



# Estructuras de Datos

GRAFOS

ACTIVIDAD A-5.1: CAMINO MÁS CORTO

René Ornelis

Vacaciones de junio de 2024

# Camino más corto

## 1 Objetivos

Los objetivos de esta actividad son que el estudiante sea capaz de:

1. Aplicar los grafos en una necesidad de uso frecuente.

## 2 Problema

En el vasto campo de la teoría de grafos, el problema del camino más corto es uno de los temas fundamentales y aplicativos. Este problema esencial se centra en encontrar la ruta más eficiente entre dos nodos específicos dentro de un grafo, donde los nodos representan ubicaciones o puntos de interés, y las aristas indican las conexiones o caminos entre ellas.

La relevancia del problema del camino más corto se extiende a diversas disciplinas, desde la planificación de rutas en redes de transporte y logística, hasta la optimización de rutas de comunicación en redes de computadoras. Además, tiene aplicaciones significativas en la modelización de fenómenos en la vida real, como la propagación de información, el flujo de recursos o la planificación de proyectos.

Existen varios algoritmos diseñados para abordar este desafío, cada uno con sus propias características y eficiencias. Entre ellos está el algoritmo de Dijkstra el cual deberá implementarse.

Dado un grafo dirigido que está representado así:

```
struct {  
    char nombre[50];  
    int valor;  
} Vertice;  
  
class Grafo {  
public:  
    Grafo();  
    ~Grafo();  
  
    List<Vertice> rutaMasCorta(int origen, int destino);  
  
private:  
    Vertice V[n];  
    int ma [n,n];  
};
```

Se tiene el arreglo de vértices  $V[n]$  el cual contiene la información de todos los vértices de la matriz de adyacencia  $ma[n,n]$  en la cual  $ma[i,j]$  indica con un entero mayor o igual a cero el costo del lado que va del vértice  $i$  al vértice  $j$  y -1 si dicho lado no existe.

Suponiendo que dicho grafo ya está definido, implementar el método *rutaMasCorta()* que cree una lista con los vértices, en el orden que se deben recorrer, que una el vértice *origen* con el *destino* y sea la ruta más corta entre estos.

## 2.1 Restricciones

- No se permite agregar variables a las clases
- No se permite cambiar la visibilidad de los miembros de la clase
- Cualquier método que invoque en su solución, debe desarrollarlo.
- No se permite cambiar la interfaz de los métodos público.

## 3 Tiempo de entrega

La entrega se debe realizar en la plataforma de la Facultad o, en caso de que esta no esté disponible, por correo electrónico, a más tardar el 24/junio a las 09:00. No se permitirá entregas posteriores al límite definido.

## 4 Entregables

Deberá entregar archivo ZIP con:

1. Archivo .h y .cpp compilables con la definición de clases y la implementación de método solicitado.

## 5 Criterios de evaluación

Objetivo	Puntos	Detalle por evaluar
Código compilable	1	Se entrega código fuente compilable
Eficacia	2	La solución funciona
Eficiencia	1	Es la solución óptima
TOTAL	4	