



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 1

No de Práctica(s): 7- 8

Integrante(s): Díaz Hernández Marcos Bryan

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 9

Semestre: 2020-2

Fecha de entrega: 12 de junio de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la práctica

P7: Revisar las definiciones, características, procedimientos y ejemplos de las estructuras lineales Lista simple y Lista circular, con la finalidad de que comprenda sus estructuras e implementarlas.

P8: Revisar las definiciones, características, procedimientos y ejemplos de las estructuras lineales lista doblemente ligada y lista doblemente ligada circular, con la finalidad de que comprenda sus estructuras e implementarlas.

Introducción

Este reporte lo he elaborado con esmero y con la intención de poder expresar lo aprendido durante la última práctica de la materia, así mismo este hecho para que yo pueda comprender lo que contiene.

Ejercicios de la guía de laboratorio

En esta ocasión no hubo ejercicios que elaborar en la guía del laboratorio.

Ejercicios de la Práctica

- Ejercicio 1

El primer ejercicio de la práctica consiste en modificar la biblioteca lista.h, la cual contenía las funciones para poder modificar una lista enlazada simple, las modificaciones que se realizaron en la biblioteca correspondieron a las funciones de eliminarMayores, eliminarlesimo, y buscar un elemento. Además de realizar un menú para poder realizar cada una de las funciones que estaba en la biblioteca y las que se modificaron.

- Dificultades en el código

Las partes que más se me dificultaron fueron la de eliminar l-ésimo y la de eliminarMayores porque como lo planteé, eliminarMayores necesitaba de eliminar l-ésimo por ello fallaba la segunda, en este caso mi error estaba en una igualdad ya que únicamente utilizaba un símbolo de igual y el programa toma como una condición que cumplir a ese igual (Imagen 1).

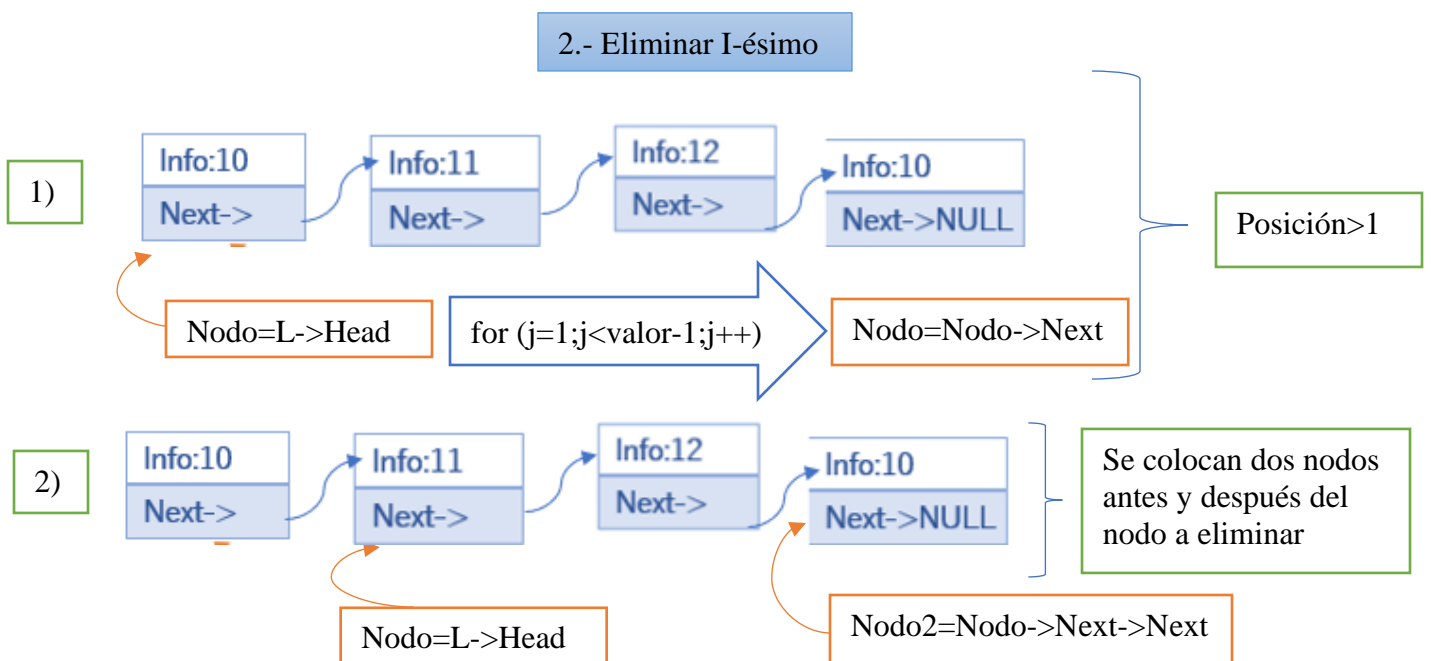
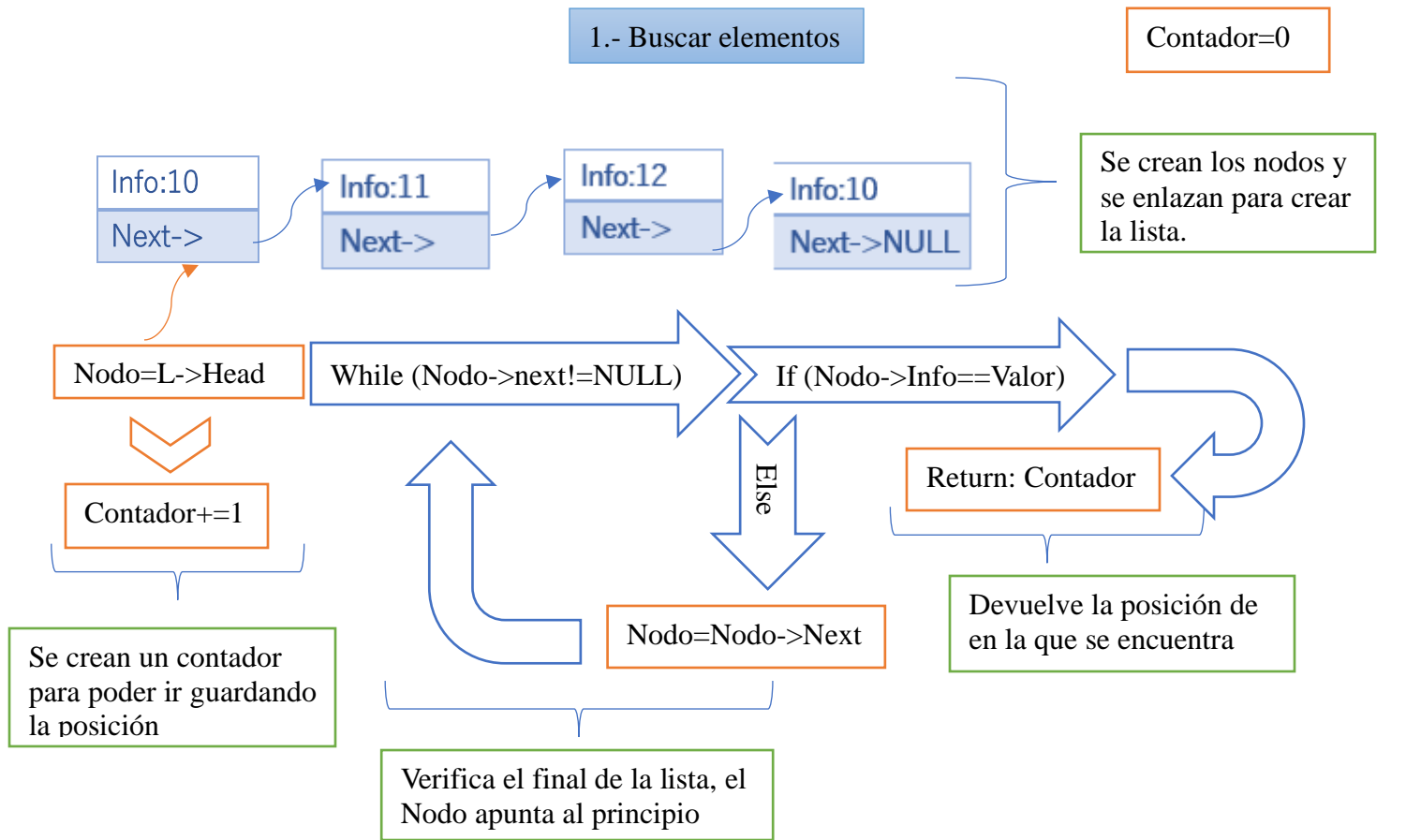
```
}  
if(valor==1){  
    borrarPrimero(Lista);  
}
```

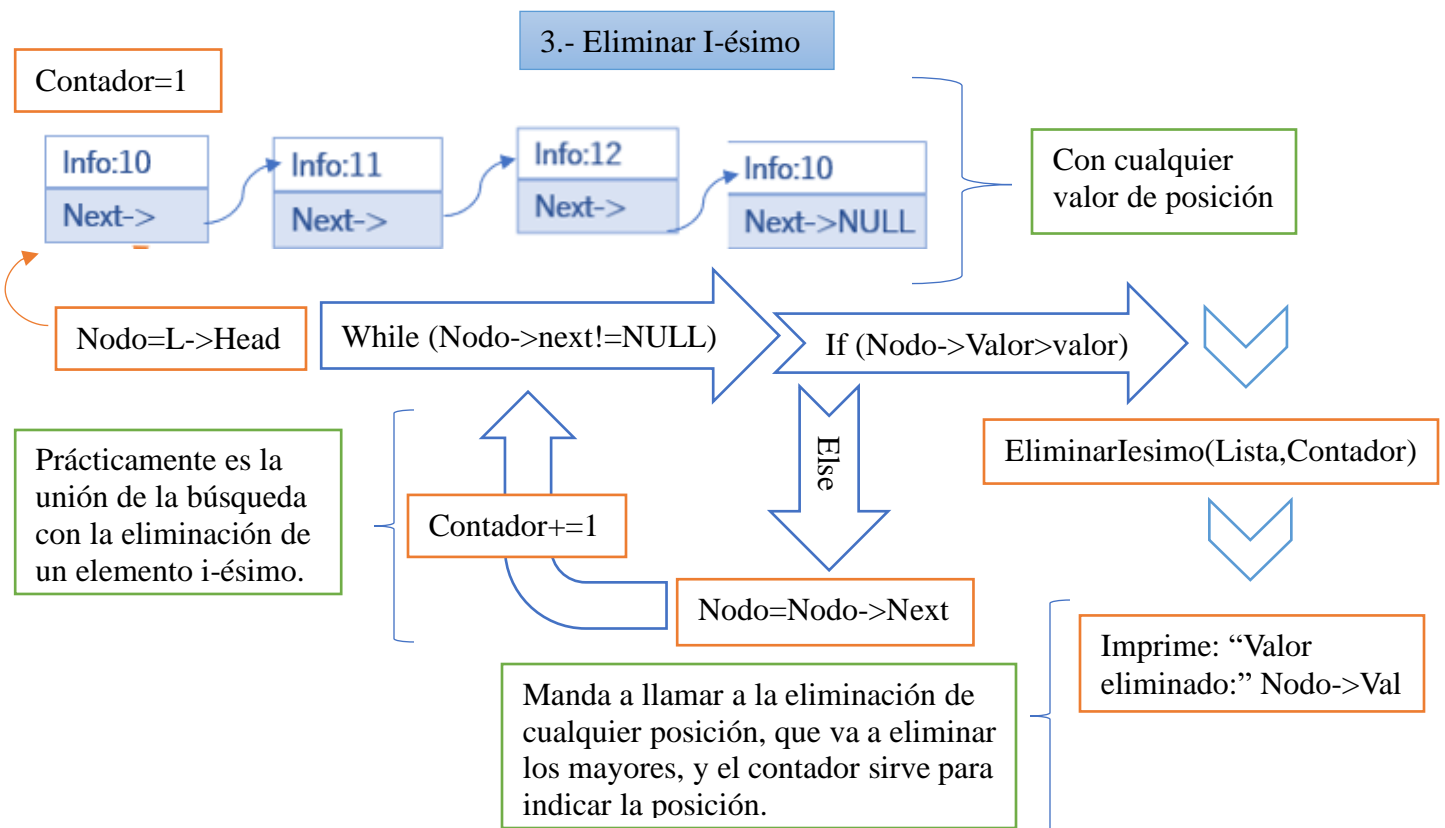
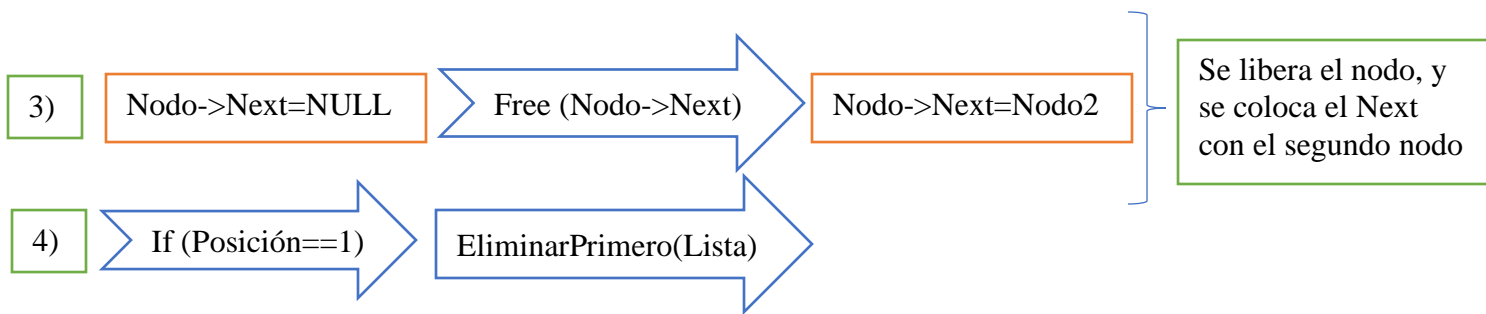
Imagen 1

- ¿Cómo lo resolví?

Para resolverlo lo único que hice fue checar varias veces el problema, y posteriormente me di cuenta del error que cometí y lo modifique colocando de forma correcta la condición (Imagen1).

- Diagrama de funcionamiento.



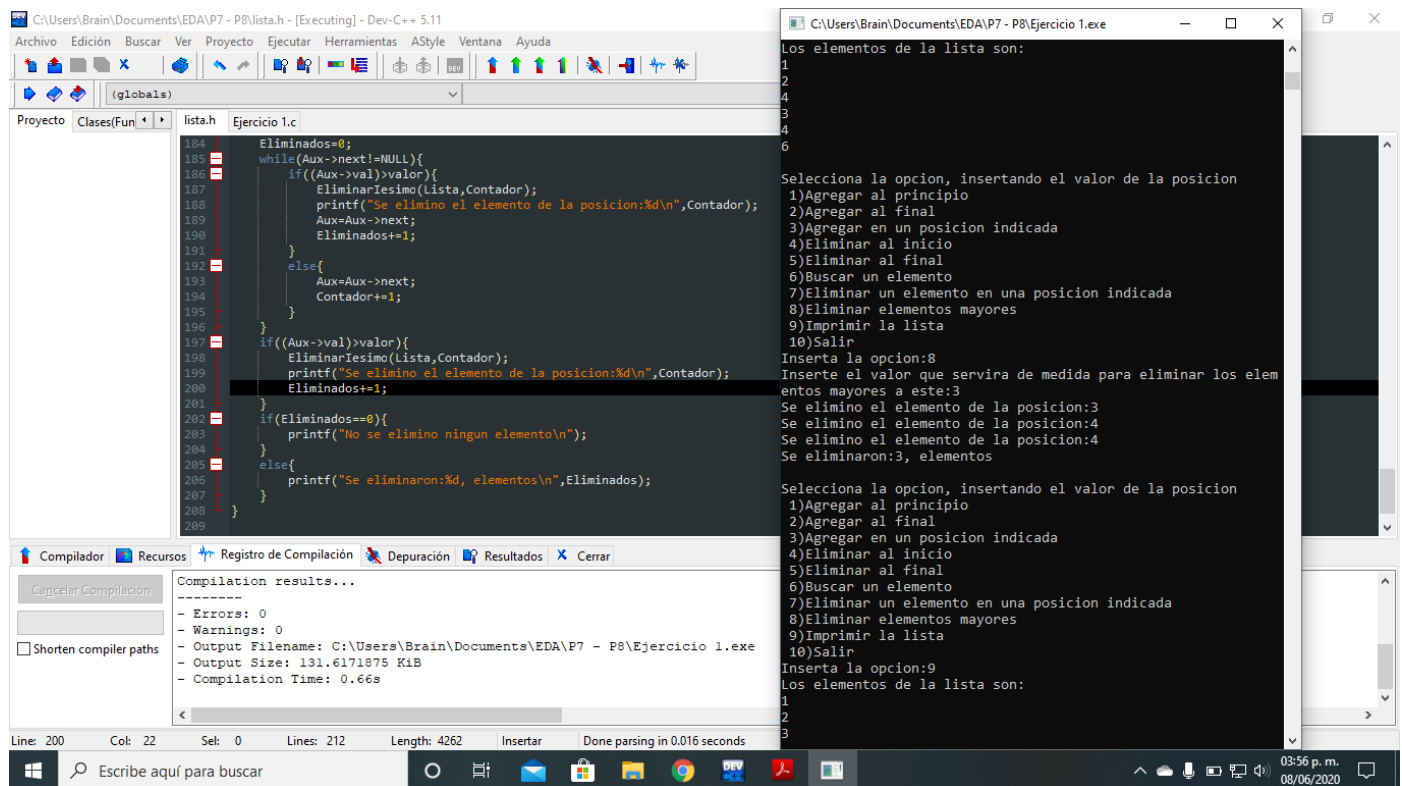


Cuando se elimina un elemento se recorre el índice de posición y por ello se repite el valor de la posición ya que estos se van moviendo a la posición que dejó el que se eliminó.

- Relación con la teoría

Los subproblemas que eran cada función estaban propuestos para poder utilizar los conceptos que se vieron, por ello el hacerlos envolvía el poder recordar lo que se nos dijo en las clases, además que el utilizar estructuras vuelve mas fácil el poder mover los elementos entre funciones y entre sí mismos.

El código de cada función lo estuvimos trabajando en las clases virtuales y en los ejercicios, por ello no se me hizo difícil de realizar, solo que no estaba en las condiciones para realizarlos, de otra forma el primer ejercicio es prácticamente todo lo que vimos en la clase de listas enlazadas simples.



• Ejercicio 2

El ejercicio consistía en realizar las instrucciones necesarias tal que se pudieran comprobar todas las funciones de la biblioteca, por ello opte por realizar un menú igual al anterior ejercicio, de esta forma es mas sencillo, de igual manera era necesario el modelar un TDA auto, y modificar la funciones para que buscaran automóvil por medio del distintivo “marca” y poder realizar una operación de recorrer la lista circular.

• Dificultades en el código

Mis dificultades con el código fueron el poder enviar a la estructura los datos que el usuario insertaba, pero entonces me recordaron las operaciones entre cadenas para poder almacenar o copiar valores. De igual forma el comparar cadenas fue otra situación que me recordaron, por ello igual utilice esa función de “strcmp”, para poder realizar la comparación en la búsqueda del auto.

• ¿Cómo lo resolví?

Para resolverlo agregué la librería string.h, la cual me permitió el poder utilizar la función “strcpy” y la función “strcmp”, para poder guardar y comparar cadenas respectivamente, con estas funciones se hicieron de forma más rápida las operaciones, y pude resolver la situación donde estaba atorado (Imagen 1-2).

```
strcpy(nodo->Placa,A);
strcpy(nodo->Marca,B);
strcpy(nodo->Modelo,C);
strcpy(nodo->Color,D);
strcpy(nodo->Estado,E);
```

Imagen 1

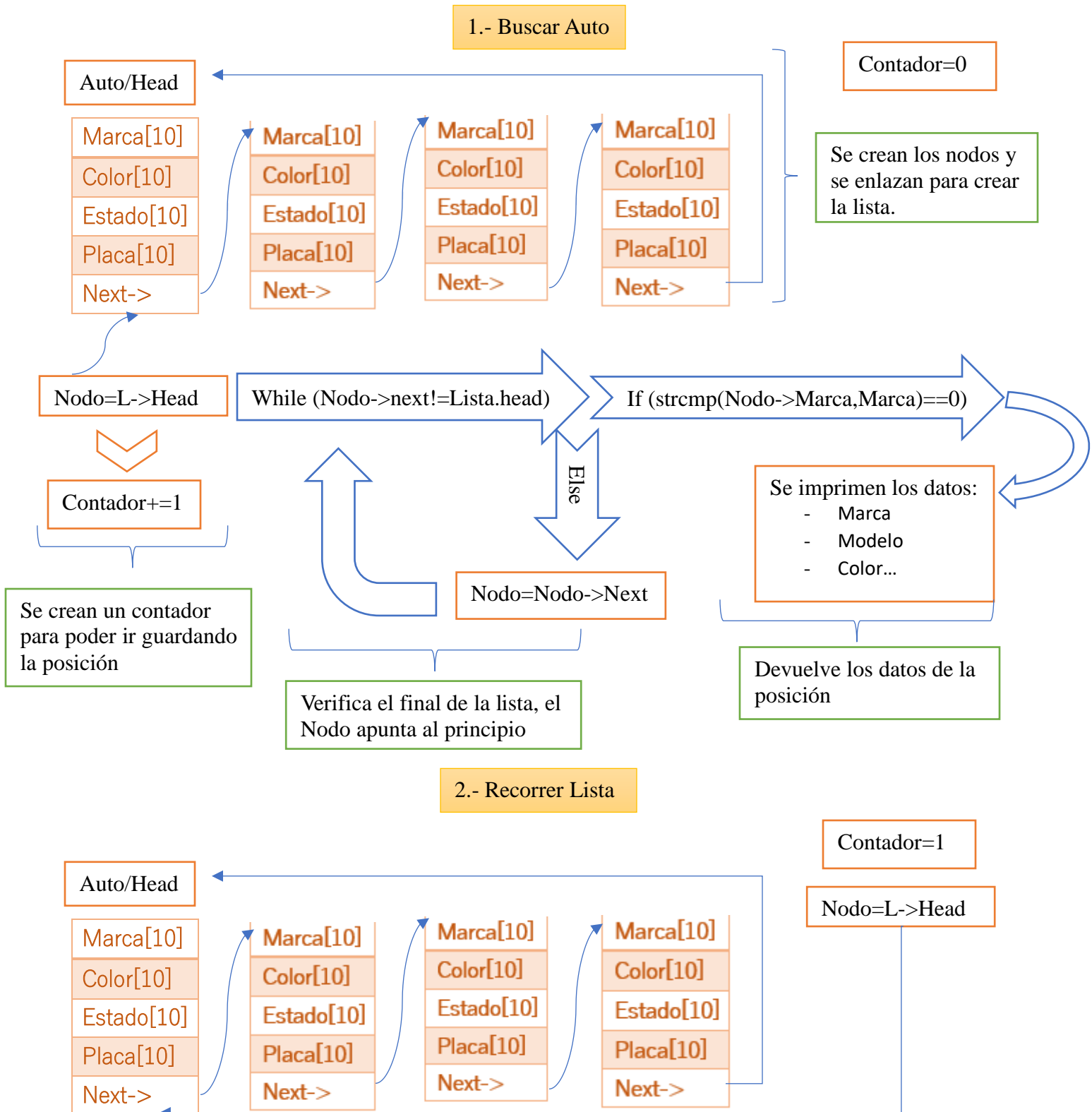
```

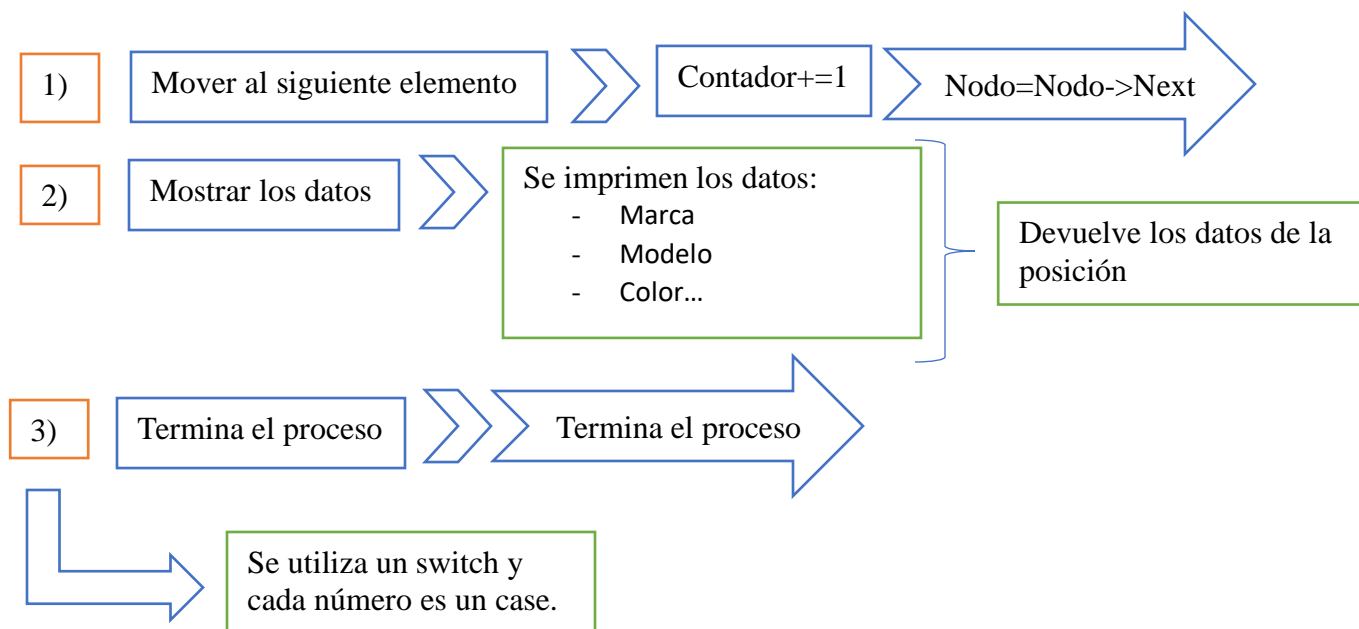
if(Current->next!=Lista.head){
    Contador+=1;
    if(strcmp(Current->Marca,A)==0){
        printf("Se encontro el automovil\n");
        printf("Auto:%d\n",Contador);
    }
}

```

Imagen 2.

- Diagrama de funcionamiento.





- Relación con la teoría

El ejercicio contiene los elementos previos de cadenas, y los elementos dados de listas circulares ligadas por ello no tuve problemas con la parte de listas, porque me quedo claro con lo que habíamos hecho en clase y con los ejercicios, por otro lado los ejercicios fueron de mucha ayuda para poder resolver este ejercicio.

```

1 #include<stdio.h>
2 #include"listacirc.h"
3
4 void main(){
5     printf("Este menu le permite realizar distintas operaciones sobre la lista circular\n");
6     Lista ListaP=crearLista();
7     int opcion;
8     char Placa[10],Marca[10],Modelo[10],Color[10],Estado[10];
9     do{
10        printf("\nSelecciona la opcion, insertando el valor de la posicion\n 1)Agregar al principio\n 2)Agregar al final\n 3)Eliminar al inicio\n 4)Eliminar al final\n 5)Imprimir la lista\n 6)Busqueda\n 7)Recorrer Lista\n");
11        printf("\nInserta la opcion:");
12        scanf("%d",&opcion);
13
14        switch(opcion){
15            case 1:
16                printf("Inserte la placa:");
17                scanf("%s",&Placa);
18                printf("Inserte la marca:");
19                scanf("%s",&Marca);
20                printf("Inserte el modelo:");
21                scanf("%s",&Modelo);
22                printf("Inserte el color:");
23                scanf("%s",&Color);
24                printf("Inserte el estado:");
25                scanf("%s",&Estado);
26                addPrincipioLista(&ListaP,Placa,Marca,Modelo,Color,Estado);
27                break;
28            case 2:
29                printf("Inserte la placa:");
30                scanf("%s",&Placa);
31                printf("Inserte la marca:");
32                scanf("%s",&Marca);
33                printf("Inserte el modelo:");
34                scanf("%s",&Modelo);
35                printf("Inserte el color:");
36                scanf("%s",&Color);
37                printf("Inserte el estado:");
38                scanf("%s",&Estado);
39                addFinalLista(&ListaP,Placa,Marca,Modelo,Color,Estado);
40                break;
41            case 3:
42                printf("Eliminar al inicio:");
43                scanf("%d",&opcion);
44                deleteInicioLista(&ListaP);
45                break;
46            case 4:
47                printf("Eliminar al final:");
48                scanf("%d",&opcion);
49                deleteFinalLista(&ListaP);
50                break;
51            case 5:
52                printf("Imprimir la lista:");
53                scanf("%d",&opcion);
54                imprimirLista(&ListaP);
55                break;
56            case 6:
57                printf("Busqueda:");
58                scanf("%d",&opcion);
59                buscarLista(&ListaP);
60                break;
61            case 7:
62                printf("Recorrer Lista:");
63                scanf("%d",&opcion);
64                recorrerLista(&ListaP);
65                break;
66            default:
67                printf("Opcion no valida\n");
68        }
69    }while(opcion!=0);
70 }
  
```

Output:

```

2)Ver detalles del elemento actual
3)Salir
1
El elemento esta en la posicion:4
1)Mover al siguiente elemento
2)Ver detalles del elemento actual
3)Salir
2
El auto es:4
Marca:Bugatti
Modelo:XZY9
Placa:DDD1
Color:Verde
Estado:Nuevo
1)Mover al siguiente elemento
2)Ver detalles del elemento actual
3)Salir
1
El elemento esta en la posicion:5
1)Mover al siguiente elemento
2)Ver detalles del elemento actual
3)Salir
2
El auto es:5
Marca:Ferrari
Modelo:Vampiro
Placa:AAA1
Color:Rojo
Estado:Nuevo
1)Mover al siguiente elemento
2)Ver detalles del elemento actual
3)Salir
3
Selecciona la opcion, insertando el valor de la posicion
1)Agregar al principio
2)Agregar al final
3)Eliminar al inicio
4)Eliminar al final
5)Imprimir la lista
6)Busqueda
7)Recorrer Lista
8)Salir
  
```

- Ejercicio 3

El tercer ejercicio consiste en realizar las distintas anotaciones que se piden en la práctica, además de como ultimo punto hacer una verificación para que no puedan añadir elementos en posiciones que no están dentro del rango de posiciones permitido.

- Explica por qué no aparece el primer dato que se agrega a la lista y que corrección se debe hacer para que aparezca
 - No aparece debido a que cuando se imprime se salta la primera posición por el ->Next, solo es cuestión de quitarlo y aparece el valor (Imagen 1).

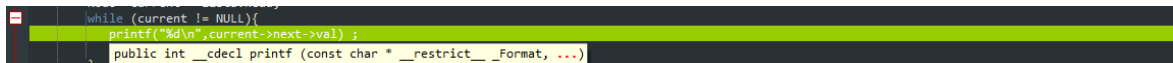


Imagen 1

- Explica la diferencia entre las funciones printList y printfList2
 - La primera función comienza desde la posición del Head, y posteriormente recorre la lista gracias a un while y va imprimiendo los elementos que encuentra, es decir va de forma lineal.
 - La versión dos, llega al final por medio de un ciclo while que le permite llegar al Tail, aunque podría hacerse directo si se añadiera la referencia al final de la lista. Lo que hace la función es llegar al último elemento y comenzar a regresar a partir de ahí, con las referencias de los miembros anteriores ->Prev imprimiendo los valores de manera contraria a como lo hace el primer print.
- Explica para que sirve la función del primer nodo
 - La función redirecciona el Head hacia el nuevo nodo creado, por lo que la información ligada se pierde por completo, ya que el ->Next, ->Prev, no direccionan a ningún elemento que existiera anteriormente.
- ¿Qué se le tendría que agregar a la función agregar i-ésimo para agregar una validación que no permita agregar una posición que no existe en la lista (Esto sin agregar el miembro “tamaño” a la estructura lista)?
 - Para poder resolverlo pensé en como volver a pedir al usuario la posición que estuviese dentro del rango de posibilidades de la lista, por ello dentro del ciclo for, coloque las instrucciones para pedir una nueva posición, donde inserte un nuevo nodo que recorriera la lista, y finalice el ciclo exterior para tener el cambio en la posición indicada (Imagen 2).


```

temp=temp->next;
if(temp==NULL){
    printf("No se puede insertar en esa posición, ya que no existen elementos que lleguen a la posición\n");
    printf("Inserte el valor que este entre el rango de valores que si se pueden añadir (1-%d):",Limite+2);
    scanf("%d",&Nuevo);
    temp=lista->head;
    for(contadorN=1;contadorN<Nuevo-1;contadorN++){
        temp=temp->next;
    }
    contador=posicion;
}

```

Imagen 2

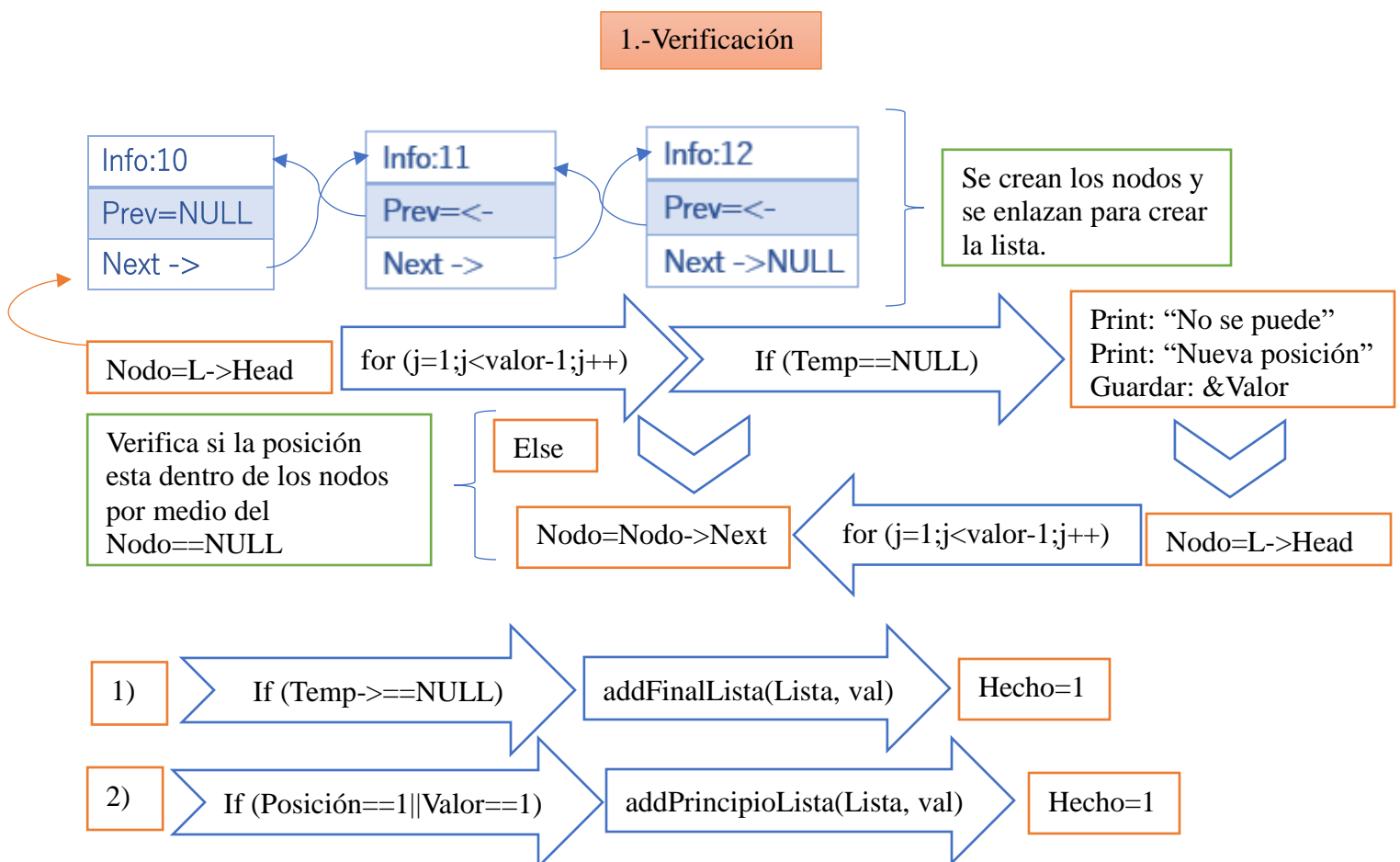
- Dificultades en el código

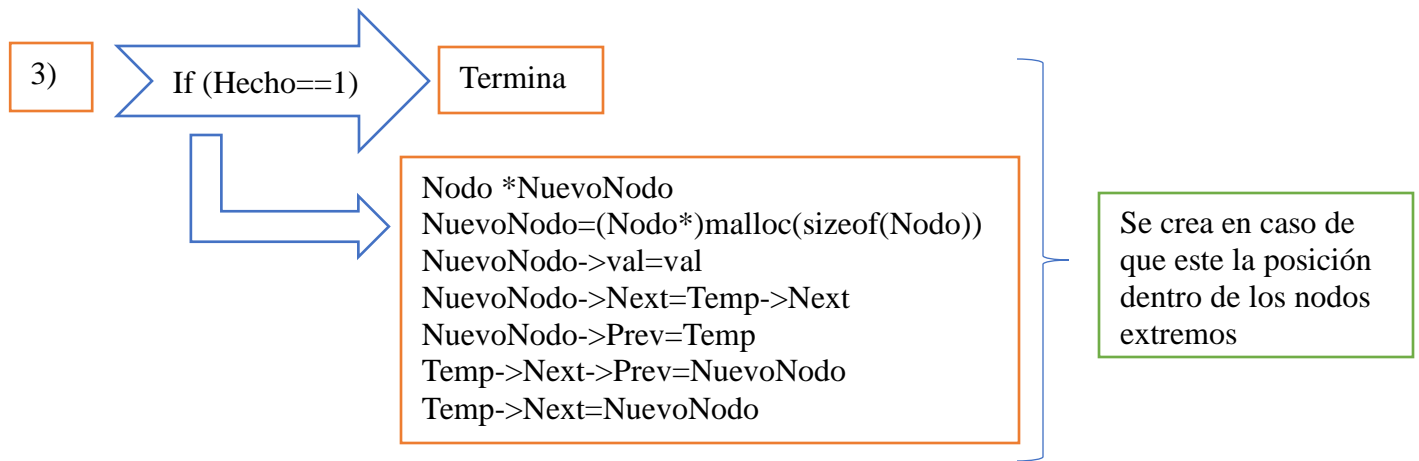
La dificultad estuvo en la verificación ya que no se me ocurría como hacerla, ya que pensaba hacer un case en caso de ciertas condiciones pero no es posible en c, así que decidí reiniciar el for.

- ¿Cómo lo resolví?

Para resolver la verificación, lo que hice fue añadir al condicional que estaba dentro del ciclo for, donde si Temp==NULL, se marcaba error, además de agregar algunas cosas para poder añadir a los extremos de la lista, en este caso lo que hice fue reiniciar el nodo, colocándolo en el head de nuevo y realizando el ciclo for de nuevo igual, en este caso el programa permite al usuario insertar valores dentro de un rango y se indica el rango permitido, así se puede dar más fluidez al programa (Imagen 2).

- Diagrama de funcionamiento.





- Relación con la teoría

El ejercicio contenía mas preguntas conceptuales, y la parte final fue algo que decidí hacer porque no me gustaba como se veía, y porque no permitía añadir al final o al principio de la lista, lo que considere como algo mal, ya que en las listas anteriores si era posible. En cuanto a la solución que decidí dar no me convence del todo ya que mi idea inicial de poner un case resultaba mejor en mi cabeza, aunque esta solución funciona no creo que sea la mejor.

En cuanto al contenido teórico lo que me pareció bastante bueno fue el poder comprobar las variantes de las listas y como estas pueden estar organizadas de distintas formas, y de nuevo el contenido del ejercicio estuvo dentro de lo visto en las clases virtuales.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include "listadoble.h"
4
5 main()
6 {
7     printf("Bienvenido al programa \n");
8
9     int opcion,elemento = 0;
10    Lista milista =crearLista();
11    addFinalista(&milista,10);
12    addFinalista(&milista,20);
13    // addFinalista(&milista,5,1);
14    addFinalista(&milista,30);
15    addFinalista(&milista,40);
16    // addFinalista(&milista,15,2);
17    // addFinalista(&milista,25,4);
18    addFinalista(&milista,50);
19    addFinalista(&milista,60);
20
21    addIesimolista(&milista,70,9);
22
23    print_list(milista);
24 }
```

```
ores que si se pueden azadir (1-7):7
Ya se agrego
Los elementos de la lista son:
10
20
30
40
50
60
70

-----
Process exited after 3.524 seconds with return
value 0
Presione una tecla para continuar . . .
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Brain\Documents\EDA\P7 - P8\P8programa.exe
- Output Size: 131.044921875 KiB
- Compilation Time: 0.36s

- Ejercicio 4

El ultimo ejercicio consistía en modificar la biblioteca de la Cola Circular Doble, como los ejercicios anteriores, en este caso se tenía que completar la función borrar Primero y borrar Ultimo, además de hacer la función recorrer para poder ir hacia adelante o hacia atrás.

- Dificultades en el código

La dificultad que encontré fue que no me di cuenta de que el elemento del TDA Lista, tiene un tamaño que es necesario modificar cuando se elimina o agrega un elemento, ya que al imprimir las instrucciones que había hecho se me repetían términos, y no sabia porque.

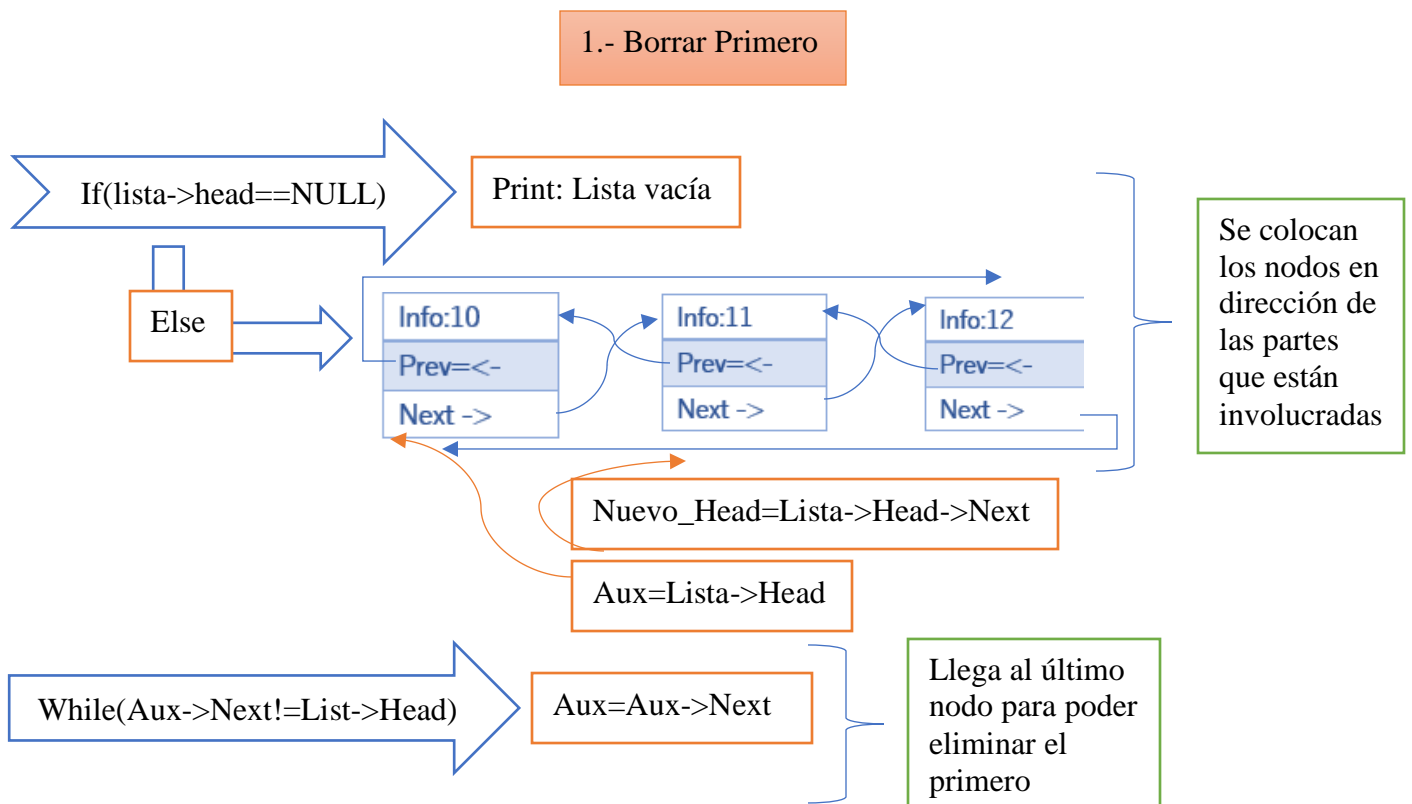
- ¿Cómo lo resolví?

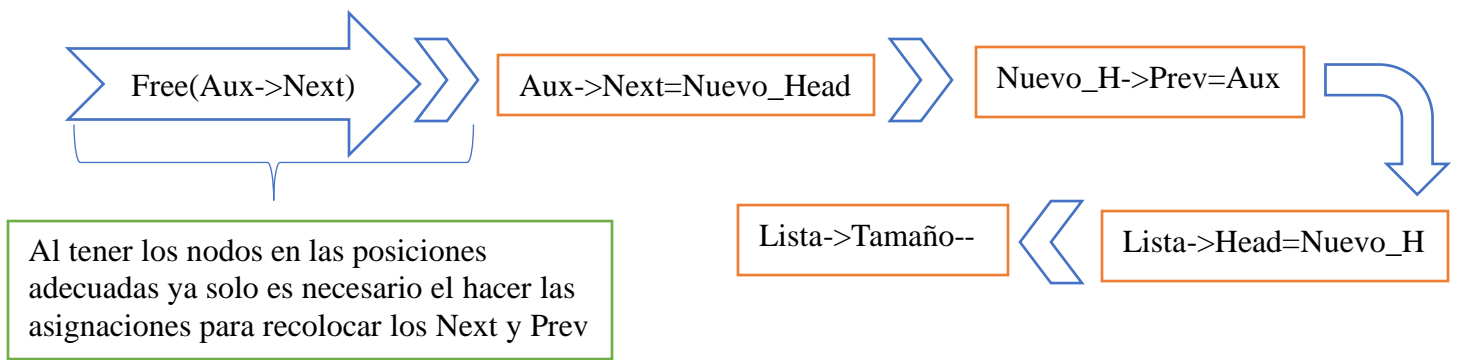
Lo único que hice fue disminuir esta parte del TDA para que no registrara como sino se hubiera borrado ningún elemento, y repitiera los elementos al imprimir los valores (Imagen1).

```
nuevo_head = prev
lista->head = Nuevo
lista->tamano--;
```

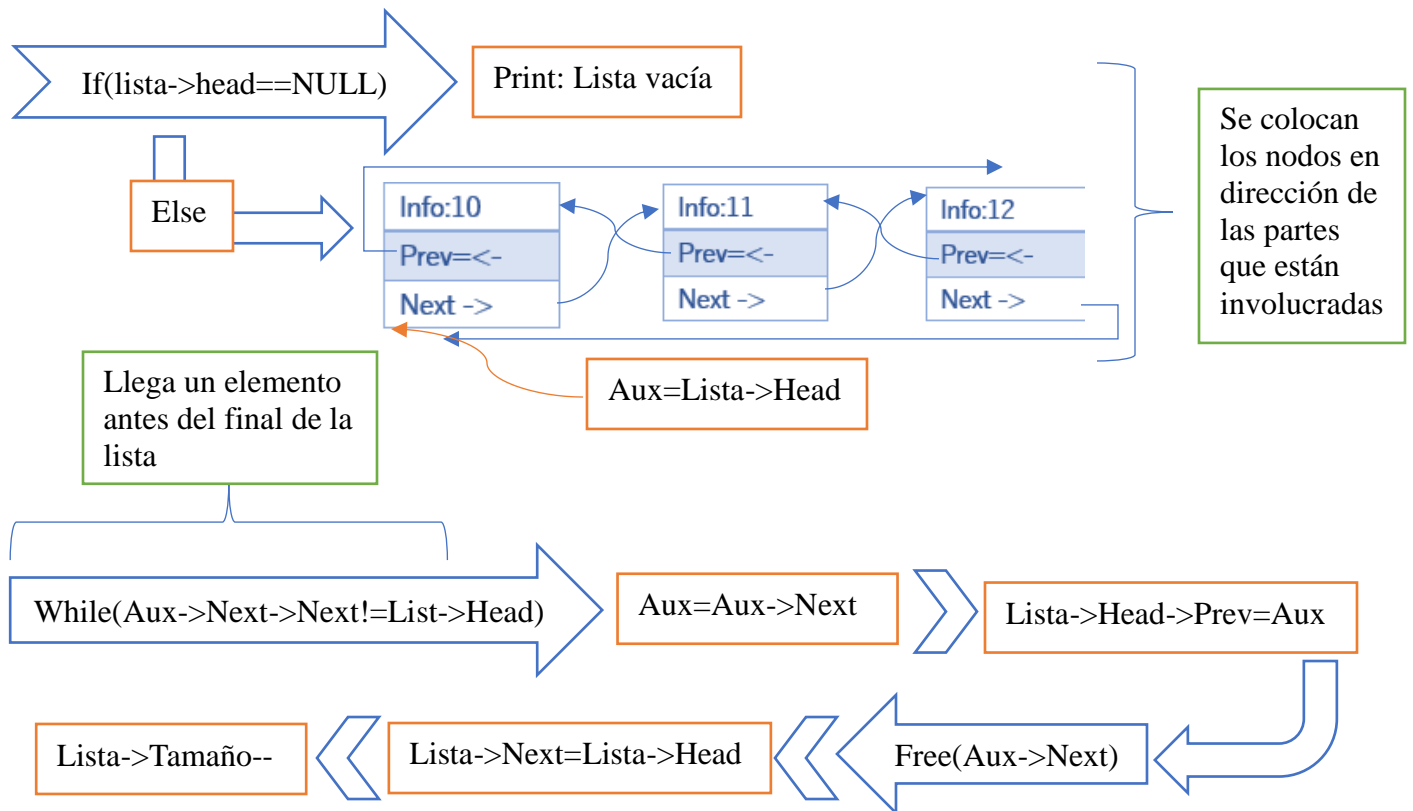
Imagen 1

- Diagrama de funcionamiento.

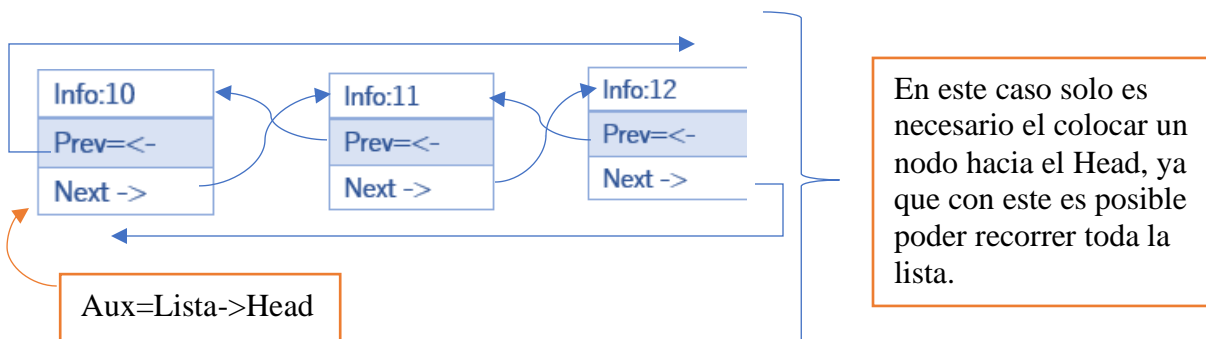


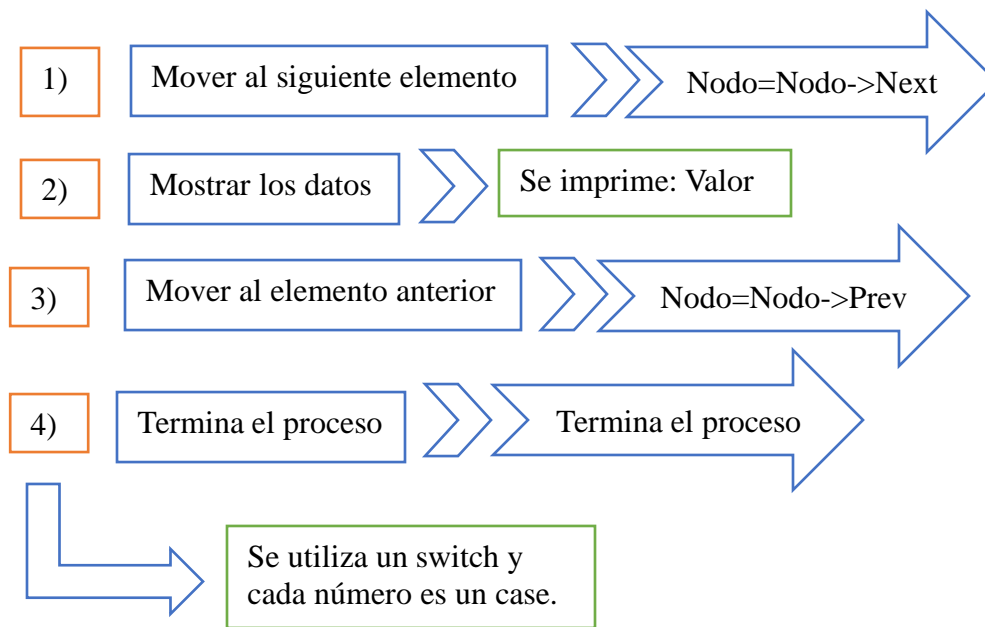


2.- Borrar Ultimo



3.- Recorrer Lista





- Relación con la teoría

El ejercicio relacionaba muy bien lo que vimos en clase con lo que se hizo en la práctica, porque para poder resolver hice lo mismo que en los ejercicios virtuales, y se puede resolver sin problema, en otro caso es necesario dar un repaso a los apuntes de las listas circulares dobles.

```
void recorrerLista(Lista lista){
    int var;
    Nodo *temp = lista.head;
    printf("Para seleccionar es necesario que coloque el valor de la opcion:");
    do{
        printf("1)Mover al siguiente elemento\n 2)Ver el valor del elemento actual\n 3)Mover al elemento anterior\n 4)Salir\n");
        scanf("%d",&var);
        switch(var){
            case 1:
                temp=temp->next;
                break;
            case 2:
                printf("El valor es:%d\n",temp->val);
                break;
            case 3:
                temp=temp->prev;
                break;
            default:
                var=4;
        }
    }while (var<4);
}

void print_list(Lista lista) {
    int tamaño = lista.tamaño;
    if(lista.head==NULL){
        printf("LA LISTA ESTA VACIA \n");
    }
    else{
        printf("Los elementos de la lista son: \n");
    }
}
```

1)Mover al siguiente elemento
2)Ver el valor del elemento actual
3)Mover al elemento anterior
4)Salir
Inserte la opcion:2
El valor es:4
1)Mover al siguiente elemento
2)Ver el valor del elemento actual
3)Mover al elemento anterior
4)Salir
Inserte la opcion:4
Selecciona la opcion, insertando el valor de la posicion
1)Imprimir la lista
2)Agregar al principio
3)Agregar al final
4)Crear un nodo unico
5)Eliminar al inicio
6)Eliminar al final
7)Recurrir la lista
8)Salir
Inserta la opcion:5
Selecciona la opcion, insertando el valor de la posicion
1)Imprimir la lista
2)Agregar al principio
3)Agregar al final
4)Crear un nodo unico
5)Eliminar al inicio
6)Eliminar al final
7)Recurrir la lista
8)Salir
Inserta la opcion:1
Los elementos de la lista son:
2
3
4
5
6
7
8
Selecciona la opcion, insertando el valor de la posicion

Conclusiones

Esta práctica me pareció al principio bastante intimidante, pero al comenzar a resolverla puede hacer de forma fluida y con tiempo así que no me avente todo de golpe, lo tome con calma, y es la primera vez que lo hago de esta forma. Ya que sentí confianza con respecto a lo que había aprendido en las clases virtuales, además el contenido de la práctica se apega a lo visto, y el ver como se hace el código de las listas es bastante curioso.

En base a las diferencias que existen entre las listas me parece que no necesariamente lo más complejo es lo mejor, ya que depende de donde se utilicen, porque al querer utilizar las listas es necesario el saber cuando utilizarlas y que modificaciones son pertinentes para que puedan satisfacer lo que se requiere. Aún así me parece que la más eficiente es la circular doble ya que permite modificar y regresar con facilidad. Por ello el contenido de la práctica y de los temas de las listas se relacionan en la lista doble circular, siendo así que fortalecí mis conocimientos además de poder reconocer la forma que debe llevar cada tipo de lista.

Por último me gustaría dar las gracias al profesor por dejar ejercicios adicionales en cada práctica y dejar que cada uno pudiera desarrollar sus habilidades de forma individual, ya que poner las manos en acción refleja mucho de lo aprendido y de lo que se debe aprender, por ello agradezco, además tuve que mejorar mis reportes que fueron cambiando hasta encontrar un modelo que me gusta y espero que cumpla con los requisitos. Sin más que decir, me despido de usted por este medio, espero se encuentre bien de salud y tome las precauciones necesarias para cuidar su salud y la de sus familiares. Hasta luego :D