

# Algoritmos y Programación II

## Informe TP n° 2 - Algueiza

Nro. de Grupo: **G24**

Datos personales de los Integrantes:

- 1° Integrante:
  - Apellidos: **Tello Meléndez.**
  - Nombres: **Edgar Tello.**
  - Padrón: **109974.**
  - Email: **etellom@fi.uba.ar**
- 2° Integrante:
  - Apellidos: **Ortiz Briceño.**
  - Nombres: **William Francisco.**
  - Padrón: **109546.**
  - Email: **wortiz@fi.uba.ar**

Ayudante Asignado: **Federico Brasburg.**

## Enunciado:

El aeropuerto de Algueiza opera la entrada y salida de aviones; los operarios de dicho aeropuerto nos han pedido que implementemos un sistema de consulta de vuelos que les permita obtener la información de los vuelos solicitados según distintos criterios, como; prioridad, fecha u código único de vuelo, y ver la información detallada del vuelo en cuestión.

## Resolución:

Para empezar a diseñar el trabajo práctico se realizaron varios pasos previos.

En primera instancia, se realizó un análisis de lo solicitado por los operarios, prestando especial atención a los tiempos de ejecución demandados, para ello se buscó relacionar directamente cada comando o acción del sistema con los TDAs implementados en la cátedra, de esta manera se concluyó que serían necesarias dos estructuras de datos (Hash y ABB) para almacenar la información de formas diferentes, dos más auxiliares (Pila, Heap) para poder ordenar dicha información según otros criterios, y adicionalmente, definimos un TDA para el manejo del sistema, y así lograr nuestro cometido con su respectiva complejidad.

Entonces, para guardar la información nos dimos cuenta que era más conveniente usar, un Diccionario Ordenado (ABB), ya que este nos permitiría recorrer la información en un rango especificado y además en tiempo logarítmico; y un Diccionario (Hash), ya que este nos facilitaría obtener los detalles de un vuelo en tiempo constante; de esta forma se obtuvo que en:

**Agregar\_archivo:** se guarda “V” veces (cantidad de vuelos en el archivo) en el sistema que ya cuenta con “n” vuelos almacenados, entonces en el ABB ordenado por fechas  $\rightarrow O(V \log n)$ . Y en el Hash  $\rightarrow O(V)$ , por lo tanto se logra que el tiempo de ejecución sea  $O(V \log n)$ .

**Ver\_tablero:** se filtran los vuelos, siendo “v” la cantidad de vuelos del sistema, en un rango especificado desde el ABB  $\rightarrow O(\log v)$ , además si se pide en modo descendente se usa una Pila auxiliar para invertir el orden en base a la información filtrada resultante, por lo que se ejecutaría también en  $O(\log v)$ ; y en el peor de los casos donde se tenga que mostrar todos los “v” vuelos esto resultaría ser  $O(v)$ .

**Info\_vuelo:** se obtiene en tiempo constante el detalle del vuelo del Hash  $\rightarrow O(1)$ , sabiendo que en este TDA sus primitivas son en tiempo constante.

**Siguiente\_vuelo:** se recorre el ABB desde una fecha hasta que coincida el origen y destino solicitado, el ABB se itera en un rango de fechas por lo que la complejidad terminaría siendo  $O(\log F_{\text{conexión}})$ , donde “ $F_{\text{conexión}}$ ” es la cantidad de fechas diferentes en las que se puede hacer dicho viaje.

**Borrar:** se borra “ $K$ ” veces (cantidad pedida de vuelos a borrar) en un rango especificado, para esto primero debo obtener los vuelos dentro del rango desde el ABB  $\rightarrow O(\log n)$  y luego tengo que borrar del ABB y del Hash para lo cual la complejidad termina siendo  $O(K \log n)$

Además, de forma auxiliar se utiliza un método de ordenamiento HeapSort únicamente para ordenar los vuelos según su prioridad y, de esta forma, tenerlos a disposición momentáneamente para lo requerido. Por ello, se tiene que:

**Prioridad\_vuelos:** se obtienen los vuelos desde el Hash  $\rightarrow O(n)$  y se aplica HeapSort a la lista resultante y como solo piden mostrar “ $K$ ” vuelos  $\rightarrow O(K \log n)$ , por lo cual la complejidad terminaría siendo  $O(n + K \log n)$ .

Finalmente, en segunda instancia, se optó por analizar la modularización aplicada en el TP1 y sus TDAs adicionales, para así obtener una idea concreta para basarnos. De esta manera, se definió un TDA adicional el cual representaría una base de datos o sistema de vuelos, siendo su comportamiento las funcionalidades solicitadas por los operarios, ya sea almacenar, borrar y consultar información de diversas formas.