

# ANÁLISIS DEL RETO

Simón Peña Alarcón - 202512907- [s.penaa2@uniandes.edu.co](mailto:s.penaa2@uniandes.edu.co)

Juan Sebastian Chacón Ochoa-202513196- [j.chacono@uniandes.edu.co](mailto:j.chacono@uniandes.edu.co)

Manuel Santiago Figueroa-202511697- [m.figueroag@uniandes.edu.co](mailto:m.figueroag@uniandes.edu.co)

## Requerimiento <<1>>

### Descripción

En este requerimiento filtré los vuelos de una aerolínea según un rango de retraso en minutos. Para eso recorrí toda la lista de vuelos, calculé el retraso comparando la hora real con la hora programada, hice la corrección del día si era necesario, y si el vuelo cumplía el rango lo guardaba en un árbol rojo-negro. La clave la armé con el delay, la fecha y la hora, para que los resultados quedaran ordenados. Al final saqué los primeros 5 y los últimos 5 vuelos.

<b>Entrada</b>	Catálogo de vuelos Código de aerolínea Rango de retraso (mínimo y máximo)
<b>Salidas</b>	Tiempo de ejecución Número de vuelos filtrados Primeros 5 vuelos Últimos 5 vuelos
<b>Implementado (Sí/No)</b>	Si se implementó por Juan Sebastian Chacón Ochoa

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Recorrer todos los vuelos	$O(n)$
Calcular retraso por vuelo	$O(1)$
Insertar en el árbol RBT Obtener los primeros 5 Obtener los últimos 5	$O(\log n)$ $O(n)$ $O(n)$
<b>TOTAL</b>	$O(n \log n)$ (por los inserts en el RBT)

## Análisis

Este requerimiento me permitió manejar bien el cálculo de retrasos y entender cómo ordenar datos usando una clave compuesta. También comprobé que usar un árbol rojo-negro hace que los vuelos queden ordenados automáticamente, lo cual ayuda para sacar los primeros y últimos 5 sin ordenar manualmente. Probé casos con rangos amplios y también casos donde no se filtraba ningún vuelo, y la función respondió bien.

## Requerimiento <<2>>

### Descripción

En este requerimiento filtré vuelos según el retraso en llegada a un aeropuerto específico. Hice un recorrido por toda la lista, calculé el retraso en minutos comparando hora real y programada, arreglé casos donde el vuelo cruzaba medianoche y luego inserté cada vuelo válido en un RBT. Después obtuve los primeros 5 y los últimos 5 vuelos de ese árbol.

<b>Entrada</b>	Catálogo Código del aeropuerto destino Rango de retraso (min, max)
<b>Salidas</b>	Tiempo de ejecución Número de vuelos filtrados Primeros 5 Últimos 5
<b>Implementado (Sí/No)</b>	Si. Implementado por Simón Peña Alarcón

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

<b>Pasos</b>	<b>Complejidad</b>
Recorrer los vuelos	$O(n)$
Calcular retraso	$O(1)$
Insertar en el RBT	$O(\log n)$
Obtener primeros 5	$O(n)$
Obtener últimos 5	$O(n)$
<b>TOTAL</b>	<b><math>O(n \log n)</math></b>

## Análisis

Este requerimiento fue parecido al primero, pero analizando los retrasos de llegada en vez de salida. Me ayudó a entender mejor cómo hacer filtros por destino y cómo construir claves grandes para ordenar de

forma precisa. Las pruebas mostraron que funciona incluso cuando hay muchos vuelos con retraso negativo o muy alto.

## Requerimiento <<3>>

### Descripción

Este requerimiento busca vuelos de una aerolínea hacia cierto aeropuerto y dentro de un rango de distancias. Los vuelos filtrados se guardan en un RBT donde la clave es la distancia. Además, dentro de cada distancia, los vuelos se ordenan internamente por fecha y hora. Al final se sacan los primeros 5 y los últimos 5 vuelos según ese orden.

<b>Entrada</b>	Catálogo Aerolínea Código del aeropuerto Rango de distancias
<b>Salidas</b>	Tiempo total Total de vuelos filtrados Primeros 5 Últimos 5
<b>Implementado (Sí/No)</b>	Si se implementó por Manuel Santiago Figueroa

### Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Recorrer los vuelos	$O(n)$
Insertar en RBT por distancia Ordenar vuelos dentro de la misma distancia Obtener primeros 5 Obtener últimos 5	$O(\log n)$ $O(k \log k)$ $O(n)$ $O(n)$
<b>TOTAL</b>	$O(n \log n)$

### Análisis

Este requerimiento fue útil para trabajar con agrupación por claves (distancia) y después ordenar dentro de cada grupo. Aunque es más largo que los anteriores, el uso del árbol me facilitó manejar las distancias ordenadas automáticamente. Las pruebas mostraron que sí devuelve vuelos en el orden correcto y que manejar las fechas como enteros ayuda bastante.

## Requerimiento <<4>>

### Descripción

Aquí debía filtrar vuelos por fecha y hora y agruparlos por aerolínea. Para cada aerolínea calculé cuántos vuelos tenía, la distancia promedio, la duración promedio y el vuelo con menor distancia. Después guardé cada aerolínea en un RBT usando como clave el número de vuelos, para quedarme con las que más aparecían.

<b>Entrada</b>	Catálogo Rango de fechas Rango de horas Número de aerolíneas a mostrar
<b>Salidas</b>	Tiempo Total de vuelos filtrados Lista de aerolíneas seleccionadas
<b>Implementado (Sí/No)</b>	Si se implementó

### Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Recorrer vuelos	$O(n)$
Agrupar por aerolínea	$O(n)$
Calcular promedios	$O(n)$
Insertar aerolíneas en RBT	$O(\log n)$
Obtener n resultados	$O(n)$
<b>TOTAL</b>	<b><math>O(n \log n)</math></b>

### Análisis

Este requerimiento fue bueno para practicar cómo agrupar información y calcular estadísticas básicas. Me permitió manejar estructuras donde cada aerolínea guarda su propia lista de vuelos. El uso del RBT ayudó a ordenar aerolíneas según cantidad de vuelos, lo cual simplificó obtener las que más actividad tenían.

## Requerimiento <<5>>

### Descripción

En este requerimiento filtré vuelos hacia un destino dentro de un rango de fechas. Luego agrupé estos vuelos por aerolínea y calculé para cada una: el retraso promedio, la distancia promedio, la duración promedio y el vuelo más largo. Después las guardé en un RBT con una llave basada en el retraso promedio y saqué las N aerolíneas más puntuales.

<b>Entrada</b>	Catálogo Código destino Rango de fechas N (cantidad de aerolíneas a mostrar)
<b>Salidas</b>	Tiempo Total de aerolíneas encontradas Lista con las más puntuales
<b>Implementado (Sí/No)</b>	Si se implementó

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Recorrer vuelos	$O(n)$
Agrupar info por aerolínea	$O(n)$
Calcular promedios	$O(n)$
Insertar aerolíneas en RBT	$O(\log n)$
Tomar primeros N	$O(n)$
<b>TOTAL</b>	<b><math>O(n \log n)</math></b>

## Análisis

Este requerimiento fue útil para practicar cálculos estadísticos simples y selección de elementos usando un árbol. Además, identificar el vuelo más largo dentro de cada aerolínea me ayudó a trabajar mejor con listas internas. Las pruebas mostraron que las aerolíneas sí quedan ordenadas por puntualidad y que los cálculos funcionan bien.

## Requerimiento <<6>>

### Descripción

Este requerimiento filtra vuelos por fecha y distancia y agrupa los resultados por aerolínea. Para cada aerolínea calcula el promedio de retraso, la desviación estándar y el vuelo más cercano a ese promedio. Después las inserta en un heap mínimo para sacar las m aerolíneas más consistentes (las que tienen menor desviación).

<b>Entrada</b>	Catálogo Rango de fechas Rango de distancias m (cantidad a mostrar)
<b>Salidas</b>	Tiempo Número de aerolíneas devueltas

	Lista final con la información de cada una
<b>Implementado (Sí/No)</b>	Si se implementó

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

<b>Pasos</b>	<b>Complejidad</b>
Recorrer vuelos	$O(n)$
Agrupar datos por aerolínea	$O(n)$
Calcular promedios y desviación	$O(n)$
Insertar en heap	$O(\log n)$
Sacar resultados del heap	
<b>TOTAL</b>	<b><math>O(n \log n)</math></b>

## Análisis

Este requerimiento fue el más estadístico, ya que incluyó promedio, varianza y desviación estándar. Me ayudó mucho a entender cómo seleccionar elementos usando un heap mínima, y cómo organizar información por aerolínea de forma eficiente. Con las pruebas pude ver que sí se devuelven las aerolíneas más estables en términos de retrasos.