



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ESTRUCTURAS DE DATOS

LISTAS DOBLEMENTE ENLAZADAS

PRÁCTICA 5

**HERNÁNDEZ CASTELLANOS CÉSAR URIEL
MARTÍNEZ ISLAS MAURICIO JOEL**

1CV7

6/04/2017

BENJAMIN LUNA BENOSO



Índice

Resumen.....	3
Introducción.....	3
Experimentación y resultados.	6
Pseudocódigo	9
Conclusiones.....	14
Referencias.	15

Resumen.

En el presente reporte se muestra la documentación de la práctica número cinco, cuya función principal es la siguiente:

Llevar un control de las temperaturas diarias en la ciudad, el cual contendrá las siguientes opciones:

- 1) Insertar en forma ordenada con base en la fecha.
- 2) Mostrar los registros ordenados de menor a mayor temperatura.
- 3) Mostrar la temperatura más alta de cierto día y la hora.
- 4) Mostrar la temperatura más baja de cierto día y la hora.
- 5) Eliminar un registro 'x'

Introducción.

Las listas enlazadas fueron desarrolladas en 1956 por Cliff Shaw y Herbert Simon en RAND Corporation como una estructura de datos de su lenguaje de procesamiento de la información (IPL), las listas fueron usados por los autores para el desarrollo de distintos programas de inteligencia artificial

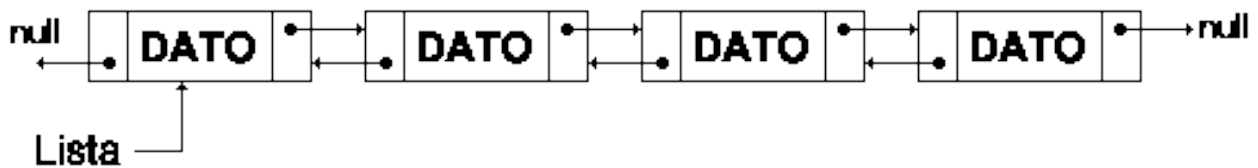
Las listas enlazadas se tratan de estructuras de datos similares a los array, sin embargo el acceso en las listas enlazadas se realiza con un puntero y no con un índice, la asignación de la memoria se realiza durante la ejecución del programa.

En cambio, mientras que en un array los elementos están de manera secuencial en la memoria en una lista se encuentran dispersos, el enlace entre los nodos se realiza por un puntero.

El puntero siguiente del último elemento apunte hacia NULL, lo que indica el final de la lista.

Tipos de listas enlazadas:

- Listas simplemente enlazadas.
- Listas doblemente enlazadas.
- Listas enlazadas circulares.
- Listas enlazadas simples circulares.
- Listas enlazadas doblemente circulares.



Lista doblemente enlazada.

Para poder acceder a un elemento específico de la lista, la lista es recorrida comenzando por el inicio, el puntero siguiente permite el cambio hacia el próximo elemento.

El desplazamiento únicamente se realiza en una dirección del primero al último elemento, esto para las LSE, en cambio si queremos recorrer la lista hacia delante y hacia atrás tendremos que recurrir a las listas doblemente enlazadas.

Aplicaciones de las listas enlazadas.

Las listas enlazadas se usan como módulos para otras estructuras de datos, como lo son las pilas, colas y sus variaciones.

En algunas ocasiones, las listas enlazadas se usan para implementarse con árboles binarios de búsqueda equilibrados, Sin embargo, algunas veces la lista enlazada es dinámicamente creada fuera de un subconjunto propio de nodos semejante a un árbol, y son más eficientes para recorrer diversos datos.

Ventajas de las listas enlazadas respecto a los arrays.

1- Tamaño dinámico: Lo que implica optimización de la memoria.

Desventajas de las listas enlazadas respecto a los arrays.

1- El acceso es secuencial (para llegar a una posición deberemos pasar por todas las anteriores). Esto significa lentitud. Imaginad por un momento el tinglado que tendríamos que montar para ordenar una lista enlazada. Si buscamos un elemento de una lista ordenada también tardaremos más, no vale la búsqueda dicotómica que vimos en la Ampliación 1 de C (métodos de clasificación en memoria dinámica).

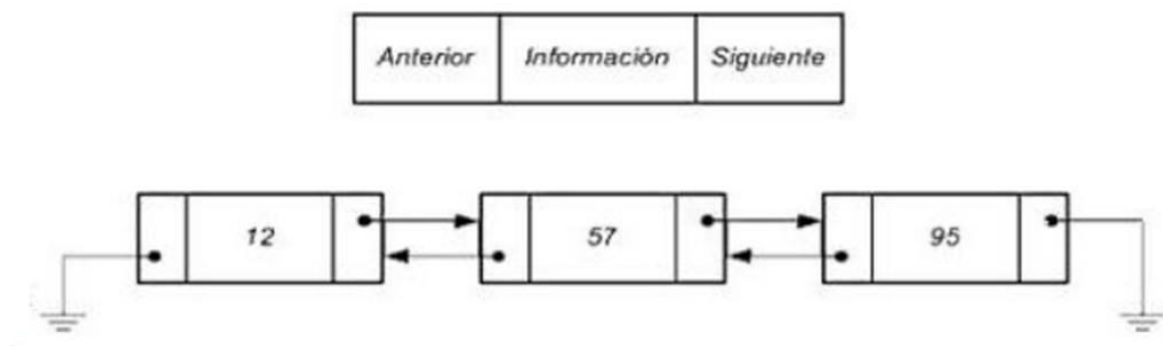
2- El código se complica.

Listas doblemente enlazadas.

Las listas doblemente enlazadas permiten recorridos en ambos sentidos gracias al almacenamiento de dos referencias por nodo. Una referencia al siguiente nodo de la lista y otro al anterior. Para terminar de hacer simétrica la representación, añadiríamos además un nodo final además del nodo cabecera.

La simetría exige que usemos un nodo cabecera y un nodo final, y que, para sacar partido de ella, proporcionemos el doble de operaciones.

Cuando avanzamos más allá del final de la lista, ahora llegamos al nodo final en lugar de a null.



Lista doblemente enlazada.

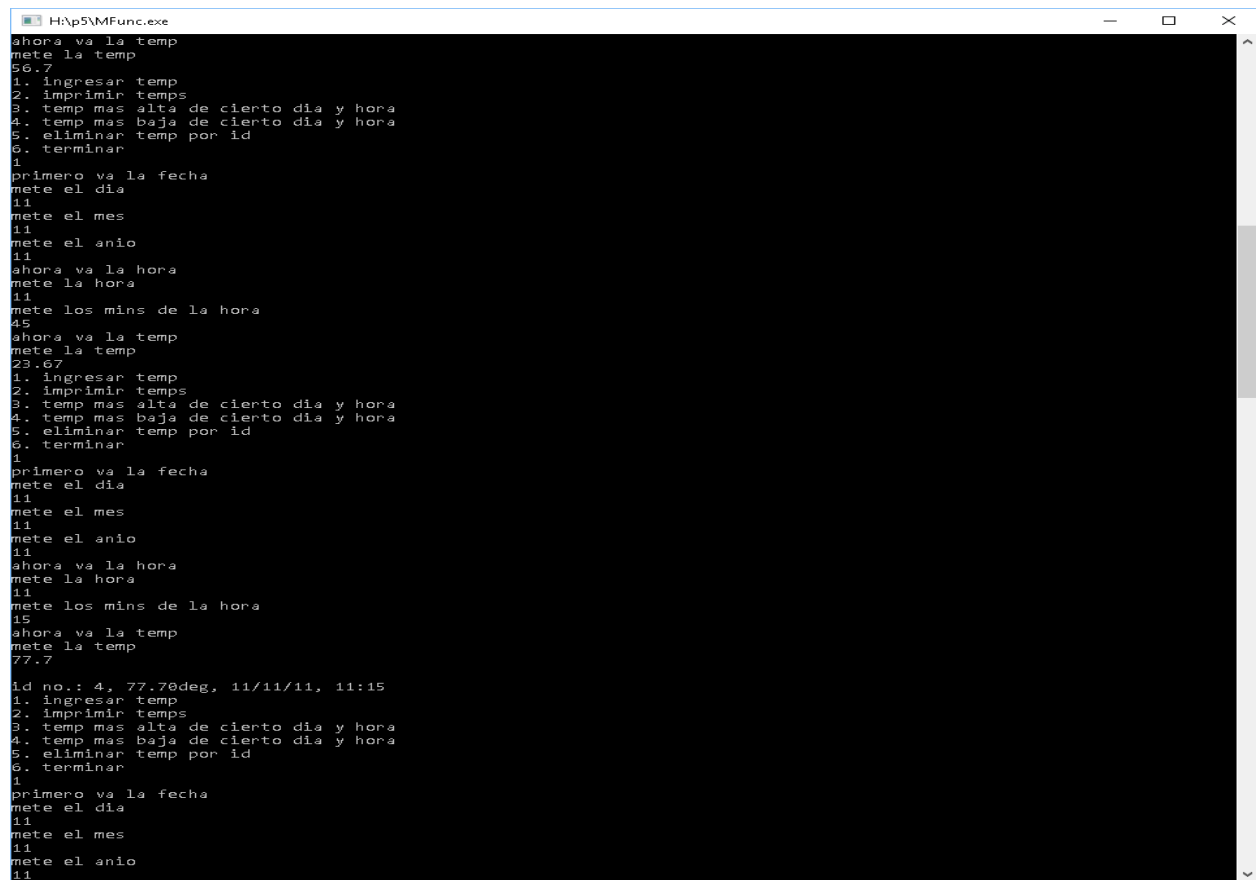
Características:

- Recorrido secuencial en ambas direcciones
- Mayor ocupación: 2 referencias en cada nodo
- Inserción y borrado: Modificar más referencias

Experimentación y resultados.

La práctica consiste en llevar a cabo una aplicación en el que se haga uso de listas doblemente enlazadas, empleando las siguientes opciones en el menú.

- 1) Insertar en forma ordenada con base en la fecha.
- 2) Mostrar los registros ordenados de menor a mayor temperatura.
- 3) Mostrar la temperatura más alta de cierto día y la hora.
- 4) Mostrar la temperatura más baja de cierto día y la hora.
- 5) Eliminar un registro 'x'



```
H:\p5\MFunc.exe
ahora va la temp
mete la temp
56.7
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
1
primero va la fecha
mete el dia
11
mete el mes
11
mete el anio
11
ahora va la hora
mete la hora
11
mete los mins de la hora
15
ahora va la temp
mete la temp
77.7
id no.: 4, 77.70deg, 11/11/11, 11:15
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
1
primero va la fecha
mete el dia
11
mete el mes
11
mete el anio
11
```

Algunas inserciones

```
H:\p5\MFunc.exe
mete los mins de la hora
55
ahora va la temp
mete la temp
55.77
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
2

id no.: 5, 55.77deg, 11/11/11, 11:55
id no.: 3, 23.67deg, 11/11/11, 11:45
id no.: 4, 77.70deg, 11/11/11, 11:15
id no.: 2, 56.70deg, 11/11/11, 11:12
id no.: 0, 45.60deg, 11/11/11, 11:1
id no.: 1, 34.20deg, 11/11/11, 1:11

1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
```

Tabla de registros

```
H:\p5\MFunc.exe
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
3
mete el anio
11
mete el mes
11
mete el dia
11
mete la hora
11
temperatura maxima: 77.700
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
```

Máxima temperatura

```
H:\p5\MFunc.exe
temperatura maxima: 77.700
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar

4

mete el anio
11
mete el mes
11
mete el dia
11
mete la hora
11
temperatura minima: 23.670
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
```

Mínima temperatura

```
H:\p5\MFunc.exe
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
5
mete el id que vas a borrar
0
1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. terminar
2

id no.: 5, 55.77deg, 11/11/11, 11:55
id no.: 3, 23.67deg, 11/11/11, 11:45
id no.: 4, 77.70deg, 11/11/11, 11:15
id no.: 2, 56.70deg, 11/11/11, 11:12
id no.: 1, 34.20deg, 11/11/11, 1:11

1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
```

Borrando registro


```
H:\p5\MFunc.exe
id no.: 4, 23.66deg, 11/11/11, 11:34
id no.: 3, 23.50deg, 11/11/11, 11:5
id no.: 2, 34.70deg, 11/11/11, 11:3
id no.: 1, 45.00deg, 11/11/11, 11:2
id no.: 0, 12.00deg, 11/11/11, 11:1

1. ingresar temp
2. imprimir temps
3. temp mas alta de cierto dia y hora
4. temp mas baja de cierto dia y hora
5. eliminar temp por id
6. ver por temperatura
7. terminar
6
COPY id no.: 4, 23.66deg, 11/11/11, 11:34
COPY id no.: 3, 23.50deg, 11/11/11, 11:5
COPY id no.: 2, 34.70deg, 11/11/11, 11:3
COPY id no.: 1, 45.00deg, 11/11/11, 11:2
COPY id no.: 0, 12.00deg, 11/11/11, 11:1

id no.: 0, 12.00deg, 11/11/11, 11:1
id no.: 3, 23.50deg, 11/11/11, 11:5
id no.: 4, 23.66deg, 11/11/11, 11:34
id no.: 2, 34.70deg, 11/11/11, 11:3
id no.: 1, 45.00deg, 11/11/11, 11:2

1. ingresar temp
2. imprimir temps
```

Ver por temperatura (con un nuevo set de registros)

Pseudocódigo

Funcion imprimirTemps (NODO* cab)

NODO* aux

Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer

imprimir aux->dato

FinPara

Fin Funcion

Funcion eliminarReg (NODO** cab, int idDel)

NODO* aux = NULL

Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer

si aux->dato->id = idDel Entonces

si aux->prev y aux->sig Entonces

aux->prev->sig = aux->sig

aux->sig->prev = aux->prev

FinSi

si aux->prev y !aux->sig Entonces

aux->prev->sig = NULL

aux->prev = NULL

FinSi

si aux==*cab y aux->sig Entonces

*cab = aux->sig;

aux->sig->prev = NULL;

aux->sig = NULL;

FinSi

si aux==*cab y !aux->sig Entonces

*cab = NULL;

FinSi

FinSi

FinPara

Fin Funcion

Funcion verTempMax (NODO* cab)

NODO* aux

int a, m, d, h

double minTemp

```

    leer a, m, d, h

    Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer
        si aux->dato->f->ano= y aux->dato->f->mes=m y aux->dato->f->dia
        =d y aux->dato->h->hr = h Entonces
            minTemp = aux->dato->temp
        FinSi
    Fin Para

    Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer
        si aux->dato->f->ano= y aux->dato->f->mes=m y aux->dato->f->dia
        =d y aux->dato->h->hr = h Entonces
            si aux->dato->temp<=minTemp entonces
                minTemp = aux->dato->temp
            FinSi
        FinSi
    Fin Para

Fin Funcion

```

Funcion verTempMin (NODO* cab)

```

    NODO* aux
    int a, m, d, h
    double maxTemp
    leer a, m, d, h

    Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer
        si aux->dato->f->ano= y aux->dato->f->mes=m y aux->dato->f->dia
        =d y aux->dato->h->hr = h Entonces
            maxTemp = aux->dato->temp
        FinSi
    Fin Para

```

```

        FinSi
    Fin Para
    Para aux<-cab Hasta !aux Con Paso aux=aux->sig Hacer
        si aux->dato->f->ano= y aux->dato->f->mes=m y aux->dato->f->dia
        =d y aux->dato->h->hr = h Entonces
            si aux->dato->temp>=maxTemp entonces
                maxTemp = aux->dato->temp
            FinSi
        FinSi
    Fin Para
Fin Funcion

Funcion verPorTemperatura( NODO* cab )
    NODO* copia = copia de cab
    bubbleSortPorTemperatura(copia)
    imprimir copia
Fin Funcion

Funcion insertarPorFecha ( item x, NODO** cab )
    NODO* aux = *cab
    NODO* n
    int gt=0, nLvl, cLvl=0, stop=0
    Si aux Entonces
        mientras stop = 0 Hacer
            si x > aux->dato Entonces

```

```

        gt=1
        stop=1
    FinSi
    si stop <> 1 Entonces
        si !aux->sig Entonces
            stop=1
        FinSi
        si aux->sig y stop=0 Entonces
            aux=aux->sig
        FinSi
    FinSi
FinMientras
si gt = 1 entonces
    si aux->prev Entonces
        insertar en aux->prev+
    sino
        insertar por cabecera
    FinSi
Sino
    insertar en aux
FinSi
Sino
    insertar por cabecera
FinSi
Fin Funcion

```

Conclusiones.

Hernández Castellanos César Uriel.

Como conclusión de la presente práctica sobre listas como estructuras de datos, es de vital importancia mencionar que las listas son una de las estructuras de datos fundamentales, y que pueden ser utilizadas para la implementación de otras estructuras de datos. La principal ventaja entre las listas y un arreglo es que el orden de los elementos puede ser diferente al orden de almacenamiento de la memoria, lo que nos permitió que el orden de recorrido sea diferente.

Además se pudo percatar la gran ventaja que supone las listas doblemente enlazadas con respecto a la lista simplemente enlazada implementada en la práctica anterior, ya que se corría mayor riesgo en perder el acceso a algún nodo, en cambio en las listas doblemente enlazadas tenemos un doble enlace.

En cuanto a la práctica considero que se tuvo un desarrollo relativamente bueno, excepto por cierta repetición de código lo cual le resta calidad, de ahí en fuera considero que todo se realizó con normalidad.

Martínez Islas Mauricio Joel

Se tuvieron algunas inconsistencias en el diseño, pero lo bueno es que se lograron comentar todas en el código. Otro problema que resulto es la repetición de código, cosa que no debió de haber pasado, pero simplemente por evitar un posible problema, se ignoró la repetición.

No es del todo inútil el código, ya que se cubrieron la mayoría de los casos de entrada de usuario, pero como siempre, no ha quedado debidamente validado y en lo único en lo que se concentro fue en la elaboración de las funciones principales del menú.

Referencias.

[1] Y. Langsam, M. Augenstein and A. Tenenbaum, Estructuras de datos con C y C++, 1st ed. Pearson Prentice Hall, 2000.

[2] S. Villalobos, a las Estructuras de Datos, 1st ed Pearson Educación de México, de México, [3]"Data Structures - GeeksforGeeks", GeeksforGeeks, 2017. [Online]. Available: <http://www.geeksforgeeks.org/data-structures/>. [Accessed: 06-Apr- 2017].

[4]"Data Structures - University of California, San Diego, Higher School of Economics | Coursera", Coursera, 2017. [Online]. Available: <https://www.coursera.org/learn/data-structures>. [Accessed: 06- Apr- 2017].

[5]"Estructuras de datos: listas enlazadas, pilas y colas.", Calcifer.org, 2017. [Online]. Available: <http://www.calcifer.org/documentos/librognome/glib-listsqueues.html>. [Accessed: 06- Apr- 2017].