

# **TRABAJO**

## **PRACTICO OBLIGATORIO**

### **PROGRAMACION II**

Alumnos:

- Tomas D'Elia - LU: 1188972
- Leandro Manavella - LU: 1187848
- Sebastián Clerc - LU: 1167078

Profesor: Nicolás Pérez

Carrera: Ingeniería informática

Turno: Viernes - Mañana

Fecha de entrega: 27/06



## 1. Introducción

La gestión de vuelos entre aeropuertos es una tarea fundamental para optimizar el tiempo y los costos del transporte aéreo. En este trabajo se modela una red de vuelos como un grafo dirigido y ponderado, con el objetivo de encontrar rutas óptimas en términos de costo monetario o duración del vuelo. Utilizamos el algoritmo de Dijkstra para calcular los caminos mínimos entre aeropuertos.

Este modelo tiene aplicaciones reales en la planificación de horarios aéreos, reservas automáticas y optimización de rutas de las aerolíneas.

---

## 2. Problema

Se plantea una red de aeropuertos interconectados por vuelos. El sistema debe permitir:

- Agregar aeropuertos y vuelos con su respectivo costo y duración.
- Calcular la ruta de menor costo o menor duración entre dos puntos.

Cada aeropuerto es un nodo en el grafo, y cada vuelo es una arista dirigida con dos pesos asociados:

- **Costo:** precio del vuelo.
- **Duración:** tiempo estimado de vuelo.

El objetivo es que, dado un origen y un destino, el sistema devuelva la ruta óptima según el criterio elegido (costo o duración).

---

## 3. Algoritmo utilizado: Dijkstra

Se implementó el algoritmo de Dijkstra, que permite encontrar el camino mínimo desde un nodo origen a un nodo destino en un grafo con pesos no negativos.

#### *Pasos principales:*

1. Inicializar distancias a infinito, excepto el nodo origen (cero).
2. Usar una cola de prioridad para seleccionar el nodo con distancia más baja.
3. Actualizar distancias de los vecinos si se encuentra un camino más corto.
4. Repetir hasta alcanzar el nodo destino o vaciar la cola.

Este algoritmo garantiza rutas óptimas en términos de costo o duración, dependiendo del peso seleccionado.

---

#### 4. Implementación del sistema

Se desarrollaron las siguientes clases:

- **Aeropuerto:** contiene información del código, ciudad y país.
- **Vuelo:** contiene aeropuerto origen y destino, costo y duración.
- **GrafoVuelos:** estructura de grafo con métodos para agregar vuelos y calcular caminos mínimos.
- **TipoPeso (enum):** define los criterios de optimización (COSTO o DURACION).

El método `dijkstra(String origen, String destino, TipoPeso criterio)` permite obtener la ruta óptima según el criterio elegido.

Se realizaron pruebas con aeropuertos "BUE", "SCL" y "LIM" con distintas combinaciones de vuelos:

- Test de **mínimo costo:** elige rutas más económicas, aunque más largas.
  - Test de **mínima duración:** elige rutas directas, aunque sean más costosas.
- 

#### 5. Comparación con lo visto en clase

La solución es coherente con el algoritmo de Dijkstra estudiado en clase. En este caso se trabaja con grafos donde cada arista tiene **dos pesos posibles**, y el usuario elige cuál usar.

## 6. Conclusión

Este trabajo aplica conceptos de grafos, caminos mínimos, siempre bajo el paradigma de TDA. La implementación modular (modelo, interfaces y tests) facilita la extensión y mantenimiento del código.

Ejemplo graficado del código:

