

```

1  #include <iostream>
2  #include "rankingVinos.h"
3  #include "lista.h"
4  #include "infoMensual.h"
5  #include "vino.h"
6  #include "RankingVarietales.h"
7  #include <iomanip>
8  #include <string.h>
9
10
11  using namespace std;
12
13  void rankingPorVinos (Lista seleccionados, Lista vinos)
14  {
15      Lista ranking = crearLista();
16      int ultimo = ultimoAnio();
17
18      for(int i=1; i<= seleccionados->tamano ; i++)
19      {
20          Nodo nodoInforMensual=nodoSeleccionado(seleccionados,i);
21          infoMesual info = (infoMesual) nodoInforMensual->dato;
22          if(getsAnioSeleccion(info) == ultimo){
23              Vino vino1 = traerVino(vinos, getsVino1(info));
24              Vino vino2 =traerVino(vinos, getsVino2(info));
25              Vino vino3 =traerVino(vinos, getsVino3(info));
26              Vino vino4 =traerVino(vinos, getsVino4(info));
27              Vino vino5 =traerVino(vinos, getsVino5(info));
28              Vino vino6 =traerVino(vinos, getsVino6(info));
29
30              agregarVinos(ranking,vino1);
31              agregarVinos(ranking,vino2);
32              agregarVinos(ranking,vino3);
33              agregarVinos(ranking,vino4);
34              agregarVinos(ranking,vino5);
35              agregarVinos(ranking,vino6);
36
37          }
38      }
39
40      Lista ordenada=crearLista();
41      ordenarListaAux(ranking,ordenada);
42
43      mostrarListaRankigVino(ordenada);
44  }
45
46  RankingVinos buscarVino(Lista lista,int idVino)
47  {
48      Nodo aux=lista->inicio;
49      int encontrado=0;
50      while(aux!=NULL && encontrado==0)
51      {
52          RankingVinos rankig=(RankingVinos) aux->dato;
53          Vino vino=rankig->vino;
54          if(getsVino(vino)==idVino)
55          {
56              encontrado=1;
57          }
58          else
59          {
60              aux=aux->siguiente;
61          }
62      }
63      if(encontrado==1)
64      {
65          return (RankingVinos)aux->dato;
66      }

```

```

67     else
68     {
69         return NULL;
70     }
71 }
72
73 void agregarVinos(Lista ranking, Vino vino){
74
75     RankingVinos rankingDato=buscarVino(ranking, getsVino(vino));
76     if (rankingDato!=NULL)
77     {
78         rankingDato->cantidadSelecciones+=1;
79     }
80     else
81     {
82         rankingDato=new RankingVinosStruct();
83         rankingDato->cantidadSelecciones=1;
84         rankingDato->vino=vino;
85         agregarNodo(ranking, rankingDato);
86     }
87
88
89 }
90
91 void ordenarListaAux(Lista lista, Lista ordenada)
92 {
93     Nodo actual;
94     Nodo siguiente;
95     actual=lista->inicio;
96
97     Nodo auxNodo;
98
99     int nro;
100    int i=1;
101
102    while (actual!=NULL )
103    {
104        nro=actual->nro;
105        siguiente=actual->siguiente;
106        RankingVinos rankig1=(RankingVinos)actual->dato;
107        while( siguiente !=NULL )
108        {
109
110            RankingVinos rankig2=(RankingVinos)siguiente->dato;
111
112            if(rankig1->cantidadSelecciones<rankig2->cantidadSelecciones)
113            {
114                rankig1=rankig2;
115                nro=siguiente->nro;
116            }
117            siguiente=siguiente->siguiente;
118
119        }
120        RankingVinos nuevo=new RankingVinosStruct;
121        auxNodo=nodoSeleccionado(lista, nro);
122        RankingVinos vinoAnt=(RankingVinos)auxNodo->dato;
123        nuevo->cantidadSelecciones=vinoAnt->cantidadSelecciones;
124        nuevo->vino=vinoAnt->vino;
125
126        agregarNodo(ordenada, nuevo);
127        eliminarNodo(lista, nro);
128
129        actual=lista->inicio;
130        i++;
131
132    }

```

```

133
134 }
135
136
137
138
139 void mostrarListaRankigVino(Lista lista){
140     int j=1;
141     cout << setw( 80 ) << setfill( '-' ) << '\n' << setfill( ' ' );
142     cout << " | " << left << setw( 14 ) << "Posicion" << " | " << right << setw( 1 ) << "
Vino" << right << setw( 9 )
143     << " | " << "Etiqueta" << right << setw(7) << " | " << "Anio" << right << setw(7) <<
" | " << "Cantidad Seleccionado" << " | " << endl;
144     cout << setw( 80 ) << setfill( '-' ) << '\n' << setfill( ' ' ) << '\n';
145     while(j<=lista->tamano){
146
147         Nodo nodoRanking=nodoSeleccionado(lista,j);
148         RankingVinos rankingDato=(RankingVinos)nodoRanking->dato;
149         Vino vino1=rankingDato->vino;
150         int resultado=strlen(getsetiqueta(vino1)); //cuento El tamaño de la cadena
para despues poder restar
151
152         cout << "Nro en lista: " << nodoRanking->nro << " -Vino " << getsVino(vino1) << " :
" << getsetiqueta(vino1) << right << setw(16-resultado) << " " << getsAnioCosecha(vino1) << right
<< setw(7) << " " << " Cant seleccionada: " << rankingDato->cantidadSelecciones << endl;
153         j++;
154     }
155
156     cout << "\n" << endl;
157
158 }

```