

OPERACIONES CON LISTAS ENLAZADAS

```
1  #include<iostream>
2  using namespace std;
3  struct nodo
4  {
5      char nomb[20];
6      nodo *sig;
7  };
8  nodo *crea_i(nodo *);
9  nodo *crea_f(nodo *);
10 void recorre(nodo *);
11 nodo *inserta_i(nodo *);
12 nodo *inserta_f(nodo *);
13 nodo *inserta_ad(nodo *,char *);
14 nodo *inserta_dd(nodo *,char *);
15 nodo *eliminar_p(nodo *);
16 nodo *eliminar_u(nodo *);
17 nodo *eliminar_x(nodo *,char *);
18 nodo *eliminar_ad(nodo*, char *);
19 nodo *eliminar_dd(nodo*, char *);
20 int main()
21 {
22     char codigo[10];
23     nodo *p=NULL;
24     int op;
25     do
26     {
27         system("cls");
28         cout<<"\nLISTAS ENLAZADAS";
29         cout<<"\n===== ";
30         cout<<"\n\n1. CREAR POR EL INICIO";
31         cout<<"\n\n2. CREAR POR EL FINAL";
32         cout<<"\n\n3. INSERTAR AL INICIO";
33         cout<<"\n\n4. INSERTAR AL FINAL";
34         cout<<"\n\n5. INSERTAR ANTES DE";
35         cout<<"\n\n6. INSERTAR DESPUES DE";
36         cout<<"\n\n7. ELIMINA EL PRIMER NODO";
37         cout<<"\n\n8. ELIMINAR EL ULTIMO NODO";
38         cout<<"\n\n9. ELIMINAR UN NODO ESPECIFICO";
39         cout<<"\n\n10. ELIMINAR EL NODO ANTERIOR A ";
40         cout<<"\n\n11. ELIMINAR EL NODO POSTERIOR A ";
41         cout<<"\n\n12. M O S T R A R ";
42         cout<<"\n\n13. S A L I R";
43         cout<<"\n\n\t\t\t OPCION: ";
44         cin>>op;
45         switch(op)
46         {
47             case 1:
48                 system("cls");
49                 p= NULL;
50                 p=crea_i(p);
51                 break;
52             case 2:
```

```

53         system("cls");
54         p=NULL;
55         p=crea_f(p);
56         break;
57     case 3:
58         system("cls");
59         p=inserta_i(p);
60         break;
61     case 4:
62         system("cls");
63         p=inserta_f(p);
64         break;
65     case 12:
66         system("cls");
67         recorre(p);
68         system("pause");
69         break;
70     case 13:
71         break;
72     }
73 }
74 while(op>0&&op<13);
75 return 0;
76 }
77
78 nodo *crea_i(nodo *p)

```

```

79 {
80     nodo *q;
81     char op;
82     do
83     {
84         system("cls");
85         q=new(nodo);
86         cout<<"\n\nNOMBRE    :  ";
87         cin>>q->nomb;
88         q->sig=p;
89         p=q;
90         cout<<"\n\n\t\t\tMAS DATOS (S/N)?  ";
91         cin>>op;
92     }
93     while(op=='s' || op=='S');
94     return(p);
95 }
96 nodo *crea_f(nodo *p)
97 {
98     nodo *q,*r;
99     char op;
100     do
101     {
102         system("cls");
103         q=new(nodo);
104         cout<<"\n\nNOMBRE    :  ";

```

```

105         cin>>q->nomb;
106         q->sig=NULL;
107         if(p==NULL)
108             p=q;
109         else
110         {
111             r->sig=q;
112         }
113         r=q;
114         cout<<"\n\n\t\t\tMAS DATOS (S/N)?  ";
115         cin>>op;
116     }
117     while(op=='s' || op=='S');
118     return(p);
119 }
120 nodo *inserta_i(nodo *p)
121 {
122     nodo *q;
123     system("cls");
124     cout<<"\n\nINSERTANDO DATO POR EL INICIO ... ";
125     q=new(nodo);
126     cout<<"\n\nNOMBRE      :  ";
127     cin>>q->nomb;
128     q->sig=p;
129     p=q;
130     return(p);
131 }
132 nodo *inserta_f(nodo *p)
133 {
134     nodo *q,*r;
135     system("cls");
136     cout<<"\n\nINSERTANDO DATO POR EL FINAL ... ";
137     q=new(nodo);
138     cout<<"\n\nNOMBRE      :  ";
139     cin>>q->nomb;
140     q->sig=NULL;
141     r=p;
142     if(p==NULL)
143         p=q;
144     else
145     {
146         while(r->sig!=NULL)
147             r=r->sig;
148         r->sig=q;
149     }
150     return(p);
151 }
152 void recorre(nodo *p)
153 {
154     nodo *q;
155     system("cls");
156     q=p;

```

```
157 cout<<"\n\n\tLISTA DE DATOS:\n";
158 cout<<"\t=====";
159 while(q!=NULL)
160 {
161     cout<<"\n\n\t\t"<<q->nomb;
162     q=q->sig;
163 }
164 cout<<"\n\n\t\t\t";
165 }
```