

## UNIDAD TEMÁTICA 4 – GRAFOS DIRIGIDOS

### PRACTICOS DOMICILIARIOS INDIVIDUALES - 4

En el Trabajo de Aplicación 5, Ejercicio 3, se debían completar los métodos para encontrar **todos los caminos** entre dos vértices dados.

A efectos de consolidar el aprendizaje sobre estos algoritmos, desarrollar:

- Verificar que el código completado es correcto y que funciona correctamente
- Revisar el código a la luz de los algoritmos vistos en clase y desarrollados en pseudocódigo

#### TRANSCRIPCIÓN TRABAJO DE APLICACIÓN 5 EJERCICIO 3

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!!!**

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.

**NOTA IMPORTANTE: EL CODIGO FINAL DEL TDA GRAFO con todas las funcionalidades vistas y requeridas en todos los trabajos será necesario para los ejercicios que se han de realizar al comenzar la próxima clase. ESTOS EJERCICIOS DE APLICACIÓN SERÁN CALIFICADOS, POR LO CUAL ES IMPRESCINDIBLE QUE LOS PROGRAMAS EJECUTEN CORRECTAMENTE, SOBRE NUEVOS ARCHIVOS DE PRUEBA QUE SERÁN PUBLICADOS OPORTUNAMENTE.**