

15-9-2021



Lista estatica

Practica: 2

Materia: Estructura de datos

Sección: D01.

Código: 216584703

Carrera: Ingeniería en computación.

Nombre alumno: Padilla Pérez Jorge
Daray

Nombre profesor: Julio Esteban
Valdes Lopez

Introducción

Mi práctica consiste en la implementación de una lista estructurada en la cual se implementan lo típico de un TDA Lista lo que viene siendo insertar en orden como si fuera una lista consultar elemento remover etc. En esta practica se logro hacer todo lo pedido para esta, además de poder realizar la función de localizar posición por elemento la cual me costó más, pero si se logró.

También se pudo lograr recorrer los datos una vez ingresados en la lista, aunque no se implementaron las funciones anterior y siguiente creo que es una función fácil de comprender y de hacer por lo que me parece bien que se hayan saltado esas 2 funciones que en si no existe una utilidad buena para ellos.

Pantallazos

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
```

Aquí se aprecia el menú para el usuario la verdad que no se explica mucho más que ahí se ven todas las funciones del programa.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
1
En que posicion desea insertar el elemento :
0
Inserte elemento:
500
```

Aquí insertamos en la posición 0 ya que al ser una lista no podemos insertar más allá del primer índice, claro si es que esta vacía.

También a la derecha se ve que pasa si se inserta en otra posición que no es la correcta.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
1
En que posicion desea insertar el elemento :
40
Inserte elemento:
300
Ingresa un elemento consecutivo valido
Presione una tecla para continuar . . .
```

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
5
Posicion numero: 0
Dato: 300
Posicion numero: 1
Dato: 500
Posicion numero: 2
Dato: 500
Continuar 1 salir 7:
```

La función imprimir no tiene mas chiste que imprimir los datos registrados.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir lista
6) Anular lista
7) Salir
Seleccione opcion:
2
Que posicion deseas eliminar :
1
Continuar 1 salir 7:
```

Si eliminamos la posición 1 en esta caso 500 al imprimirlo queda en la posición 2 el 500 pero la posición 3 desaparece.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir lista
6) Anular lista
7) Salir
Seleccione opcion:
5
Posicion numero: 0
Dato: 300
Posicion numero: 1
Dato: 500
Continuar 1 salir 7:
```

Aquí se aprecia lo anterior.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
3
Que posicion deseas consultar :
1
Posicion numero: 1
Dato: 500

Continuar 1 salir 7:
```

aquí recuperamos el elemento por posición en este caso buscamos la posición 1 y no arroja el dato 500 almacenado ahí

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"

Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
4
Que dato deseas buscar :
500
Posicion numero: 1
Dato: 500

Continuar 1 salir 7:
```

En este caso parece igual a recupera, pero aquí se localiza la posición a través del dato en este caso pide el dato yo pongo 500 y como ya sabíamos está en la posición 1 que ahí aparece.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"
Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
6
Continuar 1 salir 7:
```

Al momento de anular la lista pues se eliminan todos los índices y lo que contenían estos mismos.

```
"D:\Trabajos Seda\Practicas\Practica 2 Lista Estatica\main.exe"
Practica 2
1) Insertar elemento (Por posicion)
2) Eliminar elemento
3) recupera elemento
4) localiza posicion
5) Imprimir Lista
6) Anular Lista
7) Salir
Seleccione opcion:
5
La lista esta vacia
Presione una tecla para continuar . . .
```

Aquí se aprecia mejor eso de anular la lista la cual al quererla imprimirla arroja el mensaje la lista esta vacia. .

Conclusión

Respecto a la realización del código concluyo que se logró bien la realización de este programa ya que a lo que se puede apreciar funciona de manera correcta como debería hacerlo una lista además de implementar bien sus funciones recalco mis conocimientos, y estoy abierto a posibles errores que pueda tener el programa al momento de que el profe la evalúe.

También se concluye que aunque no parezca que tiene una utilidad buena al momento de referirse a trabajo si lo piensas un rato le hayas muchas utilidades importantes tanto que pensándolo muchas aplicaciones y juegos realizan una lista implementada en estos mismos por lo cual espero poder aprender mas sobre las listas y los tipos de datos abstractos

Código fuente

```
1  #include <iostream>
2  #include <cstring>
3
4  #define TAMLISTA 10
5
6  using namespace std;
7
8  void menu();
9
10 typedef int tipo_dato;
11
12 struct Lista{
13     tipo_dato datos[TAMLISTA];
14     void inicializa();
15     bool vacia();
16     bool llena();
17     void insertar(int pos, tipo_dato elem);
18     void elimina(int pos);
19     int ultimo;
20     int primero;
21     void localiza(tipo_dato dato);
22     void recupera(int pos);
23     void imprimir();
24     void anular();
25
26     Lista(){
27         inicializa();
28     }
```



```
29  };
30
31  void Lista::inicializa(){
32      ultimo = -1;
33      primero = 0;
34  }
35
36  bool Lista::vacia(){
37      return ultimo == -1;
38  }
39
40  bool Lista::llena(){
41      return ultimo == TAMLISTA - 1;
42  }
43
44
45  void Lista::insertar(int pos, tipo_dato elem){
46
47      if (llena() || pos < 0 || pos > ultimo + 1){
48
49          cout<<"Ingresa un elemento consecutivo valido"<<endl;
50          system("pause");
51          return;
52      }
53
54      for(int i = ultimo+1 ; i > pos ; i-- ){
55
56          datos[i] = datos[i - 1];
57      }
58
```

```
59         datos[pos] = elem;
60
61     ultimo++;
62 }
63
64
65 void Lista::elimina(int pos){
66     if (vacía() || pos < 0 || pos > ultimo ){
67
68         cout<<"La lista esta vacía"<<endl;
69         system("pause");
70
71         return;
72     }
73
74     for (int i = pos ; i < ultimo ; i++){
75
76         datos[i] = datos[i + 1];
77
78         ultimo--;
79     }
80 }
81
82 void Lista::imprimir(){
83     if (vacía()){
84
85         cout<<"La lista esta vacía"<<endl;
86         system("pause");
87         return;
88     }
```

```
89     for(int i = primero ; i <= ultimo ; i++){
90         cout<<"Posicion numero: "<<i<<"\n Dato: "<<datos[i]<<" "<<endl;
91     }
92 }
93
94 void Lista::recupera(int pos){
95     if (vacía() || pos < 0 || pos > ultimo ){
96
97         cout<<"La lista esta vacía"<<endl;
98         system("pause");
99
100         return;
101     }
102
103     cout<<"Posicion numero: "<<pos<<"\n Dato: "<<datos[pos]<<" "<<endl;
104     cout<<endl;
105 }
106
107 void Lista::localiza(tipo_dato dato){
108     if (vacía()){
109
110         cout<<"La lista esta vacía"<<endl;
111         system("pause");
112
113         return;
114     }
115
116     int i;
117     for (i=primero;i<=ultimo;i++){
118         if (datos[i] == dato){
```

```

119             cout<<"Posicion numero: "<<i<<"\n Dato: "<<datos[i]<<" "<<endl;
120             cout<<endl;
121         }
122     }
123 }
124
125 void Lista::anular(){
126     ultimo = -1;
127 }
128
129
130 struct Lista mi_lista;
131 int main(){
132     int opc=0;
133     do{
134         system("cls");
135         menu();
136         cout<<"Continuar 1 salir 7:"<<endl;cin>>opc;
137     }while(opc!=7);
138     system("pause>>cls");
139     return 0;
140 }
141
142 void menu(){
143     int opc=0;
144     int dato,pos;
145     cout<<" Practica 2"<<endl;
146     cout<<"1) Insertar elemento (Por posicion)"<<endl;
147     cout<<"2) Eliminar elemento"<<endl;
148     cout<<"3) recupera elemento"<<endl;

```

```
149         cout<<"4) localiza posicion"<<endl;
150         cout<<"5) Imprimir Lista"<<endl;
151         cout<<"6) Anular Lista"<<endl;
152         cout<<"7) Salir"<<endl;
153         cout<<"Seleccione opcion:"<<endl;
154         cin>>opc;
155         switch(opc){
156
157             case 1: {
158                 cout<<"En que posicion desea insertar el elemento :"<<endl;cin>>pos;
159                 cout<<"Inserte elemento: "<<endl;cin>>dato;
160                 mi_lista.insertar(pos,dato);
161
162                 }break;
163             case 2: {
164                 cout<<"Que posicion deseas eliminar :"<<endl;cin>>pos;
165                 mi_lista.elimina(pos);
166                 }break;
167             case 3: {
168                 cout<<"Que posicion deseas consultar :"<<endl;cin>>pos;
169                 mi_lista.recupera(pos);
170                 }break;
171
172             case 4: {
173                 cout<<"Que dato deseas buscar :"<<endl;cin>>dato;
174                 mi_lista.localiza(dato);
175                 }break;
176
177             case 5: {
178                 mi_lista.imprimir();
```

```
179         }break;
180
181         case 6: {
182             mi_lista.anular();
183             }break;
184
185         case 7:break;
186
187         default:
188             cout<<"La opcion: "<<opc<<"No existe"<<endl;
189     }
190 }
```