



Documento de Análisis y Diseño

Nivel 16: [CupiFlights]

[Felipe Otálora]

[Sebastián Florez]

Estructuras de Datos

Universidad de Los Andes

[III]



Documento de Análisis y Diseño

[CupiFlights]

Versión del documento	Fecha de modificación	Responsables
v1	20 Abril - 2014	Felipe Otálora Sebastián Florez



Contenido

1. Requerimientos Funcionales	3
2. Análisis del modelo del mundo.....	6
3. Diseño de Interfaces	7
3.1. Interfaces del mundo de la aplicación	7
3.2. Interfaces de las estructuras de datos.....	7
4. Diseño de estructuras de datos.....	8
5. Justificación de decisiones de diseño de las estructuras de datos	10
6. Diseño final de la aplicación	11
7. Justificación de decisiones de diseño del mundo.....	12
8. Análisis de complejidad de operaciones del mundo	13



1. *Requerimientos Funcionales*

A continuación se describen los requerimientos funcionales de la aplicación. La definición de estos es independiente de la tecnología que será usada para implementarlos.

ID	R1
Nombre	Cargar la información de por lo menos 50 aeropuertos y los vuelos entre estos
Resumen	La aplicación debe incluir la información de por lo menos 50 aeropuertos de las ciudades mas importantes del mundo y los vuelos entre estos para poder hacer todo tipo de consultas. La información debe estar en una pantalla con un rango de tiempo de 8 a 9 días.
Entradas	NA
Resultados	La información de los vuelos extraída desde el API de FlightStats ha sido almacenada por medio de la persistencia y esta disponible para consulta por el usuario.

ID	R2
Nombre	Agregar un aeropuerto al sistema
Resumen	Se debe poder agregar un nuevo aeropuerto al sistema y se deben añadir todos los existentes entre este y los demás aeropuertos.
Entradas	El aeropuerto y los vuelos que tiene.
Resultados	Se ha agregado el aeropuerto al sistema junto con sus vuelos.

ID	R3
Nombre	Eliminar un aeropuerto del sistema
Resumen	El administrador debe poder eliminar el aeropuerto junto con todos los vuelos pertinentes a este.
Entradas	El aeropuerto que se quiere eliminar.
Resultados	El aeropuerto se ha eliminado junto con los vuelos pertinentes a este.

ID	R4
-----------	----



Nombre	Consultar/actualizar la calificación de un aeropuerto
Resumen	El administrador del sistema debe poder consultar y actualizar la información de un aeropuerto.
Entradas	La nueva calificación del aeropuerto. El aeropuerto que se quiere consultar
Resultados	Se ha actualizado/consultado la calificación de ese aeropuerto.
ID	R5
Nombre	Consultar las fechas para los cuales el sistema CupiFlights tiene información disponible.
Resumen	El usuario debe poder consultar las fechas para las cuales el sistema CupiFlights tiene información disponible.
Entradas	La fecha por la cual se quiere buscar
Resultados	La información disponible para la fecha dada
ID	R6
Nombre	Consultar los aeropuertos para los cuales el sistema CupiFlights tiene información disponible.
Resumen	El usuario debe poder consultar la información disponible en el sistema CupiFlights de un aeropuerto específico.
Entradas	El aeropuerto que se quiere consultar.
Resultados	La información del aeropuerto buscado se ha presentado.
ID	R7
Nombre	Consultar los vuelos registrados en el sistema CupiFlights de un aeropuerto en una fecha específica. La consulta debe precisar el tipo de vuelo.
Resumen	El usuario debe poder buscar información de un aeropuerto y sus vuelos dada una fecha específica.
Entradas	El aeropuerto que se quiere consultar. La fecha por la cual se quiere buscar vuelos del aeropuerto
Resultados	Se ha mostrado la información de el aeropuerto y sus vuelos por la fecha dada.
ID	R8
Nombre	Ver el total de vuelos a tiempo, tardíos y cancelados por aeropuerto en un periodo de tiempo.

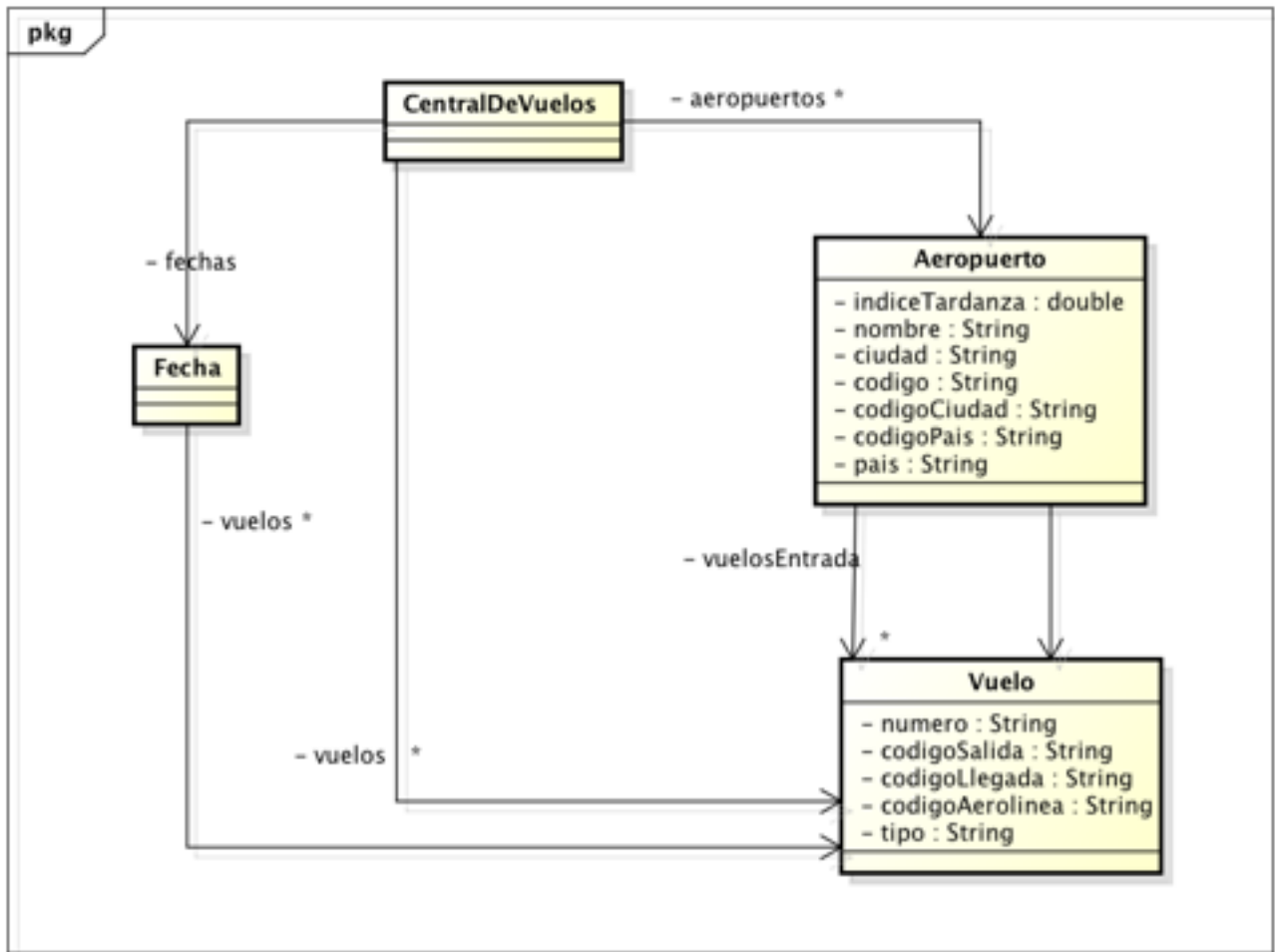


Resumen	El usuario debe poder buscar información de un aeropuerto y la cantidad de vuelos tardíos, cancelados y a tiempo dada un rango de tiempo.
Entradas	El aeropuerto que se quiere consultar. El rango de tiempo de la consulta.
Resultados	Se ha mostrado la información de el aeropuerto y las estadísticas de los vuelos.
ID	R9
Nombre	Buscar vuelos cuya calificación se encuentre dentro de un rango
Resumen	El usuario debe poder buscar información de los vuelos cuya calificación este entre el rango especificado.
Entradas	La calificación por la que se quiere buscar
Resultados	Se ha mostrado la información de los vuelos basados en la calificación buscada.
ID	R10
Nombre	Consultar las aerolíneas con el mayor y menor número de vuelos tardíos dado un rango de tiempo.
Resumen	El usuario debe poder buscar información de las aerolíneas basados en el número de vuelos tardíos en un rango de tiempo.
Entradas	El rango de tiempo por el que se quiere buscar.
Resultados	Se ha mostrado la información de la aerolínea y sus vuelos por el rango dado.
ID	R11
Nombre	Mostrar los aeropuertos según su índice de tardanza.
Resumen	La aplicación debe permitir mostrar los aeropuertos dado su índice de tardanza.
Entradas	NA
Resultados	Se ha mostrado la información de el aeropuerto dado su índice de tardanza.



2. Análisis del modelo del mundo

Diagrama de clases (UML) de la abstracción del problema que se pretende solucionar. Solo se tienen en cuenta entidades, características y relaciones relevantes para el mismo. Aún no se toman decisiones de diseño.





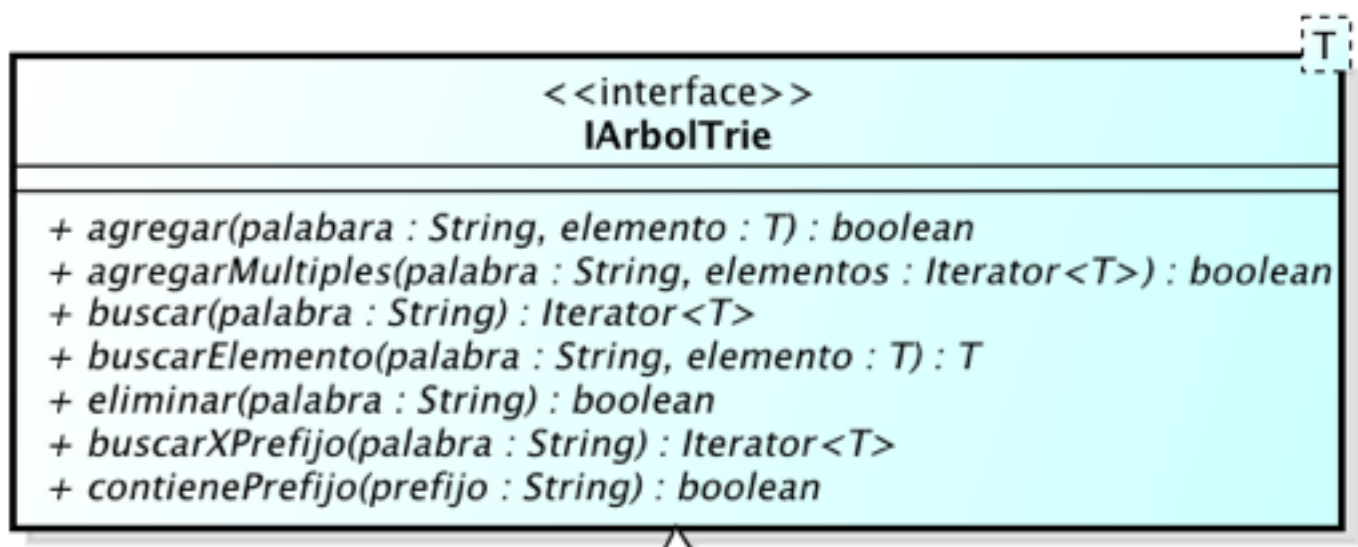
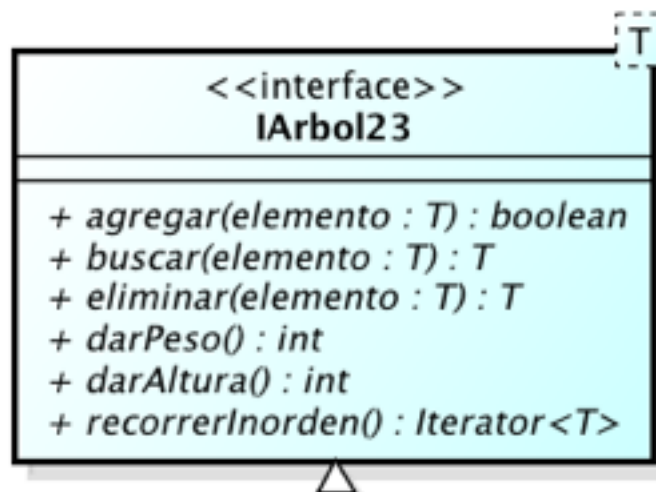
3. Diseño de Interfaces

3.1. Interfaces del mundo de la aplicación

A partir de los requerimientos funcionales, defina las operaciones (a través de métodos) que debe proveer el mundo del problema. Incluya parámetros, tipos de retorno y documentación completa de cada operación (incluyendo precondiciones).

3.2. Interfaces de las estructuras de datos

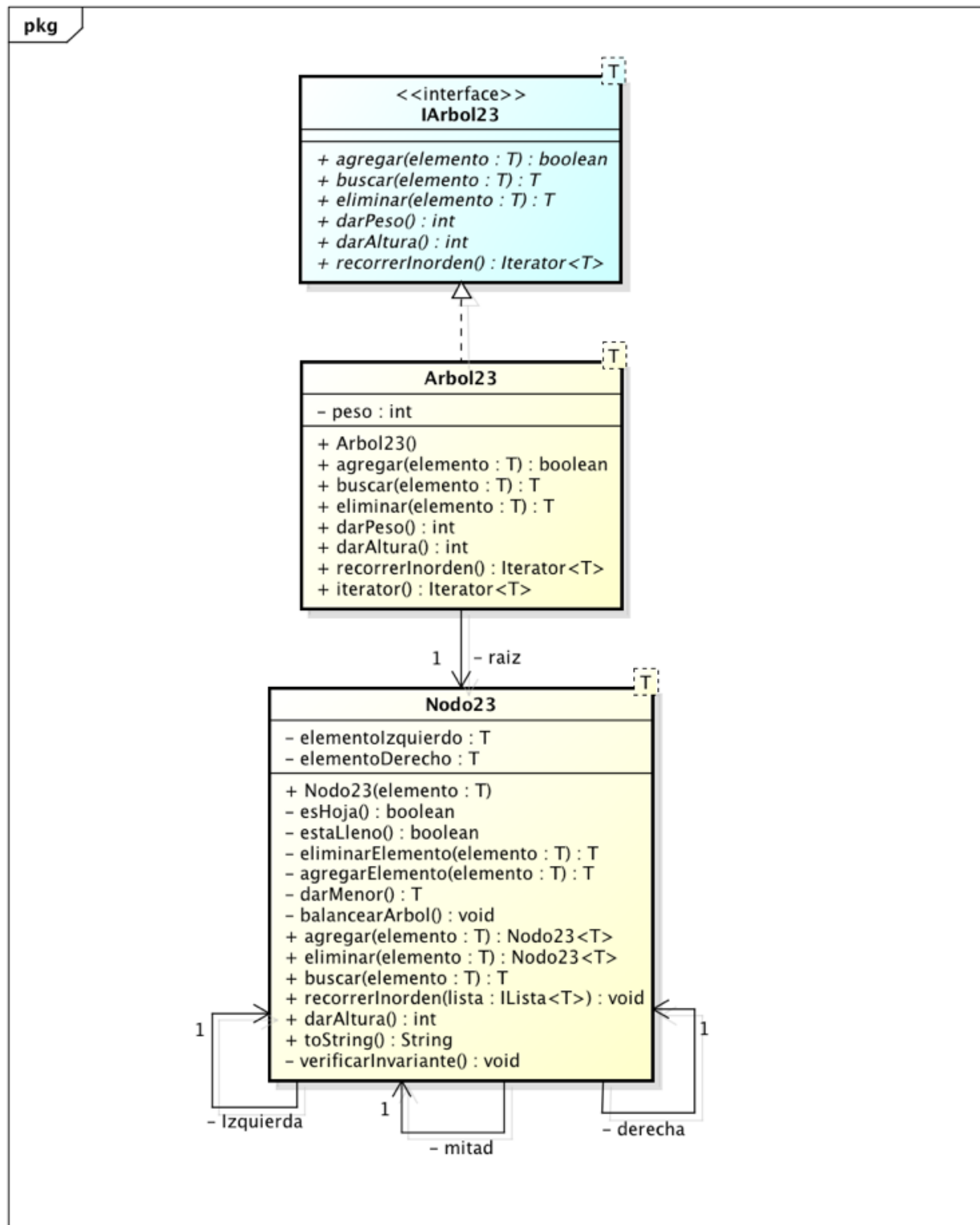
Seleccione cuáles son las estructuras de datos genéricas que debe utilizar y las operaciones que estas deben proveer para minimizar la complejidad temporal de las operaciones definidas en el numeral 3.1.





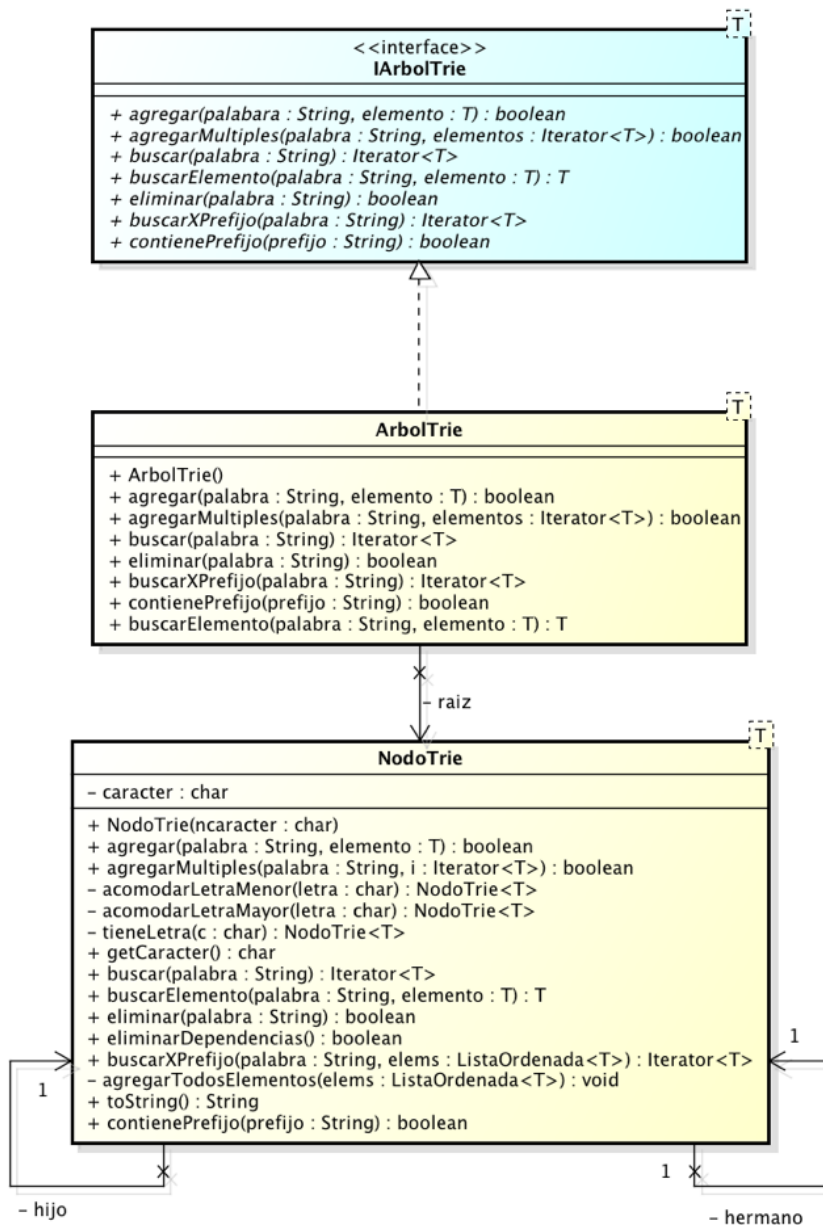
4. Diseño de estructuras de datos

Describe el detalle de las Estructuras de Datos que va a utilizar en su solución, a través de un diagrama UML de las mismas. No olvide incluir invariantes si las requiere.





pkg





5. *Justificación de decisiones de diseño de las estructuras de datos*

Describa en detalle los aspectos que tuvo en cuenta para seleccionar y diseñar las estructuras de datos descritas en el punto anterior.

- En el caso de los árboles Tries, estas estructuras permiten almacenar información de manera ordenada bajo un mismo criterio, en este caso, un prefijo, que permite centralizar todos los datos de un solo tipo, por ejemplo, dadas las fechas de los vuelos, cada vuelo se puede agrupar bajo ese parámetro.
- Los árboles 2-3 agrupan los datos de un modo similar al árbol-ordenado-AVL pero adicionalmente, permiten reducir la complejidad de las tareas que toda estructura requiere normalmente, como lo son el eliminar y agregar. Su funcionamiento es muy similar al de cualquier árbol.



7. *Justificación de decisiones de diseño del mundo*

Describa en detalle los aspectos que guiaron el diseño final de su aplicación.

Decidimos implementar las dos estructuras de datos en este nivel, tanto los árboles 2-3 como los tries, para poder aprovechar la capacidad de ordenamiento de cada uno. Por ejemplo, debido a que los árboles 2-3 reducen la complejidad algorítmica necesaria para las búsquedas, en el caso de todos los vuelos que tiene la central, es mas fácil poder encontrar un elemento comparado con el Trie. Por el contrario, el Trie permite agrupar elementos similares bajo un criterio (prefijo) que es extremadamente útil para agrupar información semejante. Como un ejemplo para el caso de los tries, los vuelos de cada aerolínea están almacenados en este tipo de estructura y solo es necesario recorrer el nombre de la aerolínea para obtener los vuelos.



8. *Análisis de complejidad de operaciones del mundo*

Basado en los diseños de las estructuras de datos y en el mundo de la aplicación, defina la complejidad de cada una de las operaciones. Utilice la notación $O(f(n))$. Justifique por qué considera que esta es la mínima posible para el problema que está resolviendo.

Requerimiento	Complejidad	Justificación
R1	$\log_3(n)$	El cargar la información desde el API no es complejo, sin embargo, el agregar esa información al árbol 2-3 si lo es, y por eso tiene la misma complejidad que le agregar.
R2	$\log_3(n)$	Después de pedir el aeropuerto desde el API, el agregar el aeropuerto toma la misma complejidad que el agregar del árbol 2-3.
R3	$\log_3(n)$	Eliminar el aeropuerto tiene la misma complejidad que el eliminar del árbol.
R4	$O(k)$	Solo es necesario hacer la asignación correcta cuando se tenga el aeropuerto para cambiar su calificación.
R5	$O(n)$	La complejidad de buscar las fechas que el sistema tiene almacenadas tiene la misma complejidad que le buscar del Trie.
R6	$\log_3(n)$	La complejidad de ver la información de un aeropuerto depende del buscar del árbol 2-3.
R7	$O(n)$	Buscar los aeropuertos que tiene el nodo dependen de la complejidad de buscar del árbol trie.
R8	$O(n)$	Es necesario recorrer todos los vuelos para conocer el estado de cada uno.
R9	$O(n)$	Es necesario preguntarle a cada aeropuerto su calificación para conocer si esta dentro del rango especificado.
R10	$O(n)$	Es necesario recorrer todos los aeropuertos para conocer su índice de tardanza y mostrarlos.
R11	$O(n)$	Es necesario recorrer al menos una vez todos los aeropuertos para poder visualizar su información dado su índice de tardanza.