

# ANÁLISIS DEL RETO

Student-1, **\*\*Re4\*\***, Pablo Ramírez, [p.ramirez2@uniandes.edu.co](mailto:p.ramirez2@uniandes.edu.co), C.E. 202321722

Student-2, **\*\*Re3\*\***, Manuel Ricardo Torres, [mr.torres@uniandes.edu.co](mailto:mr.torres@uniandes.edu.co), C.E. 202321428

Student-3, **\*\*Re5\*\***, Diego Alberto Ojeda Silva, [d.ojedas@uniandes.edu.co](mailto:d.ojedas@uniandes.edu.co), C.E. 202321126

## Requerimiento <<1>>

### Descripción

Este requerimiento se encarga de retornar el camino encontrado entre dos aeropuertos. Sin embargo, si no se encuentra alguno de los dos dentro de un rango de 30 KM, retorna únicamente el origen y destino (aeropuertos más cercanos a las coordenadas digitadas por parámetro) y la distancia correspondiente a dichos puntos desde los aeropuertos.

<b>Entrada</b>	Estructura de datos del modelo, punto de origen y punto de destino (latitudes y longitudes)
<b>Salidas</b>	La distancia total del camino, el número de aeropuertos que visita, el camino encontrado indicando las especificaciones de cada aeropuerto.
<b>Implementado (Sí/No)</b>	Implementado por Manuel Ricardo Torres, Diego Ojeda y Pablo Ramírez

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

<b>Pasos</b>	<b>Complejidad</b>
Recorrido DFS	$O(V+E)$
<b>TOTAL</b>	<b><math>O(V+E)</math></b>

## Pruebas Realizadas

Se realizó la prueba de dicho requerimiento usando como datos de entrada el origen con coordenadas (Latitud 4.601992771389502, Longitud -74.06610470441926) y el destino con coordenadas (Latitud 10.507688799813222, Longitud -75.4706488665794).

Procesadores	Apple Silicon M1 Pro
Memoria RAM	16 GB
Sistema Operativo	MacOS Sonoma

Tablas de datos

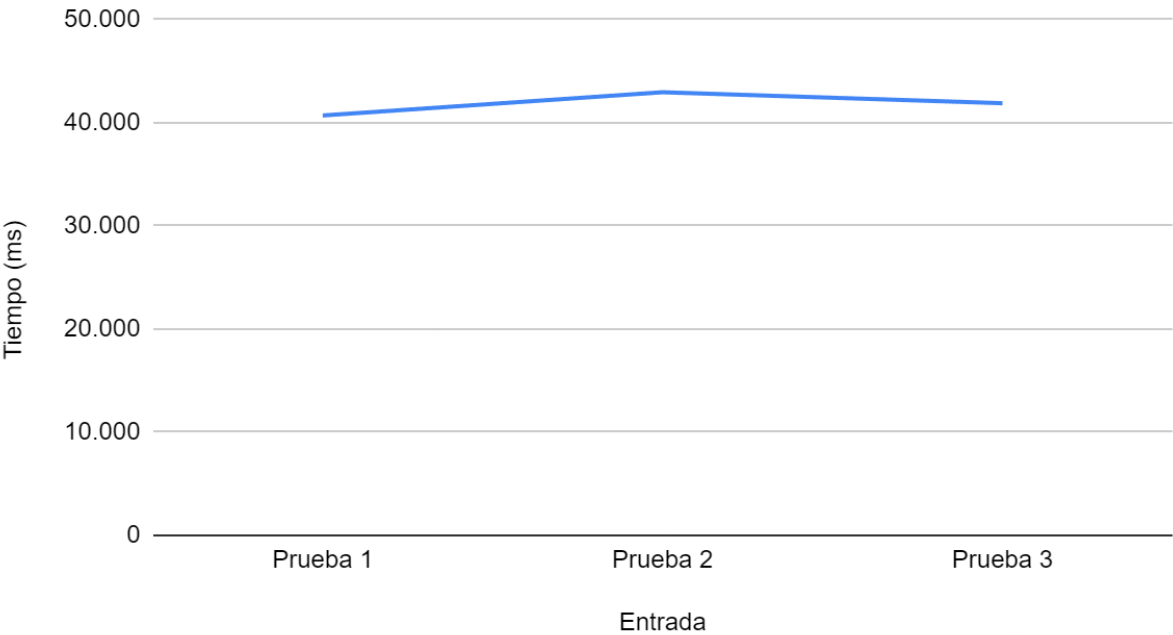
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	40.705
Prueba 2	42.956
Prueba 3	41.892

Graficas

Las gráficas con la representación de las pruebas realizadas.

Tiempo (ms) frente a Entrada



## Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

## Requerimiento <<2>>

### Descripción

Este requerimiento se encarga de retornar el camino encontrado entre dos aeropuertos. Sin embargo, si no se encuentra alguno de los dos dentro de un rango de 30 KM, retorna únicamente el origen y destino (aeropuertos más cercanos a las coordenadas digitadas por parámetro) y la distancia correspondiente a dichos puntos desde los aeropuertos.

<b>Entrada</b>	Estructura de datos del modelo, punto de origen y punto de destino (latitudes y longitudes)
<b>Salidas</b>	La distancia total del camino, el número de aeropuertos que visita, el camino encontrado indicando las especificaciones de cada aeropuerto.
<b>Implementado (Sí/No)</b>	Implementado por Manuel Ricardo Torres, Diego Ojeda y Pablo Ramírez

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

<b>Pasos</b>	<b>Complejidad</b>
Recorrido BFS	$O(V+E)$
<b>TOTAL</b>	<b><math>O(V+E)</math></b>

## Pruebas Realizadas

Se realizó la prueba de dicho requerimiento usando como datos de entrada el origen con coordenadas (Latitud 4.601992771389502, Longitud -74.06610470441926) y el destino con coordenadas (Latitud 10.507688799813222, Longitud -75.4706488665794).

<b>Procesadores</b>	<b>Apple Silicon M1 Pro</b>
<b>Memoria RAM</b>	16 GB
<b>Sistema Operativo</b>	MacOS Sonoma

## Tablas de datos

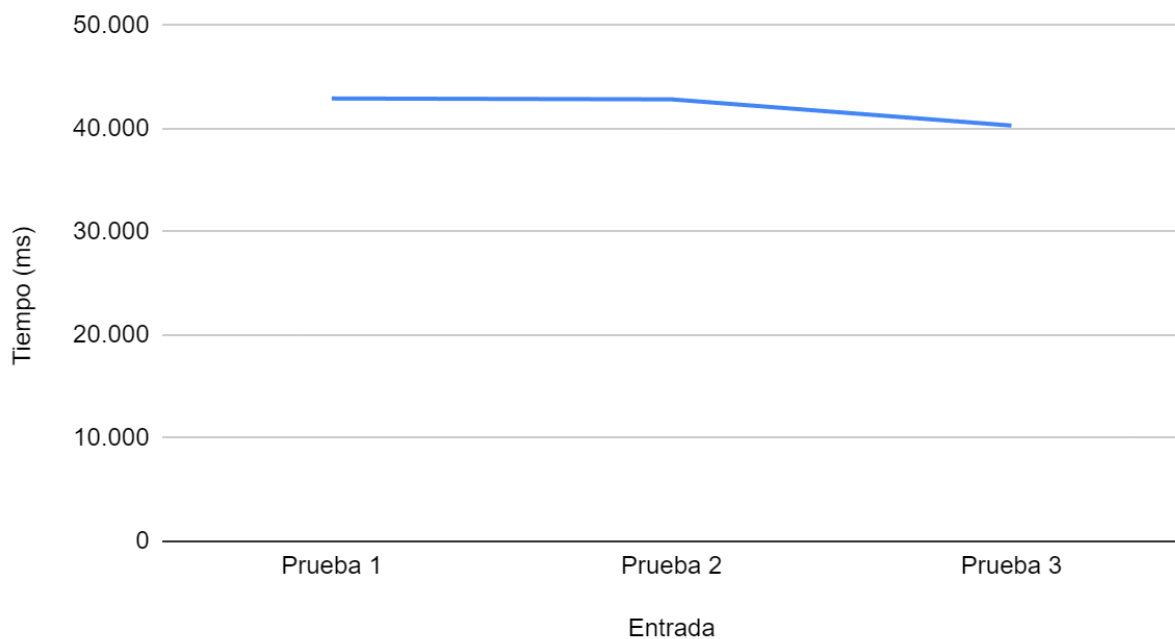
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	42.933
Prueba 2	42.849
Prueba 3	40.306

## Graficas

Las gráficas con la representación de las pruebas realizadas.

### Tiempo (ms) frente a Entrada



## Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

# Requerimiento <<3>>

## Descripción

Este requerimiento se encarga de retornar una red de trayectos desde el aeropuerto con mayor concurrencia COMERCIAL para cubrir la mayor cantidad de aeropuertos con la menor distancia posible.

<b>Entrada</b>	Estructura de datos del modelo.
<b>Salidas</b>	Aeropuerto con mayor concurrencia, distancia total de los trayectos, número total de trayectos, lista con todos los trayectos.
<b>Implementado (Sí/No)</b>	Fue implementado por Manuel Ricardo Torres

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
PrimSearch	$O(E \log V)$
Weight	$O(V+E)$
Edges	$O(E)$
<b>BFS</b>	<b><math>O(V+E)</math></b>
<b>V</b>	<b><math>O(V)</math></b>
<b>TOTAL</b>	<b><math>O(E \log V)</math></b>

## Pruebas Realizadas

Este requerimiento no necesita de parámetros de entrada: el aeropuerto de mayor concurrencia se calcula en la carga de datos.

<b>Procesadores</b>	<b>Apple Silicon M1 Pro</b>
<b>Memoria RAM</b>	16 GB
<b>Sistema Operativo</b>	MacOS Sonoma

## Tablas de datos

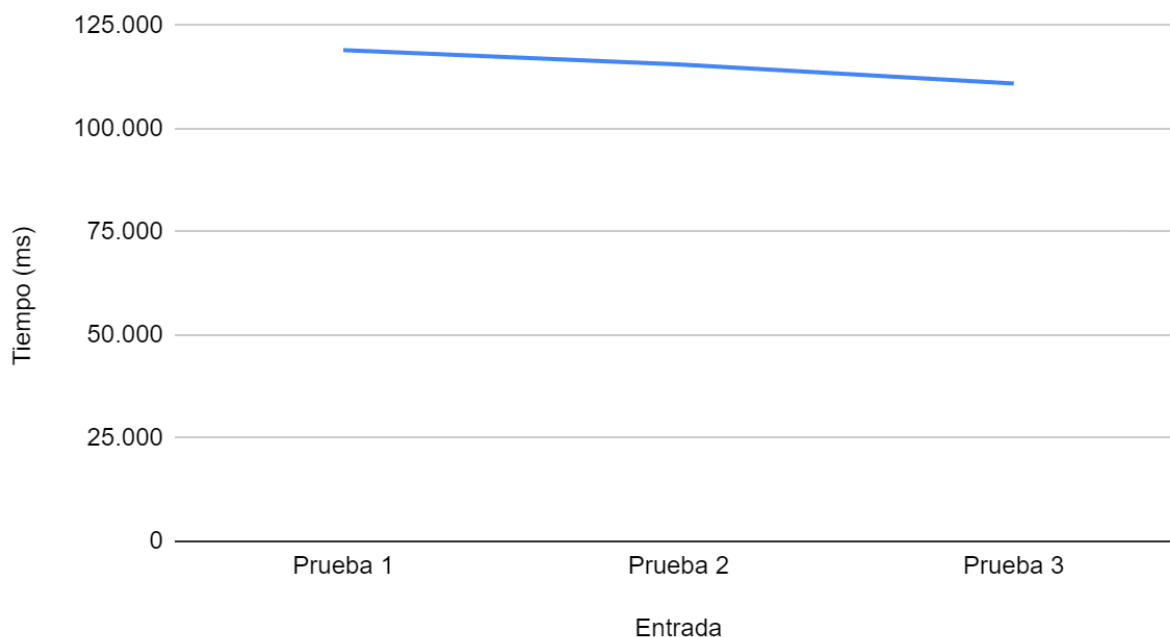
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	119.057
Prueba 2	115.593
Prueba 3	110.955

## Graficas

Las gráficas con la representación de las pruebas realizadas.

## Tiempo (ms) frente a Entrada



## Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

## Requerimiento <<4>>

### Descripción

Este requerimiento se encarga de retornar una red de trayectos desde el aeropuerto con mayor concurrencia de CARGA para cubrir la mayor cantidad de aeropuertos con la menor distancia posible.

Entrada	Estructura de datos del modelo.
Salidas	Aeropuerto con mayor concurrencia, distancia total de los trayectos, número total de trayectos, lista con todos los trayectos.
Implementado (Sí/No)	Fue implementado por Pablo Ramírez

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
PrimSearch	$O(E \log V)$
Weight	$O(V+E)$
Edges	$O(E)$
<b>BFS</b>	<b><math>O(V+E)</math></b>
<b>V</b>	<b><math>O(V)</math></b>
<b>TOTAL</b>	<b><math>O(E \log V)</math></b>

## Pruebas Realizadas

Este requerimiento no necesita de parámetros de entrada: el aeropuerto de mayor concurrencia se calcula en la carga de datos.

Procesadores	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
Memoria RAM	8 GB
Sistema Operativo	Windows 11

## Tablas de datos

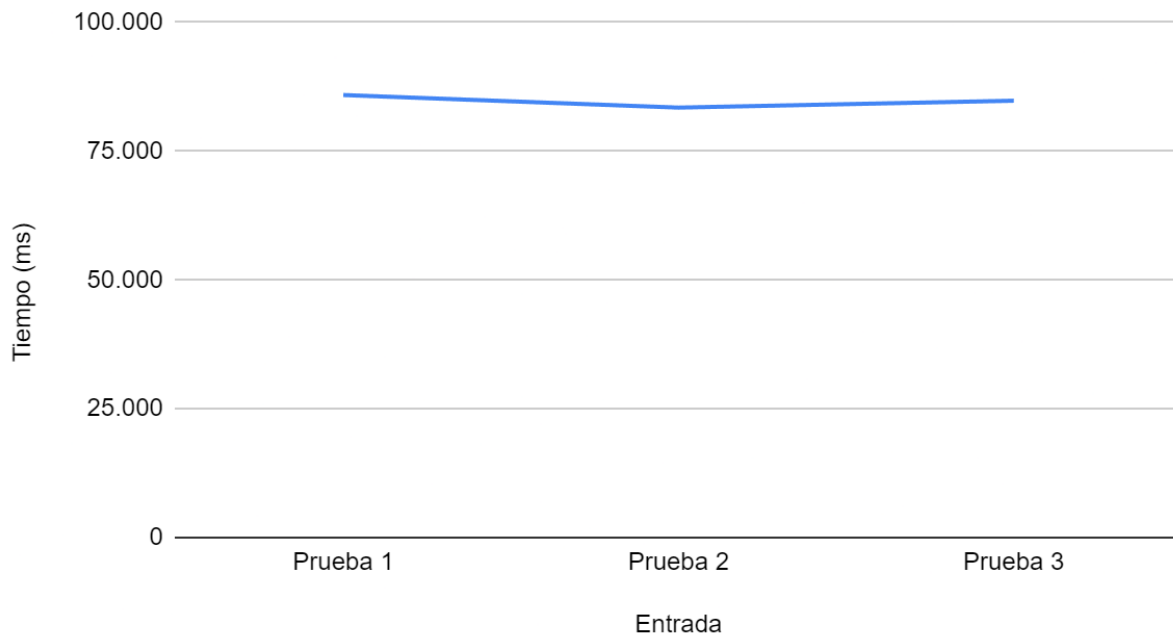
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	85.907
Prueba 2	83.463
Prueba 3	84.803

## Graficas

Las gráficas con la representación de las pruebas realizadas.

## Tiempo (ms) frente a Entrada



### Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

## Requerimiento <<5>>

### Descripción

Este requerimiento se encarga de retornar una red de trayectos desde el aeropuerto con mayor concurrencia MILITAR para cubrir la mayor cantidad de aeropuertos con la menor distancia posible.

Entrada	Estructura de datos del modelo.
Salidas	Aeropuerto con mayor concurrencia, distancia total de los trayectos, número total de trayectos, lista con todos los trayectos.
Implementado (Sí/No)	Fue implementado por Diego Alberto Ojeda

### Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo



Pasos	Complejidad
PrimSearch	$O(E \log V)$
Weight	$O(V+E)$
Edges	$O(E)$
<b>BFS</b>	<b><math>O(V+E)</math></b>
<b>V</b>	<b><math>O(V)</math></b>
<b>TOTAL</b>	<b><math>O(E \log V)</math></b>

## Pruebas Realizadas

Este requerimiento no necesita de parámetros de entrada: el aeropuerto de mayor concurrencia se calcula en la carga de datos.

Procesadores	Chip M2
Memoria RAM	16 GB
Sistema Operativo	MacOS

## Tablas de datos

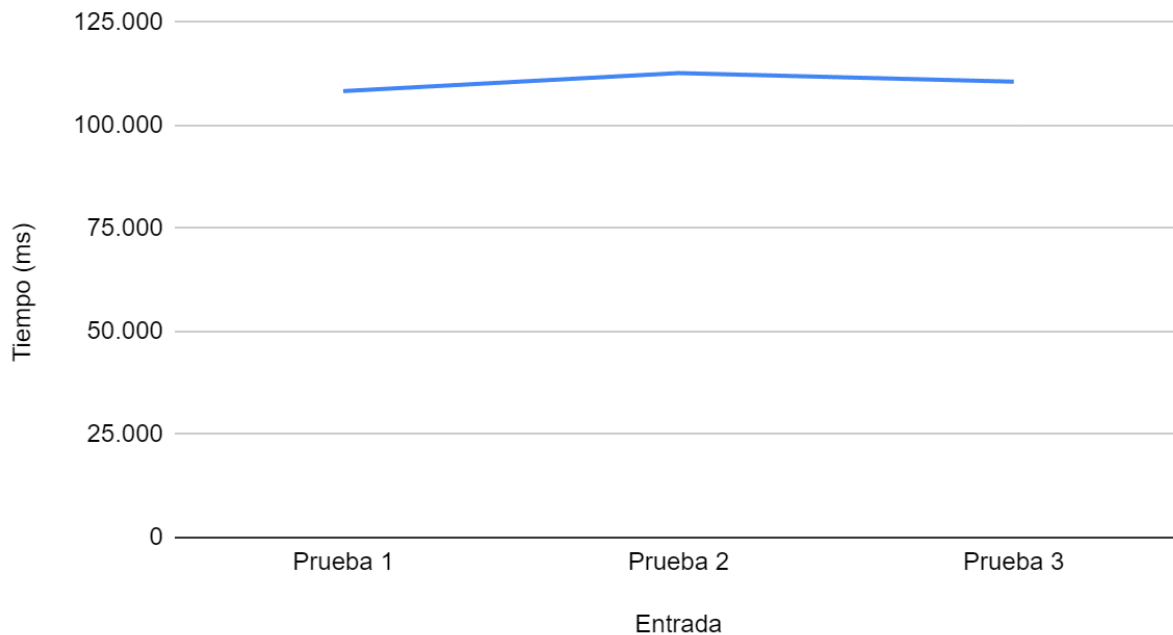
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	108.333
Prueba 2	112.708
Prueba 3	110.621

## Graficas

Las gráficas con la representación de las pruebas realizadas.

## Tiempo (ms) frente a Entrada



## Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

## Requerimiento <<6>>

### Descripción

Este requerimiento se encarga de retornar los trayectos más cortos para la cobertura de los M aeropuertos de mayor concurrencia comercial digitados por parámetro.

Entrada	Estructura de datos del modelo, top M aeropuertos de mayor concurrencia.
Salidas	Aeropuerto con mayor concurrencia comercial, lista con todos los trayectos.
Implementado (Sí/No)	Implementado por Manuel Ricardo Torres, Pablo Ramírez y Diego Ojeda

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Dijkstra	$O(V + E)$
M aeropuertos	M
<b>TOTAL</b>	<b><math>\sim O(M(V+E))</math></b>

## Pruebas Realizadas

Se realizó las pruebas con el número de aeropuertos igual a 15.

Procesadores	Apple Silicon M1 Pro
Memoria RAM	16 GB
Sistema Operativo	MacOS Sonoma

## Tablas de datos

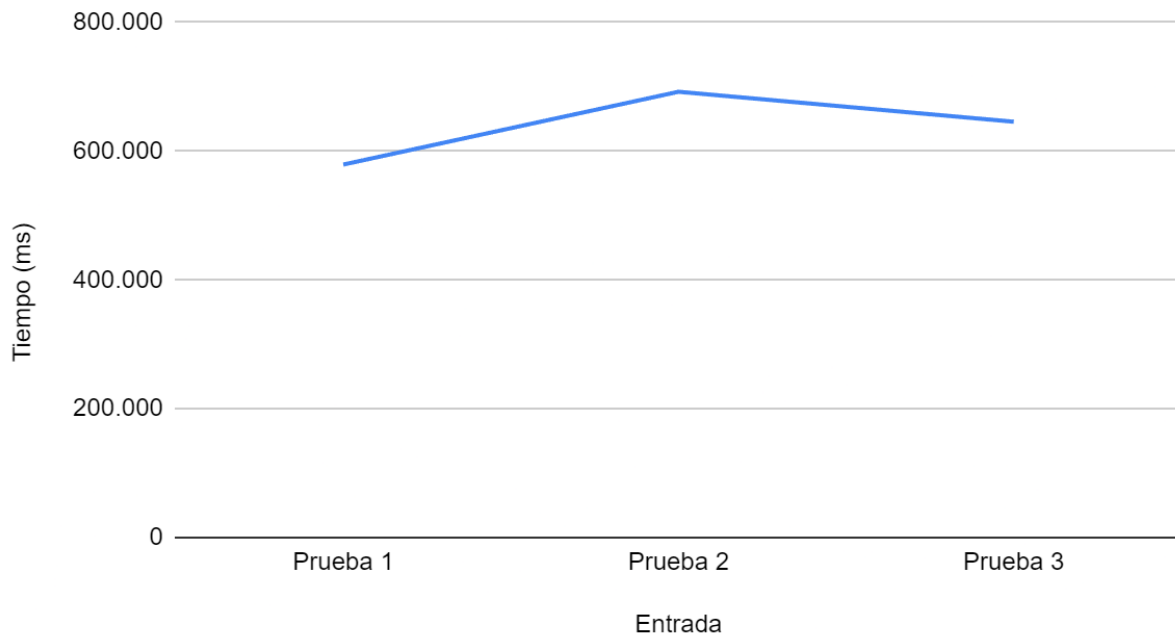
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	579.242
Prueba 2	692.312
Prueba 3	645.775

## Graficas

Las gráficas con la representación de las pruebas realizadas.

## Tiempo (ms) frente a Entrada



## Análisis

Como se observó, aunque se cargan los datos con especificaciones para que la muestra sea muy grande, se observa que la complejidad temporal aumenta linealmente excepto para los últimos dos porcentajes de datos.

## Requerimiento <<7>>

### Descripción

Este requerimiento se encarga de retornar el camino más corto entre dos puntos turísticos. Sin embargo, si no se encuentra alguno de los dos dentro de un rango de 30 KM, retorna únicamente el origen y destino (aeropuertos más cercanos a las coordenadas digitadas por parámetro) y la distancia correspondiente a dichos puntos desde los aeropuertos.

Entrada	Estructura de datos del modelo, coordenadas del punto de origen y punto destino.
Salidas	Tiempo y distancia total del camino, numero de aeropuertos que visita, lista de los aeropuertos visitados.
Implementado (Sí/No)	Implementado por Manuel Ricardo Torres

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Dijkstra	$O(V+E)$
<b>TOTAL</b>	<b><math>O(V+E)</math></b>

## Pruebas Realizadas

Se realizó la prueba de dicho requerimiento usando como datos de entrada el origen con coordenadas (Latitud 4.711, Longitud -74.246) y el destino con coordenadas (Latitud 8.633, Longitud -77.350).

Procesadores	Apple Silicon M1 Pro
Memoria RAM	16 GB
Sistema Operativo	MacOS Sonoma

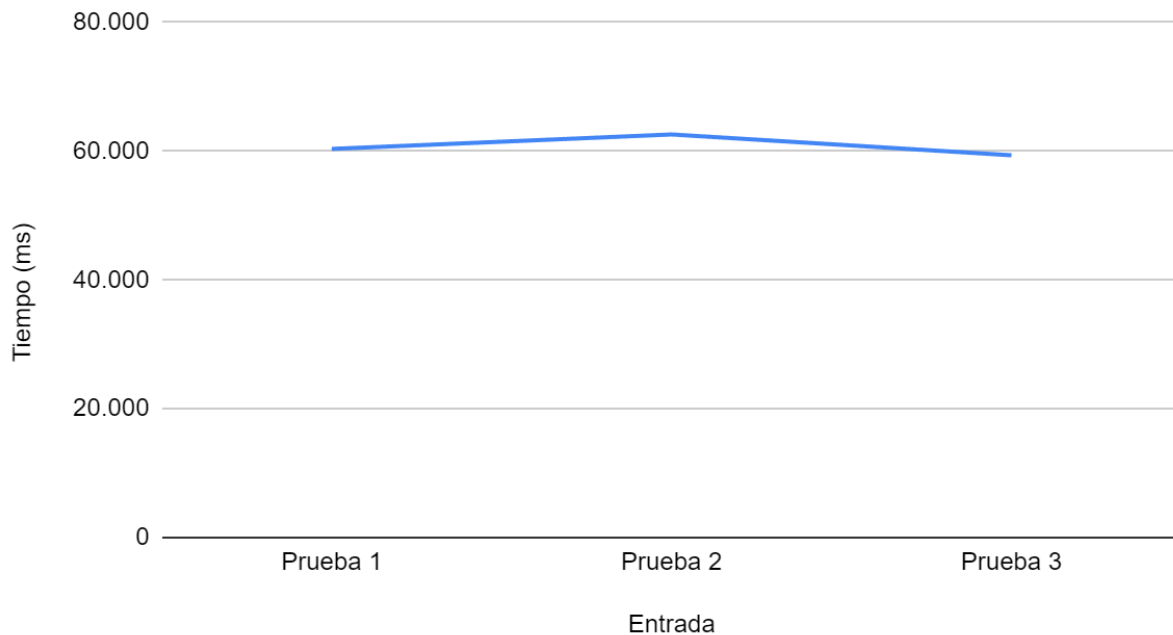
## Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
Prueba 1	60.355
Prueba 2	62.601
Prueba 3	59.369

## Graficas

## Tiempo (ms) frente a Entrada



## Análisis

La complejidad temporal depende de los arcos y vértices, por lo que su crecimiento se asemeja a una función lineal. Pero en este caso, por tener solo un archivo de prueba de datos, el crecimiento de la complejidad se asemeja a una función constante (paralela al eje x).

## Requerimiento <<8>>

### Descripción

Este requerimiento se encarga de visualizar gráficamente en un mapa interactivo mostrado en el navegador, las ubicaciones de los aeropuertos hacen parte de un requerimiento específico solicitado por el usuario.

Entrada	La lista TAD que contiene las ofertas resultantes del requerimiento.
Salidas	Un mapa construido por la librería folium
Implementado (Sí/No)	Implementado por Manuel Ricardo Torres, Pablo Ramírez y Diego Ojeda

## Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

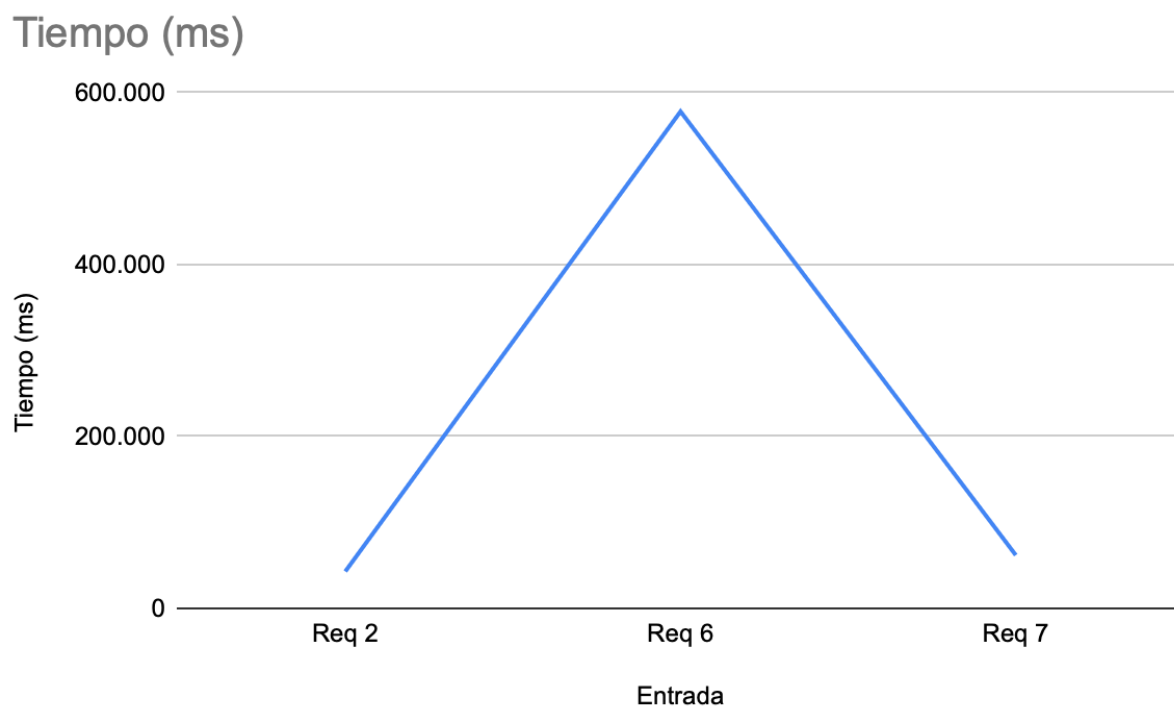
Pasos	Complejidad
Recorrer la lista de Jobs	$O(A)$ donde A es el número de aeropuertos retornados por el requerimiento.
<b>TOTAL</b>	<b><math>O(A)</math></b>

## Tablas de datos

Las tablas con la recopilación de datos de las pruebas. (De los requerimientos 2, 6 y 7).

Entrada	Tiempo (ms)
Req 2	42.933
Req 6	578.014
Req 7	61.873

## Graficas



## Análisis y pruebas realizadas

El requerimiento irá creciendo en complejidad cuando la lista de aeropuertos de entrada sea más grande. Esto, dependerá de las entradas que se les digite por parámetro.