



Universidad
Nacional
Autónoma de
Nicaragua, **León**

Facultad de Ciencias y Tecnología
Departamento de Computación
Aplicaciones de Estructuras de Datos

Listas Enlazadas

Ejemplo:

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

typedef struct _datos Datos;
struct _datos{
    int dato;
    Datos *siguiente;
};

//Define un alias "PDatos" a un puntero de de tipo Datos*
typedef Datos* PDatos;

PDatos CrearNuevoElemento(){
    PDatos nuevo = (PDatos) malloc( sizeof( Datos ) );
    if( nuevo == NULL ){
        perror("Error al asignar memoria");
        exit(-1);
    }
    return nuevo;
}

void InsertarElementoEnLaLista( int dato, PDatos *inicio ){
    PDatos nuevo = CrearNuevoElemento();

    nuevo->dato = dato;
    nuevo->siguiente = *inicio;

    *inicio = nuevo;
}

void MostrarLista( PDatos inicio ){
    if( inicio == NULL ){
        printf("\nLa lista esta vacia...");
        return;
    }

    while( inicio != NULL ){
        printf("\ndato: %d", inicio->dato);
        inicio = inicio->siguiente;
    }
}
```

```

void BorrarElemetoDeLaLista( PDatos *inicio ){
    PDatos aux = *inicio; //Apunta al inicio de la lista

    if( aux == NULL ){
        printf("\nLa lista esta vacia...");
        return;
    }

    *inicio = (*inicio)->siguiente; //Apuntura al siguiente de la lista
    free( aux ); //Liberamos la memoria asignada al elemento
}

int main(){
    PDatos inicio = NULL;
    InsertarElementoEnLaLista( 20, &inicio );
    InsertarElementoEnLaLista( 15, &inicio );
    InsertarElementoEnLaLista( 13, &inicio );
    InsertarElementoEnLaLista( 25, &inicio );

    MostrarLista( inicio );

    while( inicio != NULL )
        BorrarElemetoDeLaLista( &inicio );

    return 0;
}

```

Enunciado

Una empresa medica desea realizar un sistema de control de pacientes, el líder del proyecto le encarga el desarrollo de dicho programa utilizando listas enlazadas para llevar el registro de pacientes. Dicha aplicación debe de presentar el siguiente menú

1. Registrar paciente
2. Eliminar datos de paciente
3. Actualizar datos de paciente
4. Mostrar datos de paciente
5. Mostrar listado de paciente

La estructura a utilizar para llevar el control de pacientes

```

struct nodo{
    int codigo;        // campo codigo
    char nombres[30];  // campo que almacena el nombre
    char apellidos[30]; // campo que almacena el apellido
    char direccion[60]; // campo que almacena la direccion
    long telefono;     // campo que almacena el telefono
    struct nodo *sigte;
};

```

Nota: Recuerde que antes que finalice la ejecución del programa debe liberar toda la lista (memoria asignada)