

## PRÁCTICO N°6 – Grafos

1. Dada la siguiente interfaz:

```
public interface Grafo {

    void agregarVertice (String nVertice) throws
    VerticeRepetidoException;

    void quitarVertice (String nVertice);

    void agregarArista (String nVertOrigen, String nVertDest,
    double peso);

    void quitarArista (String nVertOrigen, String nVertDest);

    double getPeso (String nVertOrigen, String nVertDestino)
    throws NoExsisteAristaException;

    ListVertices todosLosVertices();

    ListVertices adyacentes (String nVertice);

    int gradoDeEntrada (String nVertice);

    Listvertices origenes (String nVertice);

}
```

- implemente una clase Grafo que implemente los 4 primeros métodos de la interfaz, utilizando una matriz de adyacencia.
- ídem utilizando una lista de adyacencia.
- implemente el resto de los métodos de la interfaz, utilizando una matriz de adyacencia.
- ídem c), utilizando una lista de adyacencia.

2. Usando la clase definida, implemente un método realice la ordenación topológica de un grafo dirigido.

```
public List<String> ordenTopológico (Grafo g);
```

- Dada la red de Metro de Viena, se desea obtener el camino más corto entre dos paradas, entendiendo como camino más corto aquel que tenga menos paradas entre el origen y el destino. (Se asume que la distancia entre paradas es similar).  
El resultado se debe brindar imprimiendo la sucesión de paradas por las que se debe pasar para ir de la forma más corta, del origen al destino.

4. Dadas las siguientes tarifas de trenes entre las ciudades, calcule para cada ciudad cual es el camino más económico (si existe), en ir al resto de las ciudades. Tenga en cuenta que es un grafo dirigido.

