

```
1  /*
2  * Proyecto: Grafo
3  * Archivo:  Ciudad.h
4  * Autor:    J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 19 de junio de 2024, 08:43 AM
7  */
8
9
10 #ifndef CIUDAD_H
11 #define CIUDAD_H
12 #include <string>
13 using namespace std;
14
15 class Ciudad {
16 private:
17     string nombre;
18 public:
19     Ciudad();
20     Ciudad(string nombre);
21     bool operator == (const class Ciudad &orig) const;
22     bool operator < (const class Ciudad &orig) const;
23     void imprime() const;
24 };
25
26 #endif /* CIUDAD_H */
27
28 /*
29 * Proyecto: Grafo
30 * Archivo:  Ciudad.cpp
31 * Autor:    J. Miguel Guanira E. //miguel.guanira.
32 *
33 * Created on 19 de junio de 2024, 08:43 AM
34 */
35
36 #include <iostream>
37 #include <iomanip>
38 using namespace std;
39
40 #include "Ciudad.h"
41
42 Ciudad::Ciudad() {
43     nombre.clear();
44 }
45
46 Ciudad::Ciudad(string nombre) {
47     this->nombre=nombre;
48 }
49
50 bool Ciudad::operator<(const class Ciudad &orig) const{
51     return nombre.compare(orig.nombre)<0;
52 }
53
54 bool Ciudad::operator==(const class Ciudad &orig) const {
55     return nombre.compare(orig.nombre)==0;
56 }
57
58 void Ciudad::imprime() const {
59     cout<<nombre<<endl;
60 }
61
62 /*
63 * Proyecto: Grafo
64 * Archivo:  Arista.h
65 * Autor:    J. Miguel Guanira E. //miguel.guanira.
66 */
```

```
67  * Created on 19 de junio de 2024, 08:51 AM
68  */
69
70
71  #ifndef ARISTA_H
72  #define ARISTA_H
73  #include "Ciudad.h"
74
75  class Arista {
76  private:
77      double longitud;
78      class Ciudad destino;
79  public:
80      Arista();
81      Arista(double longitud, const class Ciudad &destino);
82      void imprimeArista() const;
83
84  };
85
86  #endif /* ARISTA_H */
87
88  /*
89   * Proyecto: Grafo
90   * Archivo:  Arista.cpp
91   * Autor:    J. Miguel Guanira E. //miguel.guanira.
92   *
93   * Created on 19 de junio de 2024, 08:51 AM
94   */
95
96  #include <iostream>
97  #include <iomanip>
98  using namespace std;
99
100  #include "Arista.h"
101  #include "Ciudad.h"
102
103  Arista::Arista() {
104      longitud=0;
105  }
106
107  Arista::Arista(double longitud, const class Ciudad &destino) {
108      this->longitud = longitud;
109      this->destino = destino;
110  }
111
112  void Arista::imprimeArista() const{
113      cout.precision(2);
114      cout<<fixed;
115      cout<<"Longitud: "<<setw(10)<<longitud<<" Destino: ";
116      destino.imprime();
117  }
118
119  /*
120   * Proyecto: Grafo
121   * Archivo:  GrafoDirigido.h
122   * Autor:    J. Miguel Guanira E. //miguel.guanira.
123   *
124   * Created on 19 de junio de 2024, 09:03 AM
125   */
126
127
128  #ifndef GRAFODIRIGIDO_H
129  #define GRAFODIRIGIDO_H
130  #include <map>
131  #include <vector>
132  using namespace std;
```

```
133 #include "Ciudad.h"
134 #include "Arista.h"
135
136 class GrafoDirigido {
137 private:
138     map<Ciudad,vector<Arista>> adyacencias;
139 public:
140     void agregarArista(const class Ciudad &origen,
141                       const class Arista &destino);
142     void imprimeGrafo();
143 };
144
145 #endif /* GRAFODIRIGIDO_H */
146
147 /*
148  * Proyecto: Grafo
149  * Archivo: GrafoDirigido.cpp
150  * Autor: J. Miguel Guanira E. //miguel.guanira.
151  *
152  * Created on 19 de junio de 2024, 09:03 AM
153  */
154
155 #include <iostream>
156 #include <iomanip>
157 using namespace std;
158
159 #include "GrafoDirigido.h"
160
161 void GrafoDirigido::agregarArista(const class Ciudad &origen,
162                                   const class Arista &destino) {
163     adyacencias[origen].push_back(destino);
164 }
165
166 void GrafoDirigido::imprimeGrafo() {
167     for(auto c: adyacencias){
168         class Ciudad ciudad = c.first;
169         cout<<"Ciudad: ";
170         ciudad.imprime();
171         cout<<"Conectada con: "<<endl;
172         for(auto v: c.second)
173             v.imprimeArista();
174     }
175 }
176
177 /*
178  * Proyecto: Grafo
179  * Archivo: main.cpp
180  * Autor: J. Miguel Guanira E.//miguel.guanira.
181  *
182  * Created on 19 de junio de 2024, 08:39 AM
183  */
184
185 #include <iostream>
186 #include <iomanip>
187 using namespace std;
188 #include "Ciudad.h"
189 #include "Arista.h"
190 #include "GrafoDirigido.h"
191
192 int main(int argc, char** argv) {
193     class Ciudad ciudadA("Ciudad A");
194     class Ciudad ciudadB("Ciudad B");
195     class Ciudad ciudadC("Ciudad C");
196     class Ciudad ciudadD("Ciudad D");
197     class Ciudad ciudadE("Ciudad E");
198     class Ciudad ciudadF("Ciudad F");
```

```
199
200     class Arista aristaAB(100,ciudadB);
201     class Arista aristaAC(200,ciudadC);
202     class Arista aristaBC(175,ciudadC);
203     class Arista aristaCD(83,ciudadD);
204     class Arista aristaFC(155,ciudadC);
205     class Arista aristaFD(130,ciudadD);
206     class Arista aristaFE(125,ciudadE);
207     class Arista aristaEF(90,ciudadF);
208     class Arista aristaEA(110,ciudadA);
209
210
211     class GrafoDirigido grafo;
212
213     grafo.agregarArista(ciudadA,aristaAB);
214     grafo.agregarArista(ciudadA,aristaAC);
215     grafo.agregarArista(ciudadB,aristaBC);
216     grafo.agregarArista(ciudadC,aristaCD);
217     grafo.agregarArista(ciudadE,aristaEA);
218     grafo.agregarArista(ciudadE,aristaEF);
219     grafo.agregarArista(ciudadF,aristaFC);
220     grafo.agregarArista(ciudadF,aristaFD);
221     grafo.agregarArista(ciudadF,aristaFE);
222
223
224     grafo.imprimeGrafo();
225
226     return 0;
227 }
228
```