



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: **Manuel Enrique Castañeda Castañeda**

Asignatura:

Estructura de Datos y Algoritmos I

Grupo:

12

No de Práctica(s):

07 . Estructuras de datos lineales: Lista simple y lista circular

Integrante(s):

- **Lemus Ambrosio Aline Andrea**
- **Reyes Fuentes José Manuel**
- **Sánchez Alvirde Andrés Iván**

*No. de Equipo de cómputo
empleado:*

-

No. de Lista o Brigada:

12

Semestre:

2021-2

Fecha de entrega:

30 de julio de 2021

Observaciones:

Calificación:

Estructura de Datos y Algoritmos I

Práctica 07

Estructuras de datos lineales: Lista simple y
lista circular

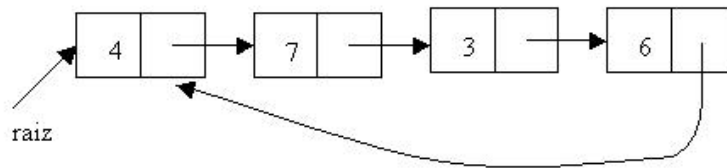
INTRODUCCIÓN

En esta práctica estuvimos usando lo que son las listas simples y las listas circulares daremos un pequeño repaso de lo que es cada una dando algunos ejemplos en los cuales podremos mostrar algunas imágenes.

Hablaremos acerca de lo que es una lista circular definiendo un tanto su concepto.

Una lista circular es una lista lineal en la que el último nodo apunta al primero. En este tipo de listas si avanzamos raíz no perdemos la referencia al nodo anterior ya que es un círculo.

Lo podremos apreciar en la siguiente imagen un tanto más claro:



Cada nodo siempre tiene uno anterior y uno siguiente

Para manejar una lista tendremos que declarar algunos punteros y nodos para poder tener un mayor control acerca de estos mismos, como se muestra de la siguiente forma:

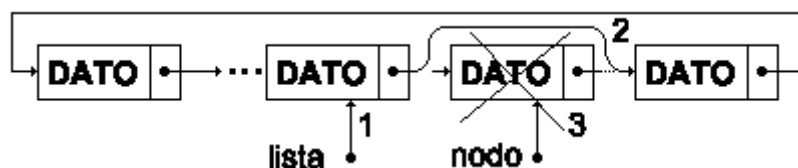
```
typedef struct _nodo {  
    int dato;  
    struct _nodo *siguiente;  
} tipoNodo;  
  
typedef tipoNodo *pNodo;  
typedef tipoNodo *Lista;
```

Para lo que son los nodos se tendrán un poco de problemas los cuales pueden ser: en un proceso de búsqueda, no es tan sencillo dar por terminada la búsqueda cuando el elemento buscado no existe. Una solución para esto sería crear un nodo especial que hará la función de nodo cabecera. De este modo, la lista nunca estará vacía, y se eliminan casi todos los casos especiales.

Para la eliminación de un nodo en la lista circular debemos tener en cuenta que puede estar ligado a otros nodos como pudimos ver en la imagen que están entrelazados todos.

1. El primer paso es conseguir que lista apunte al nodo anterior al que queremos eliminar. Esto se consigue haciendo que lista valga lista->siguiente mientras lista->siguiente sea distinto de nodo.
2. Hacemos que lista->siguiente apunte a nodo->siguiente.
3. Eliminamos el nodo.

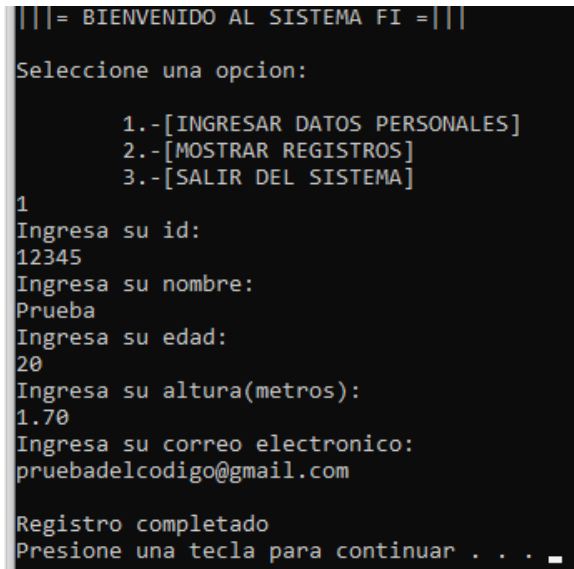
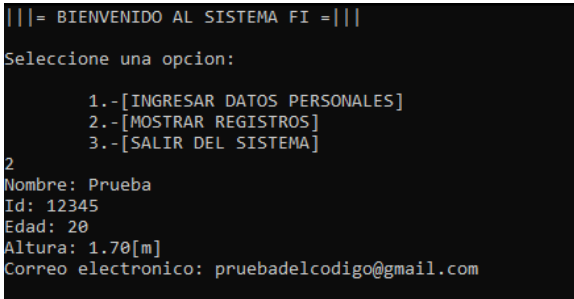
De esa forma quedaría de la siguiente manera:



De esta forma se eliminaría el nodo y la lista continuaría tomando su lugar otro nodo sin que hay algún problema.

DESARROLLO

1. Capturar los datos de una persona: id, nombre, edad, altura y correo electrónico, en una lista ligada circular

CÓDIGO FUENTE	EJECUCIÓN
<pre>#include <stdio.h> #include <stdlib.h> struct nodo{ int id,edad; float altura; char nombre[20],correo[30]; nodo *siguiente; nodo *anterior; }; nodo* primero=NULL; nodo* ultimo=NULL; void insertar(); void mostrar(); int main(){ int op; do{ printf(" = BIENVENIDO AL SISTEMA FI = \n"); printf("\nSeleccione una opcion:\n"); printf("\n\t1.-[INGRESAR DATOS PERSONALES]\n"); printf("\n\t2.-[MOSTRAR REGISTROS]\n"); printf("\n\t3.-[SALIR DEL SISTEMA]\n"); scanf("%d",&op); switch(op){ case 1: insertar(); system("pause"); system("cls"); break; case 2: mostrar(); system("pause"); system("cls");</pre>	 <p>The screenshot shows the program's output for the first execution. It starts with a welcome message, followed by a menu with three options. Option 1 is selected, and the user enters the following data: id: 12345, nombre: Prueba, edad: 20, altura: 1.70, and correo electronico: prubadelcodigo@gmail.com. The program then displays 'Registro completado' and prompts the user to press a key to continue.</p>
	 <p>The screenshot shows the program's output for the second execution. It starts with the same welcome message and menu. Option 2 is selected, and the program displays the data entered in the previous run: Nombre: Prueba, Id: 12345, Edad: 20, Altura: 1.70[m], and Correo electronico: prubadelcodigo@gmail.com.</p>

```

        break;
        case 3:

printf("\nGracias!!! vuelva pronto\n");
        break;
    }
    }while(op<3);
}

void insertar(){
    nodo *nuevo=(nodo
*)malloc(sizeof(nodo));
    printf("Ingresa su id:\n");
    scanf("%d",&nuevo->id);
    printf("Ingresa su nombre:\n");
    for(int i=0;i<1;i++){

scanf("%s",&nuevo->nombre[i]);
    }
    printf("Ingresa su edad:\n");
    scanf("%d",&nuevo->edad);
    printf("Ingresa su
altura(metros):\n");
    scanf("%f",&nuevo->altura);
    printf("Ingresa su correo
electronico:\n");
    for(int i=0;i<1;i++){

scanf("%s",&nuevo->correo[i]);
    }
    if(primer==NULL){
        primero=nuevo;

primero->siguiente=primero;
        ultimo=primero;
        primero->anterior=ultimo;
    }
    else{
        ultimo->siguiente=nuevo;

nuevo->siguiente=primero;
        nuevo->anterior=ultimo;
        ultimo=nuevo;
        primero->anterior=ultimo;
    }
    printf("\nRegistro
completado\n\n");
}

```

<pre> void mostrar(){ nodo *actual=(nodo*)malloc(sizeof(nodo)); actual=primero; if(actual!=NULL){ do{ printf("Nombre: %s\nId: %d\nEdad: %d\nAltura: %.2f[m]\nCorreo electronico: %s\n",actual->nombre,actual->id,actual- >edad,actual->altura,actual->correo); actual=actual->siguiente; }while(actual!=primero); printf("\n"); } else{ printf("No hay datos para mostrar\n"); } } </pre>	
--	--

2. Elaborar un programa que obtenga el promedio de n alumnos almacenados en una lista simple

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include<stdio.h> #include <stdlib.h> struct listA{ float dato; listA *siguiente; }; listA *lista; void insertar(); void promedioTotal(listA *siguiente); int alum; main(){ int op,b=0; do{ printf("\n== Seleccione una opcion == "); printf("\n1.Ingresar calificacion"); </pre>	<pre> == Seleccione una opcion == 1.Ingresar calificacion 2.Promedio 3.Salir 1 Cuantos alumnos va a promediar? 4 Ingresa la calificacion 1: 10 Ingresa la calificacion 2: 9.9 Ingresa la calificacion 3: 6.9 Ingresa la calificacion 4: 5.4 </pre>

```

        printf("\n2.Promedio");
        printf("\n3.Salir\n");
        scanf("%d",&op);
        switch(op){
            case 1:
                insertar();

system("pause");

system("cls");

                break;
            case 2:

promedioTotal(lista);

system("pause");

system("cls");

                break;
            case 3:
                b=1;
                break;
        }
    }while(b==0);

    return 0;
}
void insertar(){
    printf("Cuantos alumnos va a
promediar?\n");
    scanf("%d",&alum);
    for(int i=1;i<=alum;i++){
        listA
*nuevo=(listA*)malloc(alum*sizeof(listA)
);
        nuevo->siguiente=lista;
        printf("\nIngresa la
calificacion %d: ",i);
        scanf("%f",&nuevo->dato);
        lista=nuevo;
    }
}

void promedioTotal(listA *siguiente){
    listA *indice=lista;
    float suma=0;
    int n=0;
    if(indice!=NULL){
        printf("\nLas calificaciones

```

```

== Seleccione una opcion ==
1.Ingresar calificacion
2.Promedio
3.Salir
2

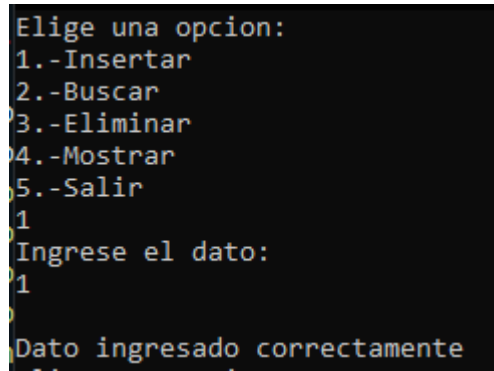
Las calificaciones son:
[5.40] [6.90] [9.90] [10.00]

-----
|| El promedio es = 8.05||
-----
Presione una tecla para continuar . . . _

```

<pre> son:\n"); while(indice!=NULL){ suma=suma+indice->dato; printf("[%.2f]\t",indice->dato); indice=indice->siguiente; n++; } printf("\n"); float promedio; promedio = suma/n; printf("\n"); printf("\t -----"); printf("\n\t El promedio es = %.2f \n\a",promedio); printf("\t -----"); printf("\n"); } else{ printf("\n No hay datos en la lista\n"); } } </pre>	
---	--

3. Implementar una lista doblemente ligada con las funciones, buscar, insertar, borrar y mostrar

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> #include <stdlib.h> struct nodo{ int dato; nodo *siguiente; nodo *anterior; }; nodo *primero=NULL; nodo *ultimo=NULL; void insertarNodo(); void buscarNodo(); </pre>	 <pre> Elige una opcion: 1.-Insertar 2.-Buscar 3.-Eliminar 4.-Mostrar 5.-Salir 1 Ingrese el dato: 1 Dato ingresado correctamente </pre>


```

void eliminarNodo();
void mostrarLista();
int main(){
    int op;
    do{
        printf("Elige una
opcion:\n");
        printf("1.-Insertar\n");
        printf("2.-Buscar\n");
        printf("3.-Eliminar\n");
        printf("4.-Mostrar\n");
        printf("5.-Salir\n");
        scanf("%d",&op);
        switch(op){
            case 1:
                insertarNodo();
                break;
            case 2:
                buscarNodo();
                break;
            case 3:
                eliminarNodo();
                break;
            case 4:
                mostrarLista();
                break;
            case 5:
                printf("\nVuelva pronto\n");
                break;
        }
    }while(op<5);
}

void insertarNodo(){
    nodo
    *nuevo=(nodo*)malloc(sizeof(nodo));
    printf("Ingresa el dato:\n");
    scanf("%d",&nuevo->dato);
    if(primeros==NULL){
        primeros=nuevo;
        primeros->siguiente=primeros; //ligarse
        ultimo=primeros;
        primeros->anterior=ultimo;
    }
}

```

```

Elige una opcion:
1.-Insertar
2.-Buscar
3.-Eliminar
4.-Mostrar
5.-Salir
2
Ingresa el dato a buscar:
1
Se encuentra el dato
Elige una opcion:

```

```

Elige una opcion:
1.-Insertar
2.-Buscar
3.-Eliminar
4.-Mostrar
5.-Salir
4
[1]

```

```

    }
    else{
        ultimo->siguiente=nuevo;

nuevo->siguiente=primero;
        nuevo->anterior=ultimo;
        ultimo=nuevo;
        primero->anterior=ultimo;
    }
    printf("\nDato ingresado
correctamente\n");
}

void buscarNodo(){
    int nodoBuscado,encontrado=0;
    nodo
*actual=(nodo*)malloc(sizeof(nodo));
    actual=primero;
    printf("Ingresa el dato a
buscar:\n");
    scanf("%d",&nodoBuscado);
    if(primero!=NULL){
        do{

if(actual->dato==nodoBuscado){
            printf("Se
encuentra el dato\n");

            encontrado=1;
        }

        actual=actual->siguiente;
    }while(encontrado==0 &&
actual!=primero);
        if(encontrado==0){
            printf("No se
encuentra el dato\n");
        }
    }
    else{
        printf("La lista se
encuentra vacia\n");
    }
}

void eliminarNodo(){
    int nodoBuscado,encontrado=0;
    nodo
*actual=(nodo*)malloc(sizeof(nodo));

```

<pre> actual=primero; nodo *ant=(nodo*)malloc(sizeof(nodo)); ant=NULL; printf("Ingresa el dato a buscar:\n"); scanf("%d",&nodoBuscado); if(primero!=NULL){ do{ if(actual->dato==nodoBuscado){ printf("Se encuentra el dato\n"); if(actual==primero){ primero=primero->siguiente; primero->anterior=ultimo; ultimo->siguiente=primero; } else if(actual==ultimo){ ultimo=ant; ultimo->siguiente=primero; primero->anterior=ultimo; } else{ ant->siguiente=actual->siguiente; actual->siguiente->anterior=ant; } printf("Nodo eliminado\n"); encontrado=1; } ant=actual; actual=actual->siguiente; }while(encontrado==0 && actual!=primero); if(encontrado==0){ printf("No se </pre>	
--	--

<pre> encuentra el dato\n"); } else{ free(ant); } } else{ printf("La lista se encuentra vacia\n"); } } void mostrarLista(){ nodo *actual=(nodo*)malloc(sizeof(nodo)); actual=primero; if(actual!=NULL){ do{ printf("[%d]\t",actual->dato); actual=actual->siguiente; }while(actual!=primero); printf("\n"); } else{ printf("La lista esta vacial\n"); } } </pre>	
---	--

4. Elaborar un programa que obtenga los números primos hasta n en una lista ligada circular

CÓDIGO FUENTE	EJECUCIÓN
---------------	-----------

```

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

struct nodo{
    int dato;
    nodo *siguiente;
    nodo *anterior;
};

nodo *primero=NULL;
nodo *ultimo=NULL;

int main(){
    int x,cont=0,pos=1;

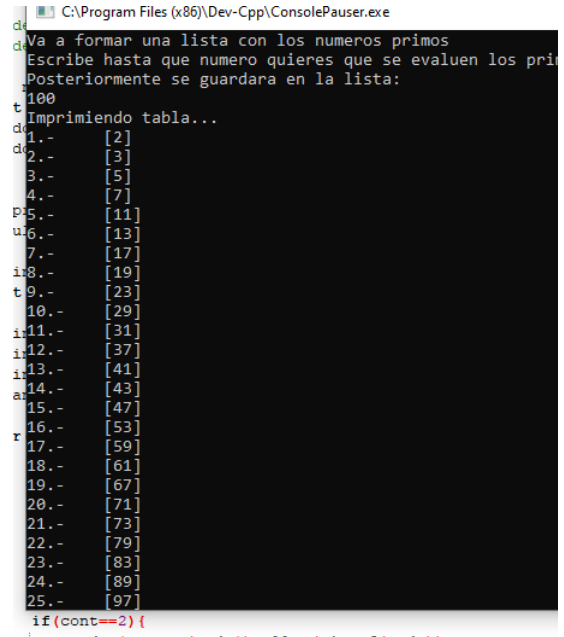
    printf("Va a formar una lista con los numeros primos\n");
    printf("Escribe hasta que numero quieres que se evaluen los primos\n");
    printf("Posteriormente se guardara en la lista:\n");
    scanf("%d",&x);

    for(int j=1;j<=x;j++){
        cont=0;

        for(int i=1;i<=j;i++){
            if(j%i==0){
                cont++;
            }
        }
        if(cont==2){
            nodo
*nuevo=(nodo*)malloc(sizeof(nodo));
            nuevo->dato=j;

            if(primero==NULL){
                //Lista
                vacia
                primero=nuevo;
                primero->siguiente=primero;
                ultimo=primero;
                primero->anterior=ultimo;
            }
        }
    }
}

```



```

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Va a formar una lista con los numeros primos
Escribe hasta que numero quieres que se evaluen los primos
100
Posteriormente se guardara en la lista:
Imprimiendo tabla...
1.- [2]
2.- [3]
3.- [5]
4.- [7]
5.- [11]
6.- [13]
7.- [17]
8.- [19]
9.- [23]
10.- [29]
11.- [31]
12.- [37]
13.- [41]
14.- [43]
15.- [47]
16.- [53]
17.- [59]
18.- [61]
19.- [67]
20.- [71]
21.- [73]
22.- [79]
23.- [83]
24.- [89]
25.- [97]
if (cont==2) {

```

<pre>else{ //Lista con datos ultimo->siguiente=nuevo; nuevo->siguiente=primero; nuevo->anterior=ultimo; ultimo=nuevo; primero->anterior=ultimo; } }</pre> <pre> nodo *actual=(nodo*)malloc(sizeof(nodo)); printf("Imprimiendo tabla...\n"); actual=primero; if(primero!=NULL){ do{ printf("%d.-\t[%d]\n",pos,actual->dato); pos++; actual=actual->siguiente; }while(actual!=primero); } }</pre>	
--	--

OBSERVACIONES (Individuales)

- **Lemus Ambrosio Aline Andrea**

En esta ocasión tuve dos problemas con la práctica, primero fue el recordar y aprenderme bien como es que los datos se van recorriendo con los nodos anterior y siguiente, mi segundo problema fue el último ejercicio de la lista ligada circular, ya que no comprendía bien cómo se hacen y funcionan. Para comprender bien estos problemas tuve que recurrir a ver los videos de clase y aparte otros, también mis compañeros me ayudaron a comprenderlo mejor, por lo que pude comprender mejor el funcionamiento de cada ejercicio.

- **Reyes Fuentes José Manuel**

El mayor reto en esta práctica al menos para mi, fue el comprender las listas circulares y doblemente ligadas, realmente ir comprendiendo poco a poco los programas para saber a dónde iba la información y cómo se entrelaza cada una fue un tanto difícil; aun así con la ayuda de diagramas encontrados en internet y la ayuda de ejemplos que dio el profesor fue un tanto más fácil comprenderlos para saber cómo se entrelaza la información.

- **Sánchez Alvirde Andrés Iván**

En esta práctica lo que más me cuesta a mi es hacerme a la idea de cómo se van recorriendo los datos con los nodos anterior y siguiente, es confuso la primera vez que se ve y uno intenta imaginarlo en la cabeza para así saber cuál operación toca colocar para que se guarden bien los datos. Me tuve que ver varias veces el video de la clase en la que se vio esto y aparte el video donde exponen los compañeros sobre este tema, gracias al verlos me ayudo mucho, los ejemplos que ponían con la ayuda de las guías visuales que hacían en excel o el profesor en word fueron de mucha ayuda para comprender el funcionamiento del siguiente y anterior en este tipo de listas.

CONCLUSIONES (Individuales)

- **Lemus Ambrosio Aline Andrea**

Con esta práctica pude reforzar mis conocimientos de los nodos, además de aprender cómo se recorren los datos con los nodos anterior y siguiente, también el aprender como funcionan las listas ligadas circulares. La práctica que hemos realizado me permitió reforzar mis conocimientos y aprender por medio de videos, haciéndolo un poco más divertido.

- **Reyes Fuentes José Manuel**

Con esta práctica pude comprender mejor el flujo de información dentro de las listas circulares, debido a que van enlazadas más que nada la información una tras de otra.

Con el primer ejercicio pudimos observar cómo se podrían entrelazar diferentes datos de distinta índole como lo son los números con algunos caracteres, el pensar como debíamos poner los códigos fue un tanto interesante como se entrelazan.

- **Sánchez Alvirde Andrés Iván**

Esta práctica me gustó debido a que se enseña el uso de las listas con sus ligaduras para crear códigos que almacenan muchos datos de manera eficiente y fácil para el usuario que vaya a ingresar datos en dichas listas, también se repasaron cosas como la memoria dinámica, uso de funciones para darle una mejor estética al código y la creación de menús.