

```

1  #include <iostream>
2  #include<fstream>
3  #include "Lista.h"
4  #include "funcionesCargarArchivos.h"
5
6  using namespace std;
7
8  Lista crearLista(){
9      Lista lista=new listaStruct();
10     lista->tamano=0;
11     return lista;
12 }
13
14 void agregarNodo(Lista lista,void* dato){
15     Nodo newNode=new nodo();
16     newNode->dato=dato;
17     if(lista->tamano>0){
18         //Busca el ultimo nodo actual
19         int i=1;
20         Nodo aux=lista->inicio;
21         while(aux->siguiente!=NULL){
22             i++;
23             aux=aux->siguiente;
24         }
25         //Se le asigna el nro del ultimo nodo + 1 y se lo asigna como siguiente al
ultimo
26         newNode->nro=(i+1);
27         newNode->siguiente=NULL;
28         aux->siguiente=newNode;
29     }else{
30         //lista vacia, crea el nodo y le asigna 1 al numero
31         newNode->nro=1;
32         newNode->siguiente=NULL;
33         lista->inicio=newNode;
34     }
35     //aumenta el tamaño de la lista
36     lista->tamano=(lista->tamano+1);
37 }
38
39
40 void eliminarNodo(Lista lista,int nro){
41     //verifica que la lista no este vacia
42     if(lista->tamano>0){
43         //busca el nodo a eliminar
44         Nodo ant=NULL;
45         Nodo aux=lista->inicio;
46         int encontrado=0;
47         while(aux!=NULL && encontrado==0){
48             if(aux->nro==(nro-1)){
49                 ant=aux;
50             }
51             if(aux->nro==nro){
52                 encontrado=1;
53             }else{
54                 aux=aux->siguiente;
55             }
56         }
57         if(encontrado==1){
58             //si lo encuentra corrige los punteros del anterior nodo
59             if(ant!=NULL){
60                 ant->siguiente=aux->siguiente;
61             }else{
62                 lista->inicio=aux->siguiente;
63             }
64             //corrige los numeros de todos los nodos siguientes
65             Nodo sig=aux->siguiente;

```

```

66         while(sig!=NULL){
67             sig->nro=(sig->nro-1);
68             sig=sig->siguiente;
69         }
70         //elimina el nodo
71         delete aux->dato;
72         delete aux;
73         //ajusta el tamaño de la lista
74         lista->tamano=(lista->tamano-1);
75     }else{
76         cout << "No se encontro el elemento "<<nro<<" en la lista"<< endl;
77     }
78 }else{
79     cout << "La lista esta vacia"<< endl;
80 }
81 }
82
83
84 void eliminarLista(Lista lista){
85     Nodo aux=lista->inicio;
86     while(aux!=NULL){
87         delete aux->dato;
88         delete aux;
89         aux=aux->siguiente;
90     }
91     delete lista;
92 }
93
94 Nodo nodoSeleccionado(Lista lista,int nro){
95     Nodo aux=lista->inicio;
96     if(aux->nro==nro){
97         return aux;
98     }
99     while(aux!=NULL){
100         if(aux->nro==nro){
101             return aux;
102         }else{
103             aux=aux->siguiente;
104         }
105     }
106     cout << "No se encontro el elemento "<<nro<<" en la lista"<< endl;
107     return NULL;
108 }
109
110 bool listaVacia(Lista lista){
111     if(lista==NULL || lista->tamano==0 || lista->inicio==NULL){
112         return true;
113     }else{
114         return false;
115     }
116 }
117
118 Nodo ultimoNodo(Lista lista){
119     Nodo aux=lista->inicio;
120     while(aux->siguiente!=NULL){
121         aux=aux->siguiente;
122     }
123     return aux;
124 }
125
126 Nodo anteriorNodo(Lista lista,Nodo nodo){
127     Nodo anterior=lista->inicio;
128     while(anterior->siguiente!=nodo && anterior->siguiente!=NULL){
129         anterior=anterior->siguiente;
130     }
131     if(anterior->siguiente==nodo){

```

```

132         return anterior;
133     }else{
134         cout << "No se encontro el nodo anterior al nodo "<<nodo->nro<<" en la
lista"<< endl;
135         return NULL;
136     }
137 }
138
139 void mostrarLista(Lista lista){
140     Nodo nodo=lista->inicio;
141     cout << "Nodo nro:"<<nodo->nro<<" Dato:"<<(nodo->dato)<< endl;
142     while(nodo!=NULL){
143         cout << "Nodo nro:"<<nodo->nro<<" Dato:"<<(nodo->dato)<< endl;
144         nodo=nodo->siguiente;
145     }
146 }
147
148 int ultimoAnio(){
149     Lista elecciones = crearLista();
150     leerYpasarAseleccion(elecciones);
151     int anio = 0;
152     for(int i = 1; i <= elecciones->tamano; i++){
153         Nodo nodoInforMensual=nodoSeleccionado(elecciones,i);
154         infoMesual info = (infoMesual) nodoInforMensual->dato;
155         if(anio<getsAnioSeleccion(info)){
156             anio = getsAnioSeleccion(info);
157         }
158     }
159
160     return anio;
161 }

```