

## Tema 9

### Grafos

El objetivo de este ejercicio es aplicar los algoritmos de búsqueda en amplitud y búsqueda en profundidad. Para ello se desarrollará las clases necesarias para representar un grafo, y los algoritmos de camino mínimo y ordenamiento topológico.

Crear las clases Nodo, Arista y Grafo.

La clase Nodo tiene un nombre, y una lista de aristas.

La clase Arista viene identificada por un número, este identificador se genera automáticamente al crear un objeto. Como atributos tiene dos nombres, que son el origen y destino de la arista (si es un grafo dirigido), o los nodos que conecta (si es un grafo no dirigido).

La clase grafo tiene una lista de nodos y una lista de aristas. Además, debe guardar un booleano (o tipo enumerado) para saber si es un grafo dirigido o no. Debe tener un método para agregar un nuevo nodo, recibe el nombre y crea el objeto y lo añade a la lista de nodos. Otro método para crear una arista que recibe los dos nombres de los nodos involucrados.

Implementar en la clase grafo el método `camino_mas_corto`, que recibe un nodo y devuelve una lista con los pares nodo y distancia. Asumimos que el grafo es no dirigido, si lo es que devuelva un error.

Implementar en la clase grafo el método `orden_topológico`, que no recibe parámetros y devuelve la lista ordenada de nodos, o bien, la lista vacía si no existe tal orden. Asumimos que el grafo es dirigido, si no lo es devuelve un error.

#### Opcionales:

(2 EC) Implementar el método `orden_topológico` tanto para grafos dirigidos como no dirigidos.

(2 EC) Implementar el método `camino_mas_corto` tanto para grafos dirigidos como no dirigidos.