```
//// Crear archivo: Grafo.hpp

#include <vector>
using namespace std;

template<class T,T vacio=-1>
class CGrafo {
private:
        class CArco {
        public:
                T info;
                int v; //indice del vertice de llegada
                CArco(int vLlegada) {
                        info = vacio;
                        v = vLlegada;
                }
        };
        class CVertice {
        public:
                T info;
                vector<CArco*>* ady; //Lista de adyacencia
                CVertice() {
                        info = vacio;
                        ady = new vector<CArco*>();
                }
        };
        //Lista de vértices
        vector<CVertice*>* vertices;

public:
        CGrafo() {
                vertices = new vector<CVertice*>();
        }

        //Operaciones del Grafo
        int adicionarVertice(T info) {
                CVertice * vert = new CVertice();
                vert->info = info;
                vertices->push_back(vert);
                return vertices->size() - 1;
        }

        int cantidadVertices() {
                return vertices->size();
        }
```

```
        T obtenerVertice(int v) {
                return (vertices->at(v))->info;
        }
        void modificarVertice(int v, T info) {
                (vertices->at(v))->info = info;
        }
        //Operaciones del arco
        int adicionarArco(int v, int vLlegada) {
                CVertice* ver = vertices->at(v);
                //Crear el objeto ARCO
                CArco* arc = new CArco(vLlegada);
                ver->ady->push_back(arc);
                return ver->ady->size() - 1;
        }

        int cantidadArcos(int v) {
                return (vertices->at(v))->ady->size();
        }

        T obtenerArco(int v, int apos) {
                CVertice* ver = vertices->at(v);
                return (ver->ady->at(apos))->info;
        }

        void modificarArco(int v, int apos, T info) {
                CVertice* ver = vertices->at(v);
                (ver->ady->at(apos))->info = info;
        }

        int obtenerVerticeLlegada(int v, int apos) {
                CVertice* ver = vertices->at(v);
                return (ver->ady->at(apos))->v; //indice del vertice de
llegada
        }
};


**************************************


//Crear archivo consola CPP

#include <iostream>
#include "Grafo.hpp"

using namespace std;
```

```
int main() {
	//Crear el grafo
	CGrafo<int>* G = new CGrafo<int>();

	//Agregar Vértices
	G->adicionarVertice(2); //indice=0
	G->adicionarVertice(15); //indice=1
	G->adicionarVertice(30); //indice=2
	G->adicionarVertice(7); //indice=3

	//Agregar los arcos
	G->adicionarArco(0, 1); //indice=0
	G->modificarArco(0, 0, 10);
	G->adicionarArco(0, 3);//indice=1
	G->modificarArco(0, 1, 20);
	G->adicionarArco(1, 2);//indice=0
	G->modificarArco(1, 0, 30);
	G->adicionarArco(2, 0);//indice=0
	G->modificarArco(2, 0, 40);

	//Imprimir los vértices con sus arcos
	for (int i = 0; i < G->cantidadVertices(); ++i) {
		cout << "Vertice : " << G->obtenerVertice(i)<<endl;
		for (int j = 0; j < G->cantidadArcos(i); j++)
		{
			cout << "Arco->" << G->obtenerArco(i, j) << " ";
		}
		cout << endl;
	}


	cin.get();
	return 0;
}
```