

TRABAJO PRACTICO N°2

Integrantes:

- **ARAUJO IGNACIO - 110041**
- **TAYLOR ALAN - 110034**

Correctora:

- **LARA SCAZZOLA**

ANALISIS Y DISEÑO DE LA ESTRUCTURA TP2 - ALGUEIZA

1 INTRODUCCIÓN

Este informe detallara las descripciones de los tipos de datos y estructuras utilizadas para la resolución de la problemática planteada para la operación de vuelos en el aeropuerto de Algueiza.

2 PROBLEMÁTICA PLANTEADA

2.1 CARACTERÍSTICAS GENERALES

Al encontrarnos con el problema planteado, este en cuestión nos solicitaba que implementemos un sistema para que el aeropuerto pueda almacenar datos de vuelos, obtener información de los mismos, borrarlos del sistema, y todo lo que sería el manejo de la información de vuelos, todo, en un tiempo determinado.

3 RESOLUCION

Para lograr esta solución, nos decidimos por implementar un TDA Aeropuerto, que como bien explicamos anteriormente que se encarga de cumplir las funciones solicitadas. Además utilizamos una estructura auxiliar para almacenar la información de cada vuelo leído en los archivos de entrada.

El TDA antes mencionado utiliza otros TDAs, para el guardado de la información, en el sistema. Decidimos utilizar un TDA Diccionario, implementado mediante un Hash, para acceder fácilmente a los vuelos por su código particular. A su vez elegimos el uso de un TDA DiccionarioOrdenado, implementado mediante un ABB, para poder separar los vuelos según su fecha de despegue. Y por último, un Diccionario de Dicciones que almacena los vuelos según su aeropuerto origen y destino, y dentro de cada uno las fechas en las que ocurrirá cada vuelo.

3.1 AGREGAR ARCHIVO

En esta función guardamos los datos en las correspondientes estructuras mencionadas en el punto anterior. E implementamos ésta, cumpliendo con el tiempo de ejecución pedido, ya que por cada vuelo del archivo, almacenamos en los diccionarios mencionados que permiten hacer esos ingresos en tiempos constantes y, en su peor caso, logarítmicos.

3.2 VER TABLERO DE VUELOS

Para esta funcionalidad simplemente accedemos al diccionario ordenado por las fechas de los vuelos para mostrar la información por el tablero. La justificación del uso de este diccionario es que nos permite cumplir con la complejidad ya que permite encontrar los vuelos buscados en tiempo logarítmico debido a que funciona mediante División y Conquista.

En el caso en el que se pida que sea mostrada de forma descendente, vimos necesario el uso del TDA Pila, ya que sus operaciones de tiempo constante permiten que se cumpla la complejidad temporal requerida y se muestren los vuelos en forma correcta.

3.3 OBTENER INFORMACION DE UN VUELO

Con el propósito de esta función simplemente accedemos a la información almacenada en el diccionario de códigos de vuelo, debido a que buscábamos que este acceso sea en tiempo constante, y el TDA Hash permite eso.

3.4 VER K VUELOS PRIORITARIOS

En este apartado se nos solicitaba mostrar los “k” vuelos ordenados de mayor a menor prioridad, que estén en el sistema. Para obtenerlos en el tiempo estipulado, vimos necesario el uso del TDA Cola Prioridad, implementado con un Heap. Previo a la creación de la cola, creamos un arreglo con los vuelos para luego crear un Heap a partir de este, lo cual termina siendo tiempo lineal. A esto le sumamos el desencolamiento “k” veces del Heap, pudiendo lograr “k” operaciones en tiempo logarítmico. Resumidamente el algoritmo terminaría resultando $O(n + k \log(n))$.

3.5 SIGUIENTE VUELO

Esta función fue implementada siguiendo lo pedido, utilizando un diccionario de diccionarios. Primeramente tenemos un diccionario implementado con TDA Hash al cual se accede en tiempo constante por la conexión (aeropuerto origen-destino) de un vuelo, y dentro de cada uno, un diccionario ordenado implementado por TDA ABB con las fechas en las que pueden ocurrir los distintos vuelos que cumplen esa conexión.

Elegimos esta manera para no tener que recorrer todas las fechas de todos los vuelos, sino que accedemos directamente a la conexión que nos interesa y terminamos recorriendo muchas menos fechas de esta

manera, logrando encontrar el siguiente vuelo a la fecha pedida, en el tiempo solicitado.

3.6 BORRAR VUELOS

Para esta última función, simplemente recorreremos el diccionario ordenado por fechas, que mencionamos previamente, en el rango de fechas ingresado, almacenando estos vuelos en un arreglo. Luego recorriendo ese arreglo, vamos borrando la información de estos vuelos de cada uno de los diccionarios que están cargados en el sistema. En resumen, terminamos cumpliendo con la complejidad porque simplemente recorreremos los vuelos del rango de fechas ingresado, y luego hacemos operaciones de tiempo constantes y logarítmicas para borrarlos del sistema.