y gestión de tareas. Estas aplicaciones aprovechan la eficiencia y el rendimiento constante de las colas circulares para asegurar una respuesta rápida y predecible a eventos externos.

Sistemas Operativos y Planificación de Procesos: En sistemas operativos, las colas circulares se utilizan en la planificación de procesos para gestionar la asignación de recursos de manera eficiente. Los planificadores de procesos utilizan colas circulares para organizar los procesos en espera, priorizando su ejecución según políticas predefinidas.

Gestión de Memoria y Almacenamiento: En sistemas informáticos y bases de datos, las colas circulares se emplean para la gestión de memoria y almacenamiento los algoritmos de reemplazo de páginas en sistemas de gestión de memoria virtual utilizan colas circulares para mantener un seguimiento de las páginas en uso y seleccionar las páginas a reemplazar.

Sistemas de Gestión de Colas: En entornos de servicio al cliente y gestión de filas, como bancos, supermercados y centros de atención las colas circulares se utilizan para organizar y gestionar la fila de clientes en espera los sistemas de gestión de colas emplean colas circulares para garantizar un flujo de clientes eficiente y equitativo.

Problemas Comunes y Soluciones

Las colas circulares son una estructura de datos eficiente y versátil, pero pueden presentar algunos problemas comunes en su implementación y uso. A continuación, se describen estos problemas junto con las soluciones correspondientes de manera formal:

Desbordamiento de la Cola (Overflow)

Problema: El desbordamiento ocurre cuando se intenta agregar un elemento a una cola circular llena lo que resulta en la pérdida de datos y un funcionamiento incorrecto del programa.

Solución: Para evitar el desbordamiento se pueden utilizar estrategias como el aumento dinámico del tamaño del contenedor o el uso de técnicas de gestión de la capacidad como el monitoreo del estado de la cola antes de realizar una inserción.

Subdesbordamiento de la Cola (Underflow)

Problema: El subdesbordamiento ocurre cuando se intenta eliminar un elemento de una cola circular vacía lo que puede provocar un error en el programa o un comportamiento inesperado.

Solución: Para evitar el subdesbordamiento se debe verificar el estado de la cola antes de realizar una operación de eliminación asegurándose de que haya elementos disponibles para eliminar.

Pérdida de Referencias (References Loss)

Problema: La pérdida de referencias puede ocurrir cuando se manipulan los índices de la cola de manera incorrecta lo que puede resultar en la corrupción de datos o la pérdida de elementos.

Solución: Es fundamental mantener un seguimiento cuidadoso de los índices de frente y trasero asegurando que se actualicen correctamente después de cada operación de inserción o eliminación. Además se pueden implementar técnicas de control de errores para detectar y corregir posibles problemas de referencia.

Rendimiento Degradado

Problema: El rendimiento de una cola circular puede degradarse si no se implementa de manera eficiente especialmente en operaciones que requieren un gran número de ajustes de índices.

Solución: Para mejorar el rendimiento, se pueden optimizar algoritmos y técnicas de implementación como el uso de aritmética modular para gestionar los índices y la selección de estructuras de datos y algoritmos adecuados para aplicaciones específicas.

Complejidad en Operaciones de Búsqueda y Actualización

Problema: Las operaciones de búsqueda y actualización pueden ser más complejas en colas circulares debido a la naturaleza cíclica de la estructura lo que puede dificultar el acceso eficiente a elementos específicos.

Solución: Para abordar este problema se pueden implementar técnicas de indexación y búsqueda eficientes como el uso de punteros adicionales para marcar elementos específicos o el diseño de algoritmos especializados para operaciones de búsqueda y actualización.

Conclusión

Las colas circulares representan un avance significativo en la gestión de datos, proporcionando una solución eficiente y optimizada para una variedad de aplicaciones en el ámbito de la informática y la programación.

Su capacidad para prevenir desbordamientos su implementación eficiente y su rendimiento constante las convierten en una opción valiosa para diseñadores de sistemas y desarrolladores de software.

Aunque presentan desafíos particulares, como la gestión de índices y la capacidad fija, estas dificultades pueden abordarse con enfoque y conocimiento adecuados.

En última instancia, las colas circulares ofrecen un equilibrio óptimo entre simplicidad y eficiencia impulsando la innovación y la excelencia en el campo de la ciencia de la computación.

Referencia

Joyanes Aguilar, L. (2006). Programación en C++: algoritmos, estructuras de datos y objetos: (2 ed.). Madrid etc, Spain: McGraw-Hill España. Recuperado de <https://elibro.net/es/ereader/epoch/50088?page=709>.

Joyanes Aguilar, L. (2005). Programación en C: metodología, algoritmos y estructura de datos: (2 ed.). Madrid etc, Spain: McGraw-Hill España. Recuperado de <https://elibro.net/es/ereader/epoch/50302?page=677>.

Malik, D. S. (2013). Estructuras de datos con C++: (2 ed.). México, D.F, Mexico: Cengage Learning. Recuperado de <https://elibro.net/es/ereader/epoch/39995?page=487>.

Garrido Carrillo, A. y Valdivia Joaquín, F. (2006). Abstracción y estructura de datos en C++: (ed.). Las Rozas (Madrid), Delta Publicaciones. Recuperado de <https://elibro.net/es/ereader/epoch/167034?page=193>.

cola circular. (s/f). estructura-de-datos. Recuperado el 23 de mayo de 2024, de <https://dariobarrios.wixsite.com/estructura-de-datos/cola-circular->

Colas Circulares Lol. (s/f). Scribd. Recuperado el 23 de mayo de 2024, de <https://es.scribd.com/document/459711265/COLAS-CIRCULARES-LOL>

My Blog, (2024). Implementación de colas circulares en C++: estructuras de datos eficientes. My Blog. <https://ccodigo.com/colas-circulares-c/>

Narváez, B. (s/f). Colas.Cpp.

Ivan Cantador, (2024). • El TADD Cola • Estructura de datos y primitivas de Cola • Estructura de datos de Cola como array circular. Uam.es. Recuperado el 23 de mayo de 2024, de <http://arantxa.ii.uam.es/~cantador/slides/prog2-tema3-Colas.pdf>

Tema 2.5. Colas circulares - Estructura de Datos - Instituto Consorcio Clavijero. (s/f). Edu.mx. Recuperado el 23 de mayo de 2024, de https://cursos.clavijero.edu.mx/cursos/144_ed/modulo2/contenidos/tema2.5.html