

ESTRUCTURAS LINEALES PARA DESARROLLAR ALGORITMOS QUE BRINDAN SOLUCIÓN ÓPTIMA A PROBLEMAS

[Elemento competencia 3](#)[Guía de evidencia](#)[Bibliografía](#)

Desarrollo temático

Tabla de contenido

De clic en los siguientes enlaces para acceder a un contenido específico.

[Estructuras de Datos dinámicas](#)

Estructuras de Datos dinámicas

A diferencia de lo estudiado anteriormente sobre vectores y matrices, en las cuales el tamaño en memoria es definido durante la compilación del programa y es invariable en la ejecución, las estructuras de datos dinámicas pueden crecer y disminuir con la ejecución del programa.

En las listas enlazadas encontramos una colección de datos dispuestos en fila (uno tras de otro), los cuales se conectan por medio de un enlace o puntero, las aplicaciones de las listas son variadas y muy flexibles.

Fundamentos

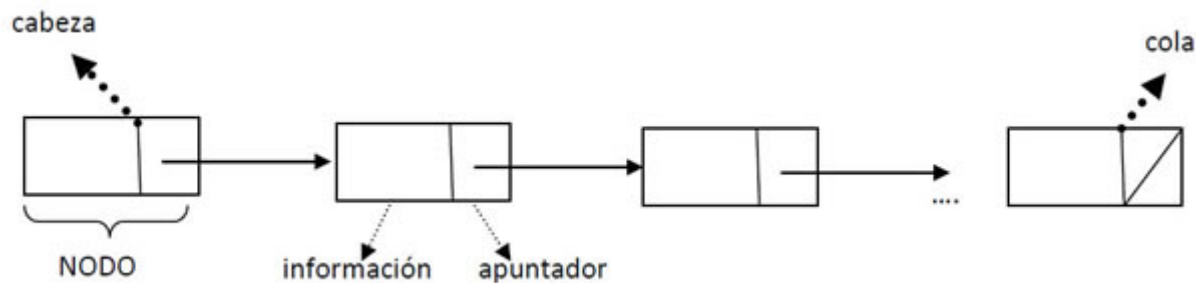
En las unidades anteriores estudiamos que los vectores y matrices se deben definir desde un comienzo con un tamaño específico en memoria, haciendo así un uso ineficiente de memoria.

La asignación dinámica de variables se puede utilizar en listas de tal manera que la memoria física que se utilice corresponda con el número de la tabla.

Conceptos básicos de las listas

- ☀ Punteros: también son llamados apuntadores, permiten utilizar la memoria de manera mas eficiente
- ☀ Lista enlazada: Secuencia de elementos uno tras de otro (nodos), los cuales se conectan por medio de un puntero.

- ☀ **Nodo:** se compone de dos partes la primera contiene la información y es un valor genérico (dato, tipoelemento, info, etc) y la segunda es el puntero (que como ya lo vimos es el enlace), este último apunta al siguiente elemento de la lista.



Ejemplo

Nos solicitan crear un listado para almacenar el nombre, apellido y día de nacimiento de los niños de una institución educativa.

Para este caso tenemos una estructura que cuenta con los siguientes elementos

Dato: Nombre, apellido y día de nacimiento

Nombre y apellido de tipo carácter

día de nacimiento de tipo fecha (algunos programas no cuentan con este tipo de dato por lo cual se deben crear como tipo entero).

La definición para esta lista enlazada será

Nodo Ninos

Cadena: Nombres, apellidos

Fecha: día de nacimiento

Puntero: Enlace

Gráficamente sería

NINOS

Nombres Apellidos Dia_nacimiento	Enlace
--	--------

Sintaxis: Crear (Nombre_Lista)

Para nuestro caso Crear (Ninos)

Para eliminar un nodo de la lista, el primer paso es ubicarnos en el nodo que se desea eliminar y utilizar la siguiente sintáxis

Eliminar(Nombre_Lista.Enlace).

Si deseamos eliminar toda la lista utilizamos

Eliminar (Nombre_Lista)

Listas enlazadas y su clasificación

De acuerdo con Joyanes, L. (2001) las listas enlazadas se pueden clasificar de la siguiente manera:

- ☀ **Listas simplemente enlazadas.** Cada nodo (elemento) contiene un único enlace que conecta ese nodo al nodo siguiente o nodo sucesor. La lista es eficiente en recorridos

directos («adelante»).

- ✿ *Listas doblemente enlazadas*. Cada nodo contiene dos enlaces, uno a su nodo predecesor y el otro a su nodo sucesor. La lista es eficiente tanto en recorrido directo («adelante») como en recorrido inverso («atrás»).
- ✿ *Lista circular simplemente enlazada*. Una lista enlazada simplemente en la que el último elemento (cola) se enlaza al primer elemento (cabeza) de tal modo que la lista puede ser recorrida de modo circular («en anillo»).
- ✿ *Lista circular doblemente enlazada*. Una lista doblemente enlazada en la **que** el último elemento se enlaza al primer elemento y viceversa. Esta lista se puede recorrer de modo circular (en anillo) tanto en dirección directa («adelante») como inversa («atrás»).

Operaciones en listas enlazadas

Para realizar la gestión en los elementos de contenido de las listas se deben tener controles los cuales se realizan por medio de las siguientes operaciones que se hacen en las listas:

1. Declaración de los tipos nodo y puntero a nodo
2. Inicialización o creación
3. Insertar elementos en una lista
4. Eliminar elementos de una lista
5. Buscar elementos en una lista
6. Recorrer una lista enlazada
7. Comprobar si la lista se encuentra vacía

Para mayor información sobre esta temática indaga en el siguiente enlace apartado *Listas Enlazadas*, en el cual encuentras definición de listas enlazadas, ejemplos y realización de las operaciones básicas con ellas <http://robotica.uv.es/pub/Libro/PDFs/CAP15.pdf>

De igual manera el siguiente video te permite conocer los fundamentos de las listas enlazadas y su creación y utilización

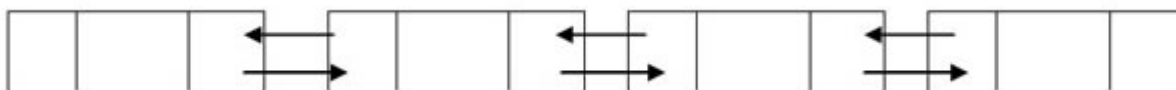
Tomado de: <http://youtu.be/l9XHUrwDdVo>

Listas doblemente enlazadas

Se crean para recorrer la lista hacia adelante y hacia atrás o dado un elemento conocer rápidamente el elemento anterior y siguiente. Para esto se asigna a cada celda sobre una lista un puntero a la celda siguiente y anterior a la lista

Otra gran ventaja es que podemos usar un puntero a la celda que contiene el i -ésimo elemento dentro de una lista para representar la posición i , de esta manera se mejora el puntero a la celda anterior aunque es posible hacerlo de la manera como se trabaja en listas estudiadas anteriormente .

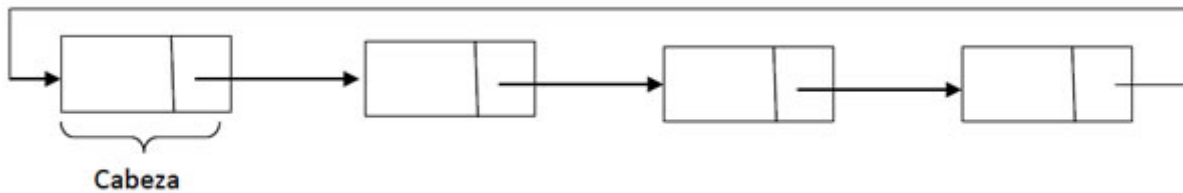
La característica de estas listas es la creación de un puntero adicional en cada celda y la realización de procedimientos algo mas extensos para algunas operaciones básicas antes habladas.



Para fortalecer esta temática te invito a explorar la información presentada en el siguiente vínculo, allí encontrarás datos generales, procedimientos para el desarrollo de operaciones en ellas <http://decsai.ugr.es/~jfv/ed1/tedi/cdrom/docs/ldoble.html>

Listas circulares

Es una lista lineal la cual en el nodo final apunta al primero. La gran ventaja de estas listas es que evitan excepciones en las operaciones que se realicen sobre ellas. Los nodos siempre cuentan con uno anterior y uno siguiente.



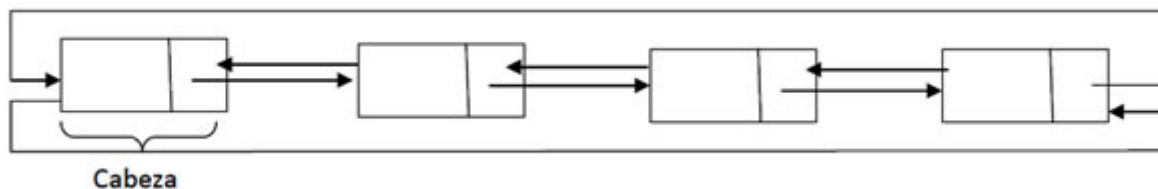
Una de las complicaciones mas grande que surge con este tipo de listas es que al momento de realizar un búsqueda, no es tan sencillo finalizarla cuando el elemento que se está buscando no existe en la lista.

Para evitar casos especiales se tiene un nodo de cabecera, de esta manera se evita también que la lista sea vacía. Otra solución es resaltar cualquier nodo dentro de la lista el cual puede variar en la ejecución y no siempre es el mismo.

Para conocer mas sobre este tema te encuentras invitado a ingresar en el sitio C++ con clase en el enlace que se presenta a continuación, en el cual encuentras la definición de este tipo de listas, tipos operaciones básicas y ejemplos de su desarrollo.
<http://c.conclase.net/edd/?cap=004#inicio>

Listas circulares doblemente enlazadas

Este tipo de listas tiene las características de listas circulares y de listas enlazadas. Los nodos en estas listas poseen siempre un predecesor y un sucesor.



Para fortalecer la temática indaga en el siguiente vínculo, encuentras información sobre su definición y ejemplos de este tipo de listas
http://artemisa.unicauca.edu.co/~mendoza/Laboratorio6_LabEdatos1.htm

Con el estudio de las listas, damos por terminado el curso de Algoritmos y Estructuras de datos I. Con el estudio de los contenidos y de las actividades, desarrollaste la competencia de crear algoritmos utilizando vectores, matrices, listas y garantizando un manejo eficiente de los datos para dar soluciones óptimas a situaciones planteadas. Realiza las actividades, entrega las evidencias de tus aprendizajes, y disponte a seguir profundizando en tus estudios en el siguiente curso de Algoritmos y Estructuras de datos II. Exitos.

