Listas enlazadas

Mariana Ariza Combita

Introducción a la Programación

Bogotá.DC, 21 de abril 2025

Conoceremos que es una lista enlazada en tecnología y programación, como las podemos usar correctamente ya sea con los diferentes tipos de datos existentes, donde es más común encontrarlas, como las podemos poner en práctica, tendremos algunos ejemplos para tener de referencia para ponerlas en práctica correctamente y no tener fallos en nuestro código, y que otros conceptos informáticos necesitamos conocer para entender las listas enlazadas

Una lista enlazada es una colección lineal de elementos llamados nodos. El orden entre ellos se establece mediante punteros; direcciones o referencias a otros nodos.

Un nodo está constituido por dos partes: ▪ Un campo INFORMACIÓN: Que será del tipo de los datos que se quiera almacenar en la lista. ▪ Un campo LIGA de tipo puntero, que se utiliza para establecer la liga o el enlace con otro nodo de la lista.

Un programa accede a una lista enlazada mediante un apuntador al primer nodo en la lista. o Y se accede a cada nodo subsiguiente a través del miembro apuntador de enlace almacenado en el nodo anterior. o El apuntador de enlace en el último nodo de una lista se establece en el valor nulo (0) para marcar el final de la lista.

Con una lista enlazada los datos se almacenan en forma dinámica en una lista enlazada; se crea cada nodo según sea necesario. Un nodo puede contener datos de cualquier tipo, incluyendo objetos de otras clases. Son estructuras de datos lineales.

Ventajas de una lista enlazada una lista enlazada es apropiada cuando el número de elementos de datos que se van a representar en un momento dado es impredecible. Las listas enlazadas son dinámicas, por lo que la longitud de una lista puede incrementarse o reducirse, según sea necesario. Las listas enlazadas se llenan sólo cuando el sistema no tiene suficiente memoria para satisfacer las peticiones de asignación dinámica de almacenamiento.

Para mantener las listas enlazadas en orden, se inserta cada nuevo elemento en el punto apropiado en la lista. Los elementos existentes de una lista no necesitan moverse. Los nodos de las listas enlazadas no se almacenan contiguamente en memoria. Sin embargo, en sentido lógico los nodos de una lista enlazada parecen estar contiguos.

Desventajas de las listas enlazadas:

* Acceso aleatorio no eficiente: Para acceder a un elemento específico, se necesita recorrer la lista desde el principio, lo que puede ser ineficiente para accesos aleatorios.
* Mayor consumo de memoria: Se necesita memoria adicional para las referencias entre los nodos.

Aplicaciones:

* Implementación de pilas y colas:

Las listas enlazadas son una forma común de implementar estas estructuras de datos.

* Gestión de memoria:

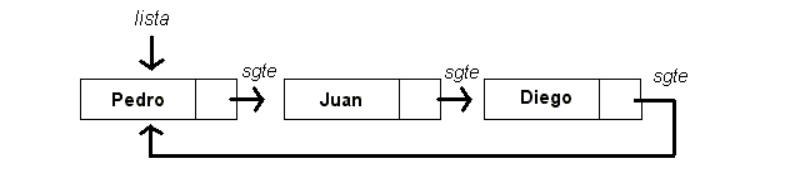
En sistemas operativos y otros sistemas que necesitan gestionar la memoria dinámica, las listas enlazadas son útiles.

* Representación de grafos:

Se pueden usar para representar grafos mediante listas de adyacencia.

* Procesamiento de texto:

En editores de texto y otros programas de procesamiento de texto, las listas enlazadas son útiles para manejar la inserción y eliminación de texto.



Fuente: <https://proyectoestructuradedatos.wordpress.com/wp-content/uploads/2022/11/image-15.png?w=790>}

ASIGNACIÓN DINÁMICA DE MEMORIA Y ESTRUCTURAS DE DATOS EN C++

o Para crear y mantener estructuras dinámicas de datos se requiere la asignación dinámica de memoria.

o Permite que un programa obtenga más memoria en tiempo de ejecución, para almacenar nuevos nodos.

o Cuando el programa ya no necesita la memoria, ésta se puede liberar.

ASIGNACIÓN DINÁMICA DE MEMORIA Y ESTRUCTURAS DE DATOS EN C++

o El operador new recibe como argumento el tipo del objeto que se va a asignar en forma dinámica y devuelve un apuntador a un objeto de ese tipo. Por ejemplo: Nodo \*nuevoPtr = new Nodo( 10 ); // crea un Nodo con el valor 10

o Asigna sizeof( Nodo ) bytes, ejecuta el constructor de Nodo y asigna la dirección del nuevo Nodo a nuevoPtr. o El valor 10 se pasa al constructor de Nodo, el cual inicializa el miembro de datos del Nodo con 10.

o Si no hay memoria disponible: bad\_alloc.

ASIGNACIÓN DINÁMICA DE MEMORIA Y ESTRUCTURAS DE DATOS EN C++

o El operador delete ejecuta el destructor de Nodo y desasigna la memoria asignada con new; esta memoria se devuelve al sistema. delete nuevoPtr;

o nuevoPtr en sí no se elimina; solo se elimina el espacio al que apunta nuevoPtr.

o Si el apuntador nuevoPtr tiene el valor 0 (apuntador nulo), la instrucción no tiene efecto.

ESTRUCTURA DINÁMICA LISTA

Una lista está formada por una serie de elementos llamados nodos los cuales son objetos que contiene como variable miembro un puntero asignado y variables de cualquier tipo para manejar datos. El puntero sirve para enlazar cada nodo con el resto de nodos que conforman la lista. De esto podemos deducir que una lista (lista) es una secuencia de nodos en el que cada nodo esta enlazado o conectado con el siguiente (por medio del puntero mencionado anteriormente). El primer nodo de la lista se denomina cabeza de la lista y el último nodo cola de la lista. Este último nodo suele tener su puntero igualado a NULL Para indicar que es el fin de la lista. En las listas de acuerdo a quién apunte su cabeza y cola y al tipo de nodo que implementa (cuántos punteros tiene) se dividen en cuatro grandes tipos:

Las listas simplemente enlazadas, permiten recorrer la lista en un solo sentido y desde la cabeza hasta la cola.

Las listas doblemente enlazadas, permiten el recorrido en dos direcciones, de la cabeza a la cola y de la cola hacia la cabeza.

Las listas simplemente circulares, permiten el recorrido en una dirección pero al llegar al último nodo (cola) este se encuentra comunicado o enlazado a la cabeza, haciendo un anillo o circulo si se representa gráficamente.

Las listas doblemente circulares, permiten el recorrido en ambas direcciones y la cabeza y cola se encuentran conectadas en ambas direcciones.

LISTA SIMPLEMENTE ENLAZADA: De estos cuatro tipos de lista en esta guía nos enfocaremos en las listas simplemente enlazadas, por ser las más genéricas y las que dan origen a otras estructuras lineales que estudiaremos más adelante. Los nodos de la lista simplemente enlazada tienen la siguiente forma: El dato que contenga un nodo puede ir desde un tipo de dato básico como un entero, decimal o carácter hasta algo más complejo como una estructura completa. Los punteros son los que enlazan al nodo con otros nodos exactamente iguales a él. La lista enlazada es una estructura de datos dinámica cuyos nodos suelen ser normalmente registros y que tienen un tamaño fijo.

Ahora bien, suelen llamarse estructuras dinámicas porque se crean y destruyen según se vayan necesitando. De este modo se solicita o libera memoria en tiempo de ejecución del programa. Gráficamente una lista enlazada puede representarse de la siguiente forma: Las operaciones típicas de la lista incluyen: Crear la lista, verificar si está vacía, insertar elementos, eliminar elementos, mostrar elementos. La lista no tiene ninguna restricción en cuanto a dónde puede eliminar o agregar nodos y es precisamente esta característica lo que la hace la estructura más versátil.

o Antes de analizar la operación Insertar, se creará la clase Nodo y la clase Lista. Clase Nodo: // definicion del nodo de la lista

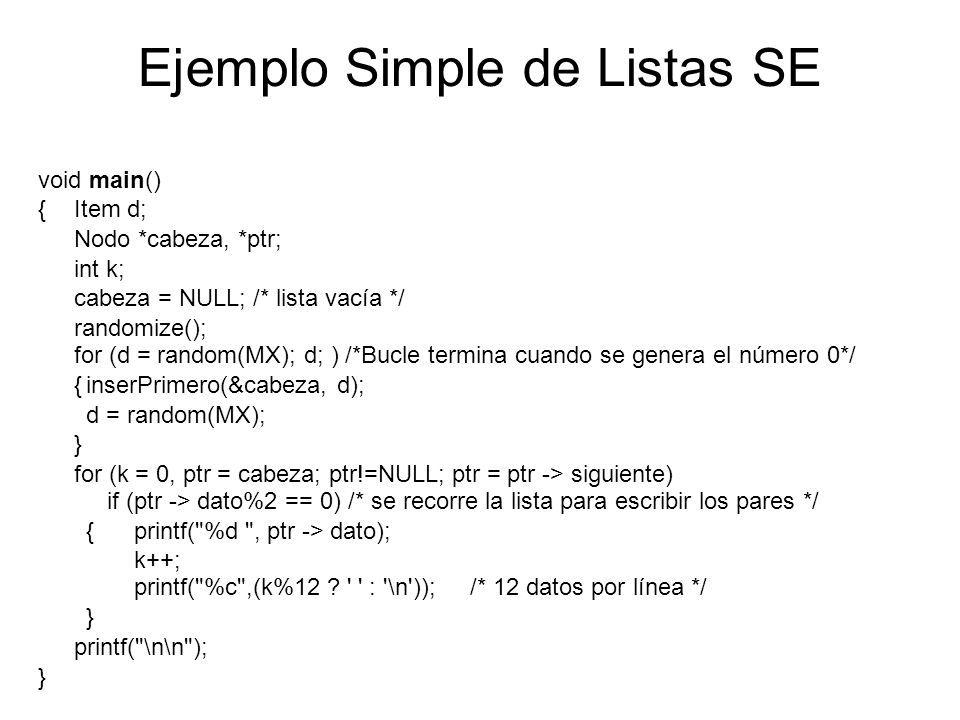
class Nodo

{

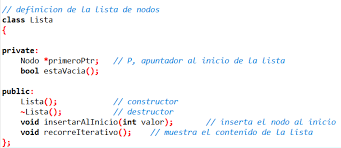
public: int datos; // INFO, variable para almacenar el valor del nodo

Nodo \*siguientePtr; // LIGA, apuntador al nodo siguiente de la lista

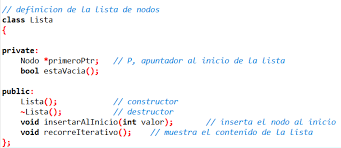
};



Fuente: <https://images.slideplayer.es/3/1088657/slides/slide_7.jpg>



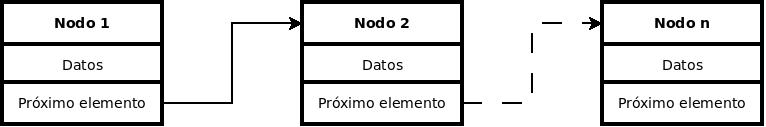
Fuente: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTrOE_YCsCH47Bxt7Lq0o3OtkXSoC5aSeHq6TNewEhf5EgiyeSUGHR7YIkTqvZ1FUex3z4&usqp=CAU>



Fuente: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTrOE_YCsCH47Bxt7Lq0o3OtkXSoC5aSeHq6TNewEhf5EgiyeSUGHR7YIkTqvZ1FUex3z4&usqp=CAU>

Conclusión

En esencia, las listas enlazadas representan un paradigma fundamental en la organización de datos, ofreciendo una alternativa dinámica y flexible a las estructuras de datos estáticas como los arrays. Su naturaleza inherentemente secuencial, definida por las conexiones explícitas entre nodos a través de punteros, les confiere una capacidad única para adaptarse a entornos donde la cantidad de datos y la necesidad de modificaciones frecuentes son impredecibles, La distinción crucial de las listas enlazadas radica en su **gestión de memoria no contigua**. Esta característica elimina la necesidad de pre-asignar bloques de memoria de tamaño fijo, permitiendo que la estructura crezca o se reduzca dinámicamente según las demandas del programa. Esta flexibilidad se traduce en una **eficiencia notable en las operaciones de inserción y eliminación**, especialmente en posiciones arbitrarias dentro de la lista, donde solo es necesario ajustar los punteros de los nodos adyacentes, una operación de costo constante (O(1)) una vez que se localiza el punto de modificac



Fuente: <https://programacionycacharreo.wordpress.com/wp-content/uploads/2018/10/singly_linked_list.jpeg?w=809>

Bibliografía

* <https://www.uv.mx/personal/ermeneses/files/2021/08/Clase5-ListasEnlazadasFinal.pdf>
* <https://proyectoestructuradedatos.wordpress.com/wp-content/uploads/2022/11/image-15.png?w=790>
* <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTrOE_YCsCH47Bxt7Lq0o3OtkXSoC5aSeHq6TNewEhf5EgiyeSUGHR7YIkTqvZ1FUex3z4&usqp=CAU>
* <https://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/programacion-con-estructuras-de-datos/2019/i/guia-3.pdf>