Estructuras de datos

Listas

Participantes:

Romero Morales Gustavo Ismael

Ramírez Zaragoza Emmanuel

Zamora Rodríguez Cristofer Jesús

Velázquez Bobadilla Gibran Alejandro

Yañez Castañeda Adiel Heriberto

Zavalza Peña Yareli Magdalena

Meza Jarquin Hiram Karim

Valdés Gallegos Héctor Geovani

Definición

Es una estructura dinámica de datos que contiene una colección de elementos homogéneos (del mismo tipo) de manera que se establece entre ellos un orden. Es decir, cada elemento, menos el primero, tiene un predecesor, y cada elemento, menos el último, tiene un sucesor. Es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior.

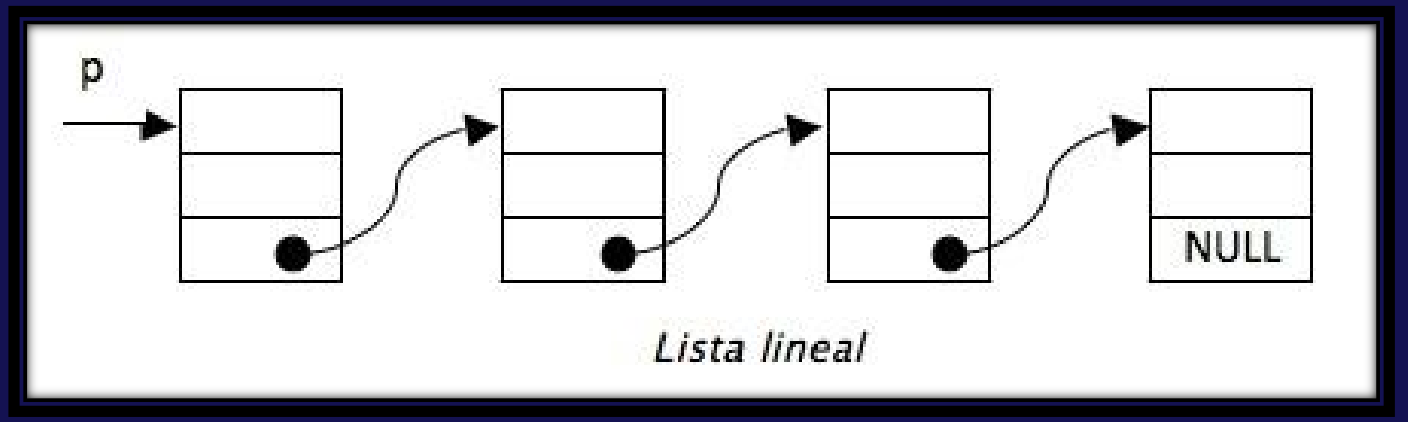
Características.

Una lista es una colección de elementos llamados generalmente nodos, se relacionan por punteros o direcciones a otros nodos.

- Las listas aprueban inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante, pero no permiten un acceso aleatorio.
- Pueden ser implementadas en muchos lenguajes, lenguajes tales como Lisp y Scheme tiene estructuras de datos ya construidas, junto con operaciones para acceder a las listas enlazadas. Lenguaje imperativos u orientados a objetos tales como C o C++ y Java, respectiva, disponen de referencias para crear lisas enlazadas.
- Todos los elementos de la lista son del mismo tipo.
- Existe un orden en los elementos, ya que es una estructura lineal, pero los elementos no están ordenados por su valor sino por la posición en que se han insertado.
- Para cada elemento existe un anterior y un siguiente, excepto para el primero, que no tiene anterior, y para el último, que no tiene siguiente.
- Se puede acceder y eliminar cualquier elemento.

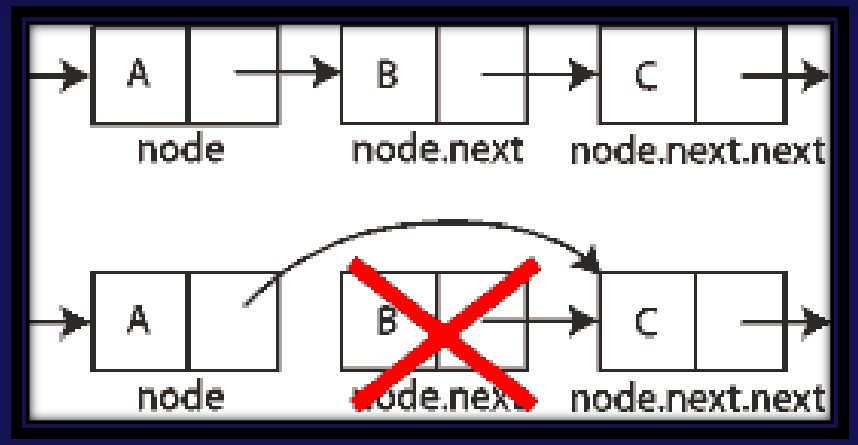
Ventajas

- Son dinámicas, se pueden almacenar en ellas los elementos que sean necesarios siempre y cuando haya espacio suficiente en la computadora.
- Al insertar un elemento, la operación tiene un tiempo constante independientemente de la posición en la que se inserto.
- Al eliminar un elemento sucede el punto anterior.
- No es preciso conocer la cantidad de elementos en tiempo de compilación.
- Las inserciones y eliminaciones no implican realizar corrimientos de los elementos de la lista.



Desventajas

- El acceso a un elemento es mas lento, ya que la información no esta en posiciones contiguas en la memoria de la computadora.
- No se puede acceder a un elemento con base a su posición como se hace en los arreglos.
- Para hacer inserciones o eliminaciones frecuentes, hay que hacer corrimientos costosos.
- No permite el acceso directo a un elemento arbitrario de la lista. Por ejemplo, para llegar al i-ésimo elemento, se debe recorrer la lista, iniciando con el primer nodo hasta llegar al elemento deseado.

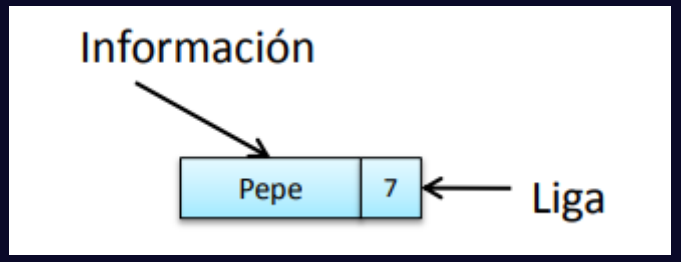


Representación en Memoria:

- La Lista es una estructura dinámica, donde el numero de nodos en una lista puede varias a medida que los elementos insertados y removidos, el orden entre estos establece por medio de un tipo de datos denominados punteros, direcciones o referencias a otros nodos, es por esto que la naturaleza dinámica de una lista contrasta con un arreglo que permanece en forma constante.
- La asignación de memoria puede hacerse en tiempo de compilación y los objetos están vigentes desde que comienza la ejecución del programa hasta que termina. Estos Registros de activación contendrá las variables locales, parámetros formales y valor devuelto por la función.

Consideraciones:

- ✓ Error en tiempo de ejecución de índice fuera del rango.
- ✓ Se debe conocer con anticipación el tamaño de la estructura.
- ✓ ü Se guardan en memorias adyacentes.
- ✓ Vectores, matrices, cubos, registros, archivos.



Aplicaciones y usos

El uso y aplicaciones que tienen estas estructuras de datos son muy amplias, tanto así que según el tipo de lista que usemos serán las aplicaciones a efectuar. A modo general podemos desglosar los siguientes usos y aplicaciones.

Listas sencillas:

- Cronogramas de actividades
- Hallar elementos de una fila de personas
- Listas de compras

Listas dobles:

- Medidor de combustible de un carro
- Almacenar información para un carrusel de imágenes
- Almacenar información para botones adelante y atrás

Listas circulares:

- Canales de televisión
- Simulaciones cíclicas como el clima
- Funcionamiento de un reloj

Operaciones Básicas

Insertar:

- La operación insertar consiste en la introducción de un nuevo elemento en la lista.

Borrar:

- La operación borrar consiste en la eliminación de la lista de un elemento concreto. El elemento a borrar será escogido por el programador.

Tamaño:

- Tamaño: Esta operación suele informar sobre el número de elementos que tiene en ese instante la lista.

Buscar:

- Comprueba si existe un determinado elemento en la lista.

Recorrer lista:

- Recorre toda la lista, realizando una operación en cada nodo. Por ejemplo, mostrar el contenido por pantalla.

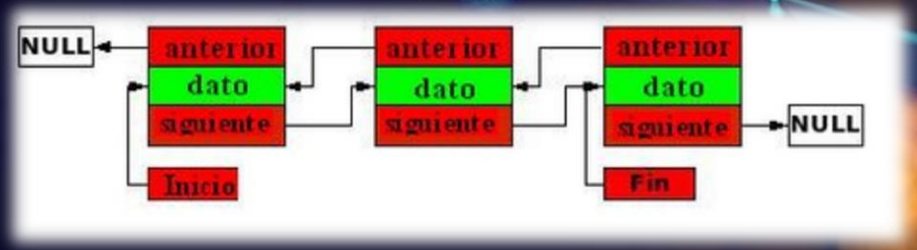
Tipos

Listas Simplemente Enlazada:

- Cada nodo de la estructura tiene un único campo de enlace que apunta al siguiente nodo en la lista. El ultimo nodo en la lista apunta NULL (vacío).

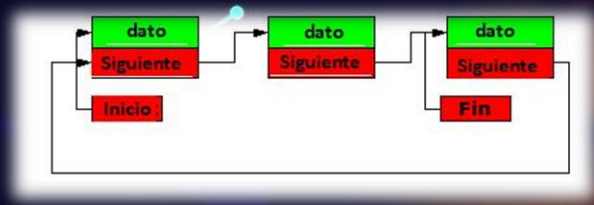
Listas Doblemente Enlazada:

- Cada nodo de la estructura tiene un campo de enlace que apunta al siguiente nodo en la lista y un campo de enlace que apunta al nodo anterior de la lista. El ultimo y primer nodo en la lista apunta a NULL.



Listas Circulares Simple:

- Es una lista simple en la que el ultimo nodo apunta al primero en lugar de a NULL.



Listas Circulares Doble

- Es una lista enlazada doble en la que el ultimo nodo apunta al primero y el primero nodo apunta al ultimo.