

ED_JuanPabloCostas_ListasDinamicas



- Universidad Autónoma De Querétaro
- Facultad de informática
- Estructura de datos
- M.I.S.D. Erika del Río Magaña

- Costas Rueda Juan Pablo
- EXP: 307081
- 4to Semestre

```
#include <cstdlib>
#include <iostream>

using namespace std;

struct Nodo{
    int data;
    Nodo *next;
};

void menu();

void insertarLista(Nodo *&, int);
```

```
void mostrarLista(Nodo *);

void buscarLista(Nodo *, int n);

void eliminarLista(Nodo *&);

void eliminarNodo(Nodo *&, int);

Nodo * head = NULL;

int main(){
    menu();
}

void menu(){
    bool loop = true;
    do
    {
        int o = 0;
        cout << "\n\nSeleccione una opcion: \n";
        cout << "1- Insertar una lista\n";
        cout << "2- Mostrar la lista\n";
        cout << "3- Buscar elemento en la lista\n";
        cout << "4- Eliminar Nodo\n";
        cout << "5- Eliminar Lista\n";
        cin >> o;

        int n = 0;

        switch (o)
        {
            case 1:
                int x;
                cout << "Ingresa la cantidad de numeros que quieras arreglar: ";
                cin >> x;
                for(int y=0; y < x; y++){
                    cout << "Ingresa un numero entero: ";
                    cin >> n;
                    insertarLista(head, n);
                }
                break;

            case 2:
                mostrarLista(head);
                break;

            case 3:
                cout << "Ingresa el numero a buscar en la lista: ";
                cin >> n;
                buscarLista(head, n);
                break;

            case 4:
                cout << "Ingresa el nodo a eliminar: ";
```

```
        cin >> n;
        eliminarNodo(head, n);
        break;
    case 5:
        eliminarLista(head);
        break;
    default:
        loop = false;
        break;
    }
} while (loop);
}

void insertarLista(Nodo *& head, int n){

    Nodo * new_nodo = new Nodo();
    new_nodo -> data=n;
    Nodo * aux1 = head;
    Nodo * aux2;

    while((aux1 != NULL) && (aux1->data < n)){

        aux2 = aux1;
        aux1 = aux1->next;

    }

    if(head == NULL){
        head = new_nodo;
    }
    else{
        aux2->next = new_nodo;
    }
    new_nodo->next = aux1;

    cout << "El dato ha ingresado a la lista \n";
}

void mostrarLista(Nodo * head){
    Nodo * actual = new Nodo();
    actual = head;
    while(actual != NULL){
        cout << actual -> data << " -> ";
        actual = actual ->next;
    }
}

void buscarLista (Nodo * head, int n){
    bool flag = false;
    Nodo *actual = new Nodo();
    actual = head;
    while ((actual!=NULL)&&(actual->data<=n)){
        if(actual->data==n){
```

```
        flag=true;
    }
    actual = actual->next;
}
if (flag==true)
{
    cout << "El numero " << n << " se ah econtrado en la posicion " <<
&actual;
} else
{
    cout << "El numero " << n << " NO se ah econtrado en la lista";
}
}

void eliminarNodo(Nodo *& head, int n){
    if (head!=NULL)
    {
        Nodo *aux_delete = new Nodo();
        Nodo *previos = NULL;
        aux_delete=head;

        while ((aux_delete!=NULL)&&(aux_delete->data != n))
        {
            previos=aux_delete;
            aux_delete = aux_delete -> next;
        }
        if (previos!=NULL)
        {
            head = head -> next;
            delete aux_delete;
        }
        else if (aux_delete==NULL)
        {
            cout << "El elemento no se ah encontrado";
        }
        else{
            previos->next=aux_delete->next;
            delete aux_delete;
        }

    }
    else{
        cout << "La lista esta vacia";
    }
}

void eliminarLista(Nodo *& head){
    while (head!=NULL)
    {
        Nodo *aux_delete;
        aux_delete=head;
        head=aux_delete->next;
        delete aux_delete;
    }
}
```

```
}
```

```
}
```