

```
1  #include <iostream>
2
3  using namespace std;
4  struct tarea {
5      struct tarea * link;
6      struct tiempo* lista_de_tiempo;
7      int id_tarea;
8      int fecha_inicio;
9
10 };
11
12 typedef tarea Ngrafo;
13
14 struct tiempo {
15     int tiempo_tarea;
16     int precio;
17     struct tarea *destino;
18     struct tarea *origen;
19     struct tiempo * link;
20 };
21 typedef tiempo Narco;
22
23
24 void grafo_agregar_nodo (Ngrafo* &lista_n, int id)
25 {
26     Ngrafo* aux = lista_n;
27
28     // Verificamos si que no exista ya el id_nodo.
29     while (aux != NULL)
30     {
31         if (aux->id_tarea == id)
32         {
33             cout << "!!! Error: Ya existe un nodo con ese id_nodo." << endl;
34             return;
35         }
36         aux = aux->link;
37     }
38
39     aux = new (Ngrafo);
40     aux->id_tarea = id;
41     aux->lista_de_tiempo = NULL;
42     aux->link = lista_n;
43     lista_n = aux;
44 }
45
46 void grafo_agregar_arco (Ngrafo* lista_n, int id, int id_nodo_origen, int id_nodo_destino)
47 {
48     Ngrafo* nodo_origen = lista_n;
49     Ngrafo* nodo_destino = lista_n;
50
51     while (nodo_origen != NULL && nodo_origen->id_tarea != id_nodo_origen)
52         nodo_origen = nodo_origen->link;
53
54     while (nodo_destino != NULL && nodo_destino->id_tarea != id_nodo_destino)
55         nodo_destino = nodo_destino->link;
56
57     // Verificamos si existen id_nodo_origen y id_nodo_destino.
58     if (nodo_origen == NULL || nodo_destino == NULL)
59     {
60         cout << "!!! Error: Alguno de los nodos no existe." << endl;
61         return;
62     }
63
64     // Verificamos que no exista ya el id_arco.
65     // Tarea :) Ponga aquí su código.
66
67     Narco* aux = new (Narco);
68     aux->tiempo_tarea = id;
69     aux->destino = nodo_destino;
70     aux->link = nodo_origen->lista_de_tiempo;
71     nodo_origen->lista_de_tiempo = aux;
72
73 }
74
75 void grafo_mostrar (Ngrafo* lista_n)
76 {
77     cout << "Grafo:\n\n";
78
79     while (lista_n != NULL)
80     {
81         cout << "Nodo " << lista_n->id_tarea << ":" << endl;
82
83         Narco* aux = lista_n->lista_de_tiempo;
84         while (aux != NULL)
85         {
86             cout << " Arco " << aux->tiempo_tarea << " -> Nodo " << aux->destino->id_tarea << endl;
```

```
87         aux= aux->link;
88     }
89 }
90     lista_n = lista_n->link;
91 }
92     cout << endl;
93     cout << endl;
94 }
95
96 void menu_opcion1 (Ngrafo* lista_n)
97 {
98     grafo_mostrar (lista_n);
99 }
100 void menu_opcion2 (Ngrafo* &lista_n)
101 {
102     int id;
103
104     cout << "Ingrese el id_nodo del nodo que desea incorporar: ";
105     cin >> id;
106     grafo_agregar_nodo (lista_n, id);
107     cout << endl;
108     cout << endl;
109 }
110 void menu_opcion3 (Ngrafo* &lista_n)
111 {
112     int id, id_nodo_origen, id_nodo_destino;
113
114     cout << "Ingrese el id_arco del arco que desea incorporar: ";
115     cin >> id;
116     cout << "Ingrese el id_nodo del nodo desde donde sale el arco: ";
117     cin >> id_nodo_origen;
118     cout << "Ingrese el id_nodo del nodo a donde llega el arco: ";
119     cin >> id_nodo_destino;
120
121     grafo_agregar_arco (lista_n, id, id_nodo_origen, id_nodo_destino);
122
123     cout << endl;
124     cout << endl;
125 }
126
127 int main(int argc, char *argv[])
128 {
129
130     Ngrafo* lista_n = NULL;
131     bool c;
132     int opcion = 0;
133     do {
134         cout << "*****Menu de Opciones*****\n";
135         cout << endl;
136         cout << "***** Grafos *****\n";
137         cout << endl;
138         cout << "1- Mostrar.\n";
139         cout << "2- Insertar Nodo.\n";
140         cout << "3- Insertar Arco.\n";
141         cout << "4- Eliminar viaje.\n";
142         //cout << "5- Contar cantidad de arcos y nodos.\n";
143         cout << endl;
144         cout << "0- Salir\n";
145         cout << endl;
146         cout << "Ingrese opcion: ";
147         cin >> opcion;
148         cout << endl;
149         cout << endl;
150
151         switch(opcion)
152         {
153             case 1:
154                 menu_opcion1 (lista_n);
155             break;
156             case 2:
157                 menu_opcion2 (lista_n);
158             break;
159             case 3:
160                 menu_opcion3 (lista_n);
161             break;
162             case 4:
163                 break;
164             case 5:
165                 break;
166         }
167     } while (opcion != 0);
168     return 0;
169 }
170
171
172
173 }
```