Header: TDAListaMemoria

```
//---------------------------------------------------------------------------

#ifndef TDAListaMemoriaH
#define TDAListaMemoriaH
//---------------------------------------------------------------------------
#endif

#include "SMemoria.h"



class TDAListaMemoria
{
    private:
    int Longitud;

  public:
    SMemoria m;

    int PtrElemento;

    TDAListaMemoria();
        bool vacia();
    int fin();
    int primero();
    int siguiente(int direccion);
    int anterior(int direccion);
    int longitud();
        int recupera(int direccion);

    void inserta(int direccion,int elemento);
        void modifica(int direccion,int elemento);

    void mostrar();

};
```

Implementacion: TDAListaMemoria

```
//---------------------------------------------------------------------------

#pragma hdrstop

#include "TDAListaMemoria.h"
#include <iostream>
//---------------------------------------------------------------------------
#pragma package(smart_init)


TDAListaMemoria::TDAListaMemoria()
{
  Longitud = 0;
    PtrElemento = NULO;
}

bool TDAListaMemoria::vacia()
{
    return (Longitud == 0);
}

int TDAListaMemoria::fin()
{
  int PtrFin;
  if(vacia())
  {
    throw("Error: Lista Vacia..");
  }else
  {
    int x = PtrElemento;
    while(x!=NULO)
    {
      PtrFin = x;
            x = m.obtener_dato(x,2);
    }

        return PtrFin;
  }

}

int TDAListaMemoria::primero()
{

        return PtrElemento;

}

int TDAListaMemoria::siguiente(int direccion)
{
  if(vacia())
  {
        throw("Error: Lista Vacia...");
  }else
  {
    if(direccion==fin())
    {
            throw("Error: Direccion Final...");
    }else
    {
            return m.obtener_dato(direccion,2);
        }
    }
}

int TDAListaMemoria::anterior(int direccion)
{
  if(vacia())
  {
        throw("Error: Lista Vacia...");
  }else
  {
    if(direccion==primero())
    {
            throw("Error: Direccion Inicial...");
```

```
76      }else
77      {
78        int x = PtrElemento;
79        int ant = NULO;
80        while(x!=NULO)
81        {
82          if(x==direccion)
83          {
84                      return ant;
85          }
86          ant = x;
87                  x = m.obtener_dato(x,2);
88              }
89          }
90      }
91  }
92
93  int TDAListaMemoria::longitud()
94  {
95      return Longitud;
96  }
97
98  int TDAListaMemoria::recupera(int direccion)
99  {
100   if(vacia())
101   {
102         throw("Error: Lista Vacia...");
103   }else
104   {
105         return m.obtener_dato(direccion,1);
106     }
107 }
108
109 void TDAListaMemoria::inserta(int direccion,int elemento)
110 {
111   int x = m.new_espacio(2);
112   if(x!=NULO)
113   {
114     m.poner_dato(direccion,1,elemento);
115     m.poner_dato(direccion,2,NULO);
116     if(vacia())
117     {
118       PtrElemento = x;
119             Longitud = 1;
120     }else
121     {
122       Longitud++;
123       if(direccion==primero())
124       {
125         m.poner_dato(x,2,direccion);
126                 PtrElemento = x;
127       }else
128       {
129         int ant = anterior(direccion);
130         m.poner_dato(ant,2,x);
131                 m.poner_dato(x,2,direccion);
132
133             }
134     }
135   }else
136   {
137         throw("Error: Existe Espacio en Memoria...");
138   }
139 }
140
141 void TDAListaMemoria::mostrar()
142 {
143
144 }
145
146
147 void TDAListaMemoria::modifica(int direccion, int elemento)
148 {
149   if(vacia())
150   {
151
```

```
152   }else
153   {
154         m.poner_dato(direccion,1,elemento);
155     }
156 }
```

Main de Proyecto 1

```
/*////////////////////////////////////////////////////////////////////////
  Tarea #3
  Docente: Mario Lopez Winnipeg
  U.A.G.R.M.
  Facultad  de Ingenieria en Ciencias de la Computacion y Telecomunicaciones
  Leonardo Anez Vladimirovna
*////////////////////////////////////////////////////////////////////////

#include <iostream.h>
#include <conio.h>
#include <fstream>
#include <string>
#include <vector>
#include <cstdlib>

#include "SMemoria.h"
#include "TDAListaVector.h"
#include "TDAListaMemoria.h"
#include "TDAListaPuntero.h"


using namespace std;


void MostrarOpciones()
{
  cout<<"Menu Proyecto 1:\n";
  cout<<"[1] Crear Memoria\n";
  cout<<"[2] Pedir Espacio\n";
  cout<<"[3] Liberar Espacio\n";
  cout<<"[4] Crear Lista\n";
  cout<<"[5] Poner en Direccion Inicial\n";
  cout<<"[6] Poner en Direccion Fin\n";
  cout<<"[7] Mostrar Memoria\n";
  cout<<"[8] Mostrar Lista\n";
  cout<<"[9] Salir\n";
  cout<<"Opcion: ";
}

int main()
{

  int opcion;
  int direccion,lugar,valor,cantidad,dato;
  TDAListaMemoria LM;

    do
  {
    // Refresh de la Consola
    system("cls");

        // Menu de Opciones
    MostrarOpciones();

    // Opcion del Menu
    cin>>opcion;

    switch(opcion)
    {
            // Crear
      case 1:
                    LM.m = SMemoria(); // Rellamada del Constructor
          break;
      // Pedir
      case 2:
          cout<<"Digite la Cantidad de Espacio:\n";
          cin>>cantidad;

          if(LM.m.espacio_disponible()>=cantidad)
          {
            cout<<LM.m.new_espacio(cantidad)<<endl;
          }else
          {
                        cout<<"No hay Suficiente Espacio\n";
                    }
```

```
76              break;
77          // Liberar
78          case 3:
79              cout<<"Digite el Espacio a Liberar:\n";
80              cin>>direccion;
81              LM.m.delete_espacio(direccion);
82              break;
83                // Poner Dato
84          case 4:
85
86                          LM = TDAListaMemoria();
87              break;
88          // Obtener Dato
89          case 5:
90              cin>>cantidad;
91              LM.inserta(LM.primero(),cantidad);
92
93              break;
94           // Mostrar
95          case 6:
96              cin>>cantidad;
97              LM.inserta(LM.primero(),cantidad);
98              break;
99          // Salir
100
101         case 7:
102             LM.m.mostrar();
103             break;
104         case 8:
105                         LM.mostrar();
106
107         default:
108             break;
109     }
110     getch();
111
112   }while(opcion!=9);
113
114
115   getch();
116
117   return 0;
118 }
```

Links a los Codigos:

https://hastebin.com/pelanuhasu.cpp
https://hastebin.com/uxarozixas.cpp
https://hastebin.com/sawajibuvu.cpp