

UNIDAD TEMÁTICA 4 – GRAFOS DIRIGIDOS– Trabajo de Aplicación 3

Escenario

La aerolínea ha expandido sus operaciones y ahora tiene los siguientes vuelos, de acuerdo a la Tabla 1.

| Origen / destino | Asunción | Buenos Aires | Curitiba | Montevideo | Porto Alegre | Rio de Janeiro | San Pablo | Santos |
|------------------|----------|--------------|----------|------------|--------------|----------------|-----------|--------|
| Asunción | x | 1600 | 800 | - | 700 | - | - | - |
| Buenos Aires | 1600 | x | - | 200 | 1250 | - | - | - |
| Curitiba | - | - | x | - | - | 1500 | - | - |
| Montevideo | - | 200 | - | x | 1000 | - | 2000 | - |
| Porto Alegre | - | - | - | 1000 | x | - | 980 | 1200 |
| Rio de Janeiro | - | - | 1500 | - | - | x | 1800 | 1900 |
| San Pablo | 1200 | - | 600 | 2000 | 980 | 1800 | x | 220 |
| Santos | - | - | - | - | 1200 | - | 220 | x |

Tabla 1

Ejercicio 1

Se provee un conjunto de clases e interfaces básicas para implementar un Grafo Dirigido mediante listas de adyacencias. Este grafo se ha de instanciar a partir de un conjunto de vértices y un conjunto de aristas.

En la webasignatura se provee un archivo “código básico TDA GRAFO - para TA3.zip”.

1. En equipo, se deberá:

- Crear un paquete UT4-TA3
- Cargar en el paquete las clases y archivos contenidos en el archivo “código básico TDA GRAFO - para TA3.zip”.
- Analizar la estructura diseñada. ¿Cuáles pueden ser las razones para la utilización de las diferentes estructuras de la API de colecciones para representar el grafo, el conjunto de vértices, las listas de adyacencias? –
- Dibujar un diagrama de clases **muy abstracto, sin detalles**, de esta estructura del TDA GrafoDirigido.
- Ilustrar con un diagrama de secuencia básico cómo se ejecutaría una operación de insertar una arista en el grafo.
- Dadas estas estructuras de datos, construir un programa Java que permita
 - crear un grafo para representar las conexiones aéreas de la aerolínea “VUELE SEGURO”,
 - cargar este grafo utilizando los archivos provistos “aeropuertos.txt” y “conexiones.txt”

NOTA: el archivo “aeropuertos.txt” contiene un nombre de ciudad por línea (vértices), y el archivo “conexiones.txt” contiene, en cada línea, nombre de ciudad origen, nombre de ciudad destino y costo de la conexión, separados por comas.

- mostrar por pantalla la matriz de conexiones.

2. **FLOYD** : Agregar una funcionalidad en el TDA GrafoDirigido que permita calcular los caminos mínimos entre todos los pares de aeropuertos posibles y:
- contestar interactivamente preguntas del tipo ¿cuál es la menor distancia posible para ir desde una ciudad **x** a otra **y**?
 - contestar interactivamente preguntas del tipo “indique el itinerario de menor costo para ir de la ciudad **x** a la ciudad **y**”
 - contestar interactivamente preguntas del tipo “si la aerolínea decide instalar un nuevo centro de mantenimiento y logística para sus aviones, ¿cuál sería el aeropuerto más apropiado para ello?

Responder las preguntas presentadas en pantalla, utilizando para ello el programa desarrollado.

Entregables:

Diagramas de clases y secuencias, Código fuente desarrollado para satisfacer las consultas especificadas.

SE CALIFICARÁ EL CODIGO FUENTE SINCRONIZADO EN EL REPOSITORIO HASTA LAS 21:15

Ejercicio 2

WARSHALL: En muchas oportunidades interesa simplemente conocer si existe algún itinerario que nos permita viajar desde una cierta ciudad a otra.

Implemente un algoritmo que permita conocer la conectividad entre cualquier par de ciudades. El programa Java resultante deberá permitir:

- contestar interactivamente preguntas del tipo “indique si es posible volar desde la ciudad **x** a la ciudad **y**”

Responder las preguntas presentadas en pantalla, utilizando para ello el programa desarrollado.

Entregables:

Código fuente desarrollado para satisfacer las consultas especificadas.

SE CALIFICARÁ EL CODIGO FUENTE SINCRONIZADO EN EL REPOSITORIO HASTA LAS 21:15