

UNIDAD TEMÁTICA 4 – GRAFOS DIRIGIDOS– Trabajo de Aplicación 5

Ejercicio 3 (30 minutos)

Se desea que el TDA Grafo cuente con funcionalidades para, dado un cierto vértice (la etiqueta) de origen y uno de destino, se encuentren y listen todos los caminos existentes del origen al destino, indicando también el costo asociado. El Equipo debe implementar entonces los siguientes métodos:

En TDAGrafoDirigido:

TCaminos todosLosCaminos(Comparable etiquetaOrigen, Comparable etiquetaDestino)

devuelve un objeto del tipo TCaminos (que contiene una colección de objetos TCamino)

En TVertice:

void todosLosCaminos(Comparable etVertDest, TCamino caminoPrevio, TCaminos losCaminos)

dado un vértice destino, una estructura del tipo TCamino “caminoPrevio” donde ir adjuntando los vértices incorporados al camino y actualizando en forma acorde el costo total, y una estructura TCaminos “losCaminos” en la que agregar un camino cada vez que se llega al destino.

Se adjunta el código fuente de implementaciones *parciales* de estos métodos. **COMPLETAR Y PROBAR!!!**

TVertice:

```
public TCaminos todosLosCaminos(Comparable etVertDest, TCamino caminoPrevio, TCaminos
todosLosCaminos) {
    this.setVisitado(true);
    for (TAdyacencia adyacencia : this.getAdyacentes()) {
        TVertice destino = adyacencia.getDestino();
        if (!destino.getVisitado()) {
            if (destino.getEtiqueta().compareTo(etVertDest) == 0) {
                TCamino copia = caminoPrevio.copiar();
                copia.agregarAdyacencia(adyacencia);
                todosLosCaminos.getCaminos().add(copia);
            } else {
                //COMPLETAR LLAMADA RECURSIVA
            }
        }
    }
    this.setVisitado(false);
    return todosLosCaminos;
}
```

TGrafoDirigido:

```
public TCaminos todosLosCaminos(Comparable etiquetaOrigen, Comparable etiquetaDestino) {
    TCaminos todosLosCaminos = new TCaminos();
    TVertice v = buscarVertice(etiquetaOrigen);
    if (v != null) {
        TCamino caminoPrevio = new TCamino(v);
        v.todosLosCaminos(etiquetaDestino, caminoPrevio, todosLosCaminos);
        return todosLosCaminos;
    }
    return null;
}
```

Ejecución:

Utilizando los archivos de entrada “aeropuertos_2.txt” y “conexiones_2.txt”:

- Ejecutar el programa para hallar todas las conexiones entre dos ciudades indicadas, junto con el costo de cada una. Responder las preguntas presentadas en pantalla.

Entregables:

Código fuente desarrollado para satisfacer las consultas especificadas, **actualizado hasta la hora 21:15** en el repositorio SVN (se calificará la última versión hasta la hora indicada). **DEBE EJECUTAR CORRECTAMENTE.**