

**Universidad Nacional Autónoma de Nicaragua, León**

**Facultad de Ciencias y Tecnología**

**Departamento de Computación**



**Componente: Algoritmo y Estructura de Datos**

**Unidad IV: ESTRUCTURAS DINÁMICAS DE DATOS**

**TEMA: LISTAS CIRCULARES**

**Elaborado por:**

➤ DAVID MARADIAGA GUTIÉRREZ.

***“A la libertad por la Universidad”***



# TABLA DE CONTENIDOS

## Contenido

Ejercicio 1.....	3
Ejercicio Evaluativo.....	5



## Ejercicio 1.

**Ejercicio que pone a prueba las operaciones básicas de una lista circular enlazada.**

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

typedef struct _nodo
{
    int valor;
    struct _nodo *siguiente;
} tipoNodo;

typedef tipoNodo *pNodo;
typedef tipoNodo *Lista;

// Funciones con listas:
void Insertar(Lista *l, int v);
void Borrar(Lista *l, int v);
void BorrarLista(Lista *);
void MostrarLista(Lista l);

int main()
{
    Lista lista = NULL;

    Insertar(&lista, 10);
    Insertar(&lista, 40);
    Insertar(&lista, 30);
    Insertar(&lista, 20);
    Insertar(&lista, 50);
    MostrarLista(lista);

    Borrar(&lista, 30);
    Borrar(&lista, 50);
    MostrarLista(lista);

    BorrarLista(&lista);

    system("PAUSE");
    return 0;
}

void Insertar(Lista *lista, int v)
{
    pNodo nodo;

    // Creamos un nodo para el nuevo valor a insertar
    nodo = (pNodo)malloc(sizeof(tipoNodo));
    nodo->valor = v;
```



```

// Si la lista está vacía, la lista será el nuevo nodo
// Si no lo está, insertamos el nuevo nodo a continuación del apuntado
// por lista
if(*lista == NULL)
    *lista = nodo;
else nodo->siguiente = (*lista)->siguiente;
// En cualquier caso, cerramos la lista circular
(*lista)->siguiente = nodo;
}

void Borrar(Lista *lista, int v)
{
    pNodo nodo;
    nodo = *lista;
    // Hacer que lista apunte al nodo anterior al de valor v
    do
    {
        if((*lista)->siguiente->valor != v)
            *lista = (*lista)->siguiente;
    } while((*lista)->siguiente->valor != v && *lista != nodo);

    // Si existe un nodo con el valor v:
    if((*lista)->siguiente->valor == v)
    {
        // Y si la lista sólo tiene un nodo
        if(*lista == (*lista)->siguiente)
        {
            // Borrar toda la lista
            free(*lista);
            *lista = NULL;
        }
        else {
            // Si la lista tiene más de un nodo, borrar el nodo de valor v
            nodo = (*lista)->siguiente;
            (*lista)->siguiente = nodo->siguiente;
            free(nodo);
        }
    }
}

void BorrarLista(Lista *lista)
{
    pNodo nodo;

    // Mientras la lista tenga más de un nodo
    while((*lista)->siguiente != *lista)
    {
        // Borrar el nodo siguiente al apuntado por lista
        nodo = (*lista)->siguiente;
        (*lista)->siguiente = nodo->siguiente;
        free(nodo);
    }
}

```



```
    }

    // Y borrar el último nodo
    free(*lista);
    *lista = NULL;
}

void MostrarLista(Lista lista)
{
    pNodo nodo = lista;

    printf("\n>Mostrar lista:\n");

    do
    {
        printf("%d -> ", nodo->valor);
        nodo = nodo->siguiente;
    } while(nodo != lista);

    printf("\n");
}
```

Salida de ejecución:

```
>Mostrar lista:
10 -> 50 -> 20 -> 30 -> 40 ->

>Mostrar lista:
10 -> 20 -> 40 ->
Presione una tecla para continuar . . . _
```

**Actividad:** Modificar este ejemplo para trabajarlo con un menú de opciones.

## Ejercicio Evaluativo.

Al momento de ingresar a un hospital, a un paciente se le solicitan los siguientes datos:

- Nombre.
- Edad.
- Sexo.
- Domicilio: (Calle, Numero, Ciudad.)
- Teléfono.
- Seguro (Este campo tendrá valor VERDADERO si el paciente tiene seguro médico y FALSO en otro caso).

Construya un programa en lenguaje C que almacene la información en una lista circular y pueda llevar a cabo las siguientes operaciones:



- Listar los nombres de todos los pacientes hospitalizados.
  - Obtener el porcentaje de pacientes hospitalizados en las siguientes categorías (dadas por edad):
    - Niños: Hasta 13 años.
    - Jóvenes: Mayores de 13 años, y hasta 30 años de edad.
    - Adultos: Mayores de 30 años.
  - Obtener el porcentaje de hombres y mujeres hospitalizados.
  - Dado el nombre de un paciente, listar todos los datos relacionados con dicho paciente.
  - Calcular el porcentaje de pacientes que poseen seguro médico.
- Construir la lista de modo que los datos vayan quedando almacenados en el orden que sean ingresados.

**Consideraciones:**

- Valor: 15 puntos.
- Integrantes: máximo 2 estudiantes.
- Se debe comentar el código fuente.
- Trabajos idénticamente iguales serán penalizados.