

## UNIDAD TEMÁTICA 7 – GRAFOS DIRIGIDOS– Trabajo de Aplicación 4

### ESCENARIO

A efectos de mejorar la atención al cliente, la aerolínea ha decidido desarrollar una aplicación que permita conocer todos los vuelos posibles para unir un origen con un destino.

### Ejercicio 1 (20 minutos de desarrollo, 5 minutos de preguntas)

Dada la matriz que indica las conexiones existentes entre ciudades, dibuja el grafo correspondiente y su representación por lista de adyacencias. Aplicando el algoritmo de búsqueda en profundidad, desarrolla el bosque abarcador resultante de recorrer este grafo, comenzando por la ciudad “**Asunción**”. Indica claramente los diferentes **tipos de arcos** encontrados.

Origen / destino	Buenos Aires					Rio de Janeiro
	Asunción	Aires	Curitiba	Montevideo	Porto Alegre	
Asunción	X	1600	800	-	700	-
Buenos Aires	1600	x	-	200	1250	-
Curitiba	-	-	x	-	-	1500
Montevideo	-	200	-	x	1000	-
Porto Alegre	-	-	-	1000	x	-
Rio de Janeiro	-	-	1500	-	-	x

Responde las preguntas en pantalla.

## UNIDAD TEMÁTICA 7 – GRAFOS DIRIGIDOS– Trabajo de Aplicación 4

### Ejercicio 2 (30 minutos de desarrollo, 5 minutos de preguntas)

Dado el TDA Grafo implementado en JAVA, se desea poder utilizar esta estructura para realizar un recorrido sistemático en profundidad de un grafo así representado.

Se deberá entonces implementar los métodos necesarios para realizar esta recorrida, comenzando por el primer vértice representado.

Métodos a implementar:

**Collection TGrafoDirigido.bpf ()** // método del grafo. Siguiendo con el algoritmo detallado en la presentación de la cátedra, se invoca sobre cada vértice no visitado del grafo. Obtiene una **lista de vértices** recorridos (deben visitarse todos los vértices del grafo una y una sola vez).

Unas variantes de este método que serán útiles tienen las firmas

**Collection TGrafoDirigido.bpf (TVertice verticeOrigen)**

y

**Collection TGrafoDirigido.bpf (Comparable etiquetaOrigen))**

// en éstas se ejecuta el bpf a partir del vértice indicado, devolviendo una **lista de vértices** visitados en esa búsqueda en profundidad.

**void TVertice.bpf (Collection visitados)** // método de vértice, actualiza una **lista** con los vértices visitados en la búsqueda en profundidad a partir de este vértice.

### Instrucciones:

1. Descargar el proyecto del github classroom indicado “TA4-alumnos” y completar los métodos y el programa principal.
2. Al culminar el trabajo, cada sub-equipo revisa el código generado por el otro sub-equipo.
3. Responder las preguntas presentadas en pantalla
4. Integrar el código y actualizar el repositorio GIT del equipo

### Entregables:

Código fuente desarrollado para satisfacer las consultas especificadas