

Estructura de Datos

TDA LISTA

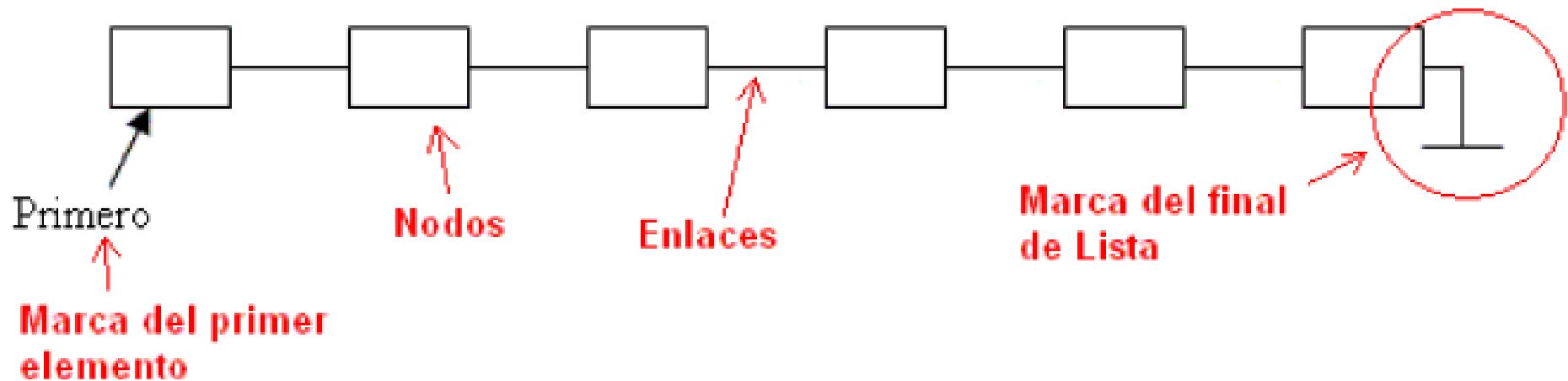
El TDA Lista

El TDA Lista es una estructura compuesta por nodos (en los cuales se encuentran los datos almacenados) los cuales se encuentran enlazados entre si uno a uno en forma lineal.

Las formas de implementarla son muchas dependiendo de las necesidades; ellas varían en el tipo de enlace, implementación de primitivas, capas de abstracción adicionales, etc.



Como un factor común de las listas cabe mencionarse que siempre, en toda lista, existe un elemento “marcado” como el primer elemento y, además, luego del último elemento de la lista se encuentra una “marca de final de lista”, como se ve en la siguiente figura.



Si de alguna manera se quiere ir del nodo de la posición “i” al nodo de la posición “j” de esta lista inevitablemente se debe pasar por todos los nodos intermedios entre el nodo “i” y el nodo “j”, con lo cual se dice que una lista (sea cual sea su implementación) es una estructura de **recorrido secuencial**.

Ahora bien, este recorrido puede ser de derecha a izquierda (recorrido en sentido progresivo) o de izquierda a derecha (recorrido en sentido regresivo). El tipo de recorrido de una lista esta dado por el tipo de enlace que ella tiene.



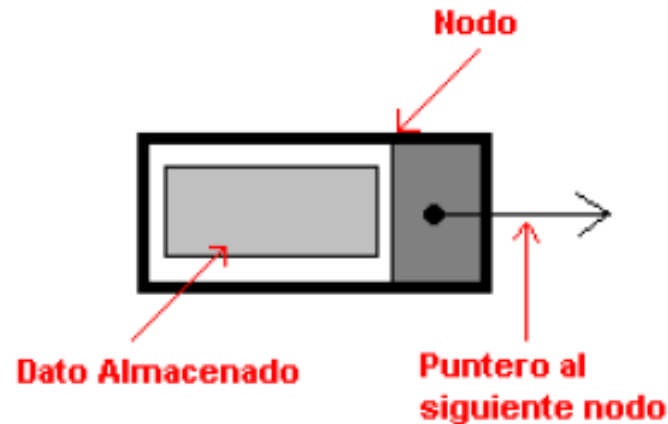
Una cosa para destacar es que cualquiera puede confundir a una TDA lista con un simple arreglo de elementos. Lo primero que se tiene que apreciar es que un arreglo de elementos es un tipo de implementación de Lista mientras que un TDA Lista es algo más que una simple Estructura

Pero nunca hay que perder de lado que los arreglos pueden ser tranquilamente un tipo de implementación de un TDA Lista o como cualquiera puede proponer, según las necesidades de su uso.



Lista Simplemente Enlazada

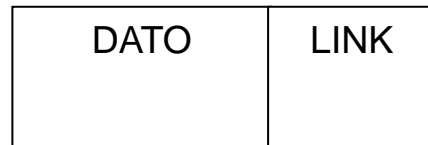
El nodo de esta lista esta formado por el dato a almacenar y un puntero a otro “*NodoLista*” que es el enlace simple (o, tambien conocido como puntero al siguiente) de este nodo



Una lista enlazada o encadenada es una colección de elementos ó nodos, en donde cada uno contiene datos y un enlace o liga.

Un **nodo** es una secuencia de caracteres en memoria dividida en campos (de cualquier tipo). Un nodo siempre contiene la dirección de memoria del siguiente nodo de información si este existe.

Un **apuntador** es la dirección de memoria de un nodo



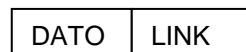
El campo liga, que es de tipo puntero, es el que se usa para establecer la liga con el siguiente nodo de la lista. Si el nodo fuera el último, este campo recibe como valor NIL (vacío).



Una lista enlazada o encadenada es una colección de elementos ó nodos, en donde cada uno contiene datos y un enlace o liga.

Un **nodo** es una secuencia de caracteres en memoria dividida en campos (de cualquier tipo). Un nodo siempre contiene la dirección de memoria del siguiente nodo de información si este existe.

Un **apuntador** es la dirección de memoria de un nodo



El campo liga, que es de tipo puntero, es el que se usa para establecer la liga con el siguiente nodo de la lista. Si el nodo fuera el último, este campo recibe como valor NIL (vacío).



Algoritmo

CREAINICIO(P)

{Este algoritmo crea una lista, agregando cada nuevo nodo al inicio de la misma}
{ P y Q son variables de tipo puntero. P apuntará al inicio de la lista}

1. CREA (P) {Crea el primer nodo de la lista}
2. Leer P->INFORMACIÓN
3. Hacer P->LIGA=NIL
4. Repetir
 - CREA (Q)
 - Leer Q->INFORMACIÓN
 - Hacer Q->LIGA= P y P = Q
5. Hasta (que ya no haya información)



```
#include <stdio.h>
#include <stdlib.h>
```

```
struct nodo {
    char nombre[50];
    float calif;
    struct nodo *sig;
};
```

```
typedef struct nodo NODO;
```



```
int main (){
    NODO *nodo inicial, *nodo anterior, *nuevo nodo;
    float aux;
    char op='s';
    nodo inicial=NULL;
    do {
        nuevo nodo=(NODO *)malloc(sizeof(NODO));
        printf("\nNombre del alumno: ");
        gets(nuevo nodo->nombre);
        printf("Calificacion: ");
        scanf("%f",&aux);
        nuevo nodo->calif=aux;
        nuevo nodo->sig=NULL;
    }
```



```
if (nodoInicial == NULL) {
    nodoInicial=nuevoNodo;
    nodoAnterior=nuevoNodo;
}
else {
    nodoAnterior->sig=nuevoNodo;
    nodoAnterior=nuevoNodo;
}
printf("\nDeseas ingresar los datos de otro
alumno? (s/n): ");
getchar();
op = getchar();
op = tolower(op);
getchar();
} while (op == 's');
```



```
/*Impresion de informacion*/
printf("\n");
nuevonodo=nodoinicial;
while(nuevonodo != NULL) {
    printf("\nNombre: %s", nuevonodo->nombre);
    printf("\nCalificacion: %.2f", nuevonodo->calif);
    nuevonodo=nuevonodo->sig;
    printf("\n");
}
printf("\n");
system("pausa");
}
```

