

ANÁLISIS DEL RETO

Silvana Archila, 202220350, s.archilag@uniandes.edu.co

Juan Esteban Salamanca, 202221474, je.salamanca@uniandes.edu.co

Pablo Castellanos, 202220548, p.castellanosr@uniandes.edu.co

Requerimiento <<1>>

Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	número de partidos que se desean conocer, torneo en el que se desean conocer dichos partidos, la condición del equipo que se desea tener.
Salidas	Un tabulate de 6 partidos. Los 3 más crecientes y los 3 más antiguos dentro de los que se indican como parámetro. En caso de que no hayan tantos partidos, el tabulate será de los partidos que se tengan.
Implementado (Sí/No)	Sí, Juan Esteban Salamanca

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Determinar condición del equipo	$O(1)$
Acceder en el mapa de data structs a los valores que tienen como llave los equipos que tienen la condición que se especifica como parámetro.	$O(1)$
Iterar sobre los partidos dentro de los valores que me sirven.	$O(a)$, donde a es mucho menor a n $O(d)$, donde d es mucho menor a n
Adicionar todos los partidos que me sirvan a una lista sobre la cual obtendré la respuesta final	$O(1)$
Ordenar la lista respuesta	$O(a \log a)$ o $O(d \log d)$ dependiendo de cuál sea la condición del equipo especificada.
TOTAL	$O(a \log a)$ o $O(d \log d)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procesadores	Apple M2 con 8 núcleos de CPU (4 de alto rendimiento y 4 de alta eficiencia), 10 núcleos de GPU, Neural Engine de 16 núcleos y 100 GB/s de ancho de banda de memoria
Memoria RAM (GB)	8 GB
Sistema Operativo	macOS Ventura

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

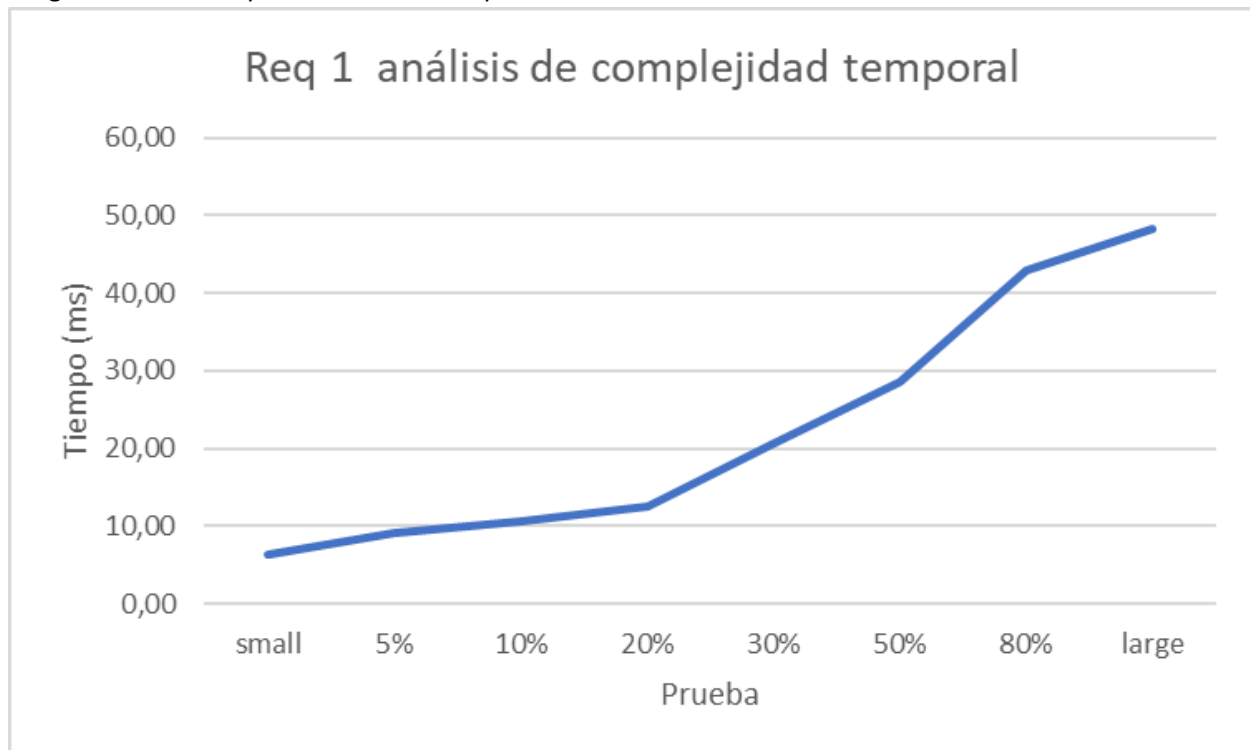
Entrada	Tiempo (ms)
small	6.27
5%	9.00
10%	10.69
20%	12.42
30%	20.61
50%	28.57
80%	42.83
large	48.30

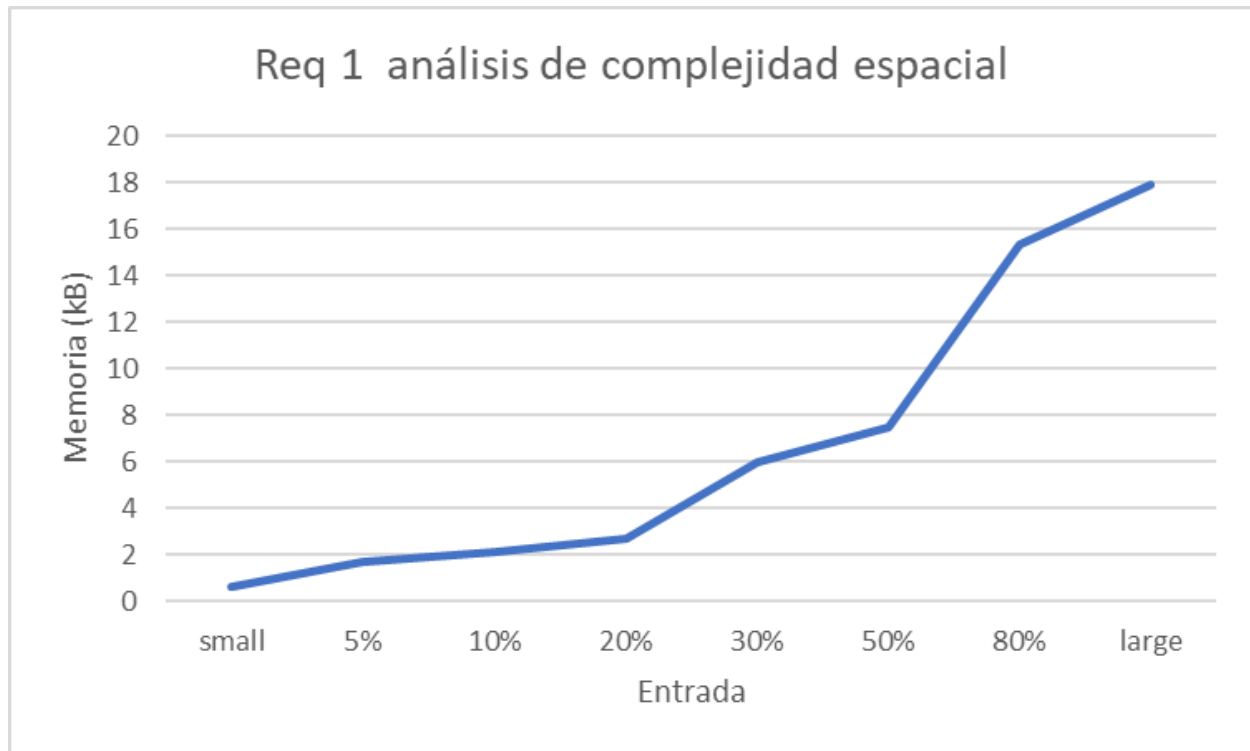
Entrada	Memoria (kB)
small	0.59
5%	1.69
10%	2.11

20%	2.67
30%	5.98
50%	7.48
80%	15.36
large	17.88

Graficas

Las gráficas con la representación de las pruebas realizadas.





Análisis

Como se puede ver, este requerimiento presenta un aumento lineáritmico en cuanto a complejidad dado que al principio no parece aumentar de forma tan significativa con respecto al aumento de datos que se presentan en cada entrada, pero a partir del 50% de carga de datos hay un aumento significativo en ambos rubros: complejidad temporal y espacial. Esto se puede deber a que las iteraciones que se hacen y las listas auxiliares que se crean tienden a crecer de forma radical.

Requerimiento <<2>>

Descripción

Para este requerimiento se hizo uso del índice ["scorers"] el cual tiene como llave en nombre del jugador . Se busca el jugador dentro del mapa y luego se entregan los goles más viejos deseados por parametro

Entrada	Tabla hash con jugadores como llaves , nombre del Jugador y número de goles
Salidas	Devuelve los primeros goles dados por número de un jugador que entra por parámetro, si son más de 6 goles devuelve los 3 primeros y los 3 últimos.
Implementado (Sí/No)	Si se implementó, por Pablo Castellanos.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Encontrar en el jugador en índice scorers	$O(1)$
Sacar el value de la pareja llave valor	$O(1)$
Hacer cmp al value el cual contiene todas las listas, donde la fecha más vieja va antes que la fechas más	$O(b \log(b))$ donde b es la cantidad de goles deseados por parámetro
Se hace una sublista con la cantidad de goles deseados	$O(1)$
En el view, se itera la respuestas dadas para hacer meterlas en una lista que luego resulta en un tabulate acorde a lo deseado	$O(b)$
TOTAL	$O(b)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procesadores	Memoria RAM (GB)	Sistema Operativo
Intel CORE I5 10th GEN	8 GB	WINDOWS 11 64 BITS

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Para la toma de datos se usaron los siguientes inputs:

jugador = "Michael Ballack"

número de goles = "7"

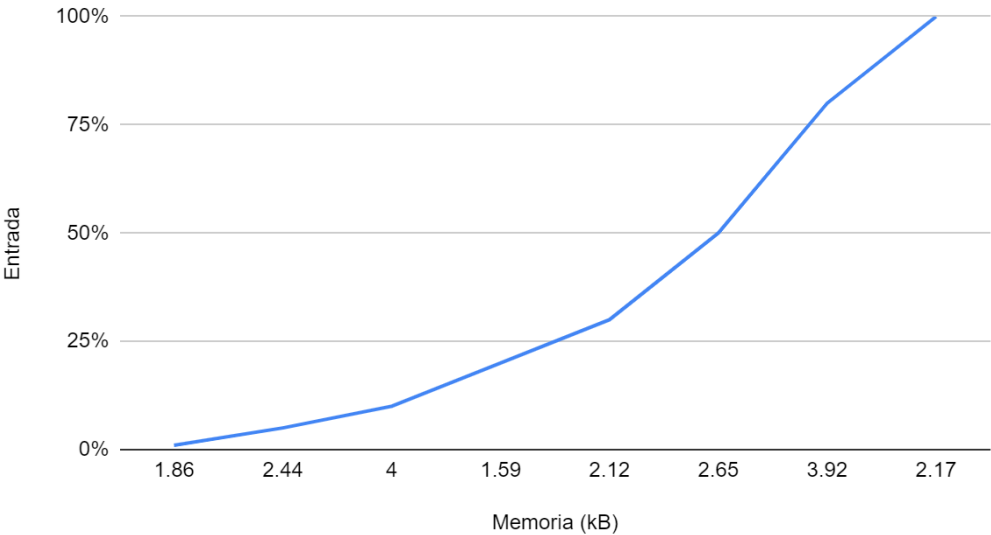
Entrada	Tiempo (ms)
small	5.02
5%	4.97
10%	5.87
20%	6.70
30%	6.33
50%	6.01
80%	6.65
large	6.79

Entrada	Memoria (kB)
small	1.86
5%	2.44
10%	4.06
20%	1.59
30%	2.12
50%	2.65
80%	3.92
large	2.17

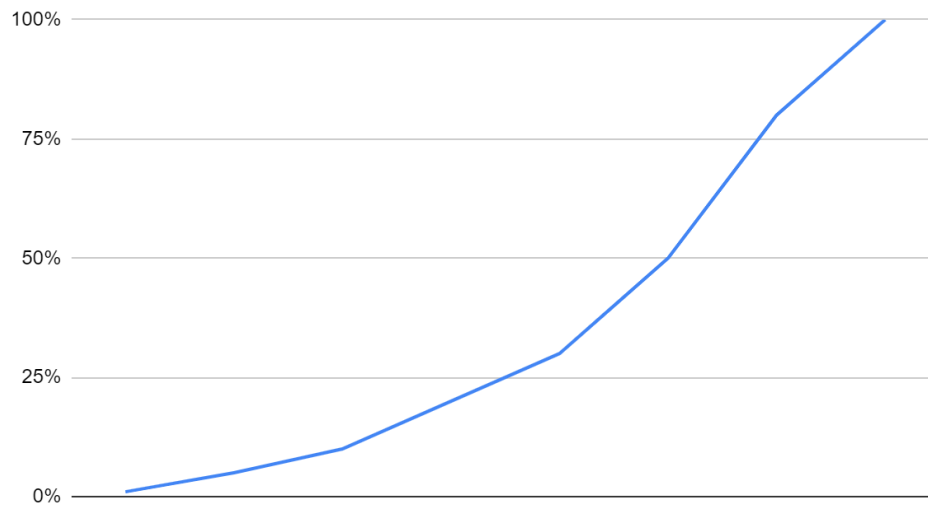
Gráficas

Las gráficas con la representación de las pruebas realizadas.

Entrada frente a Memoria (kB)



Carga vs Tiempo[ms]



Análisis

Al mantener constante el valor de goles deseados es posible evidenciar que tanto los tiempos de carga como la memoria de usada se mantiene casi estable o muy cercana a los demás datos. No obstante ambas gráficas solo se pudieron graficar de la forma inversa y por lo mismo no se puede ver la tendencia constante de la función, es decir tiene una dispersión muy baja, esto evidencia que la complejidad $O(b)$ es una complejidad fidedigna ya que ambos (memoria y tiempo) aspectos se parecen mucho a una constante.

Requerimiento <<3>>

Descripción

Para abordar el requerimiento se crearon 3 mapas en `data_structs()`. Uno que tenía de llave el nombre de un equipo y de valor la información de todos los partidos que disputó como local. Otro que tenía de llave el nombre de un equipo y de valor la información de todos los partidos que disputó como visitante. Y otro que tenía como llave el nombre de todos los equipos encontrados. Luego, usando los primeros dos mapas se filtra por las fechas dadas por el usuario por parámetro y se unen todos los datos en una sola lista.

Entrada	Team name, Start date, End date
Salidas	El total de equipos con información, el total de juegos para el equipo dado en el rango de fecha dada, el total de partidos como local y el total de partidos como visitante. Además se entrega una tabla con la información de los primeros y últimos tres partidos cargados del equipo en el rango de fecha dada.
Implementado (Sí/No)	Sí se implementó por Silvana Archila

Análisis de complejidad

Pasos	Complejidad
Extraer la información del mapa home_teams y away_teams con mp.get()	$O(M)$ siendo $M < N$
Organizar la información extraída usando merg.sort()	$O(M\log(M))$ siendo $M < N$
Recorrer la información ordenada para extraer los datos filtrados por fecha.	$O(M)$ siendo $M < N$
Encontrar el tamaño de algunas listas usando lt.size()	$O(1)$
Extraer los primeros y últimos 3 datos filtrados para presentar en una tabla.	$O(1)$
TOTAL	$O(M\log(M))$ donde $M < N$

Pruebas Realizadas

Procesadores	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 2419 Mhz, 4 procesadores principales, 8 procesadores lógicos
Memoria RAM	8 GB
Sistema Operativo	LENOVO_MT_82H_7_BU_idea_FM_Id eaPad 3 14ITL6

Tablas de datos

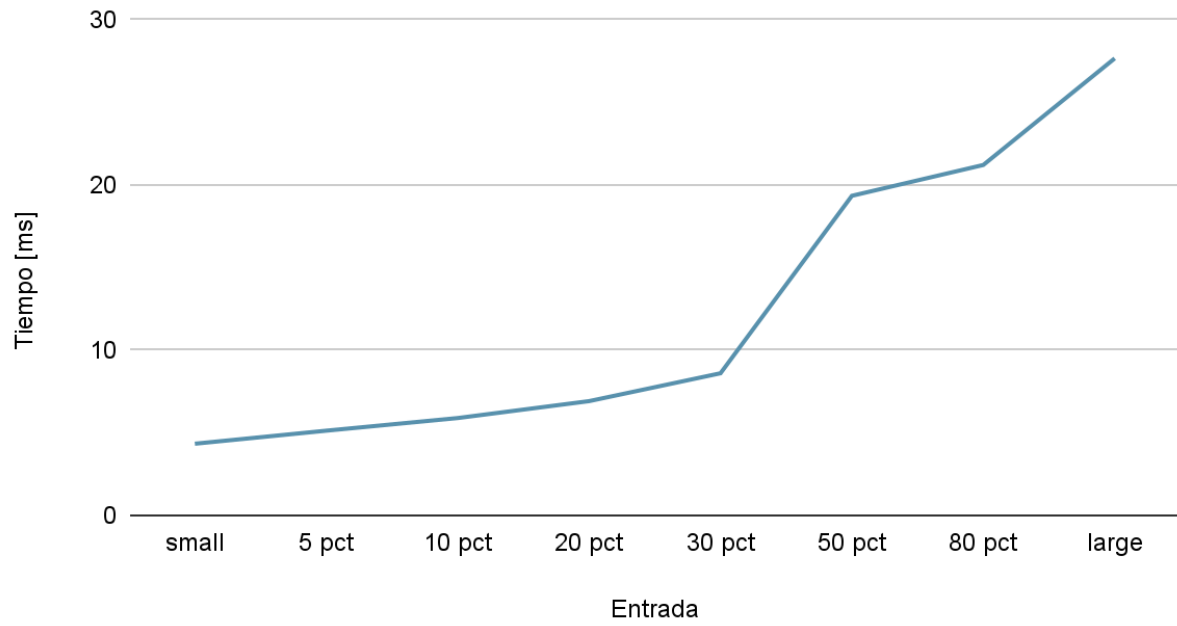
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
small	4,33689999580383
5 pct	5,12610000371933
10 pct	5,89029997587204
20 pct	6,91459995508194
30 pct	8,600300014019012
50 pct	19,341300070285797
80 pct	21,20660001039505
large	27,6564000248909

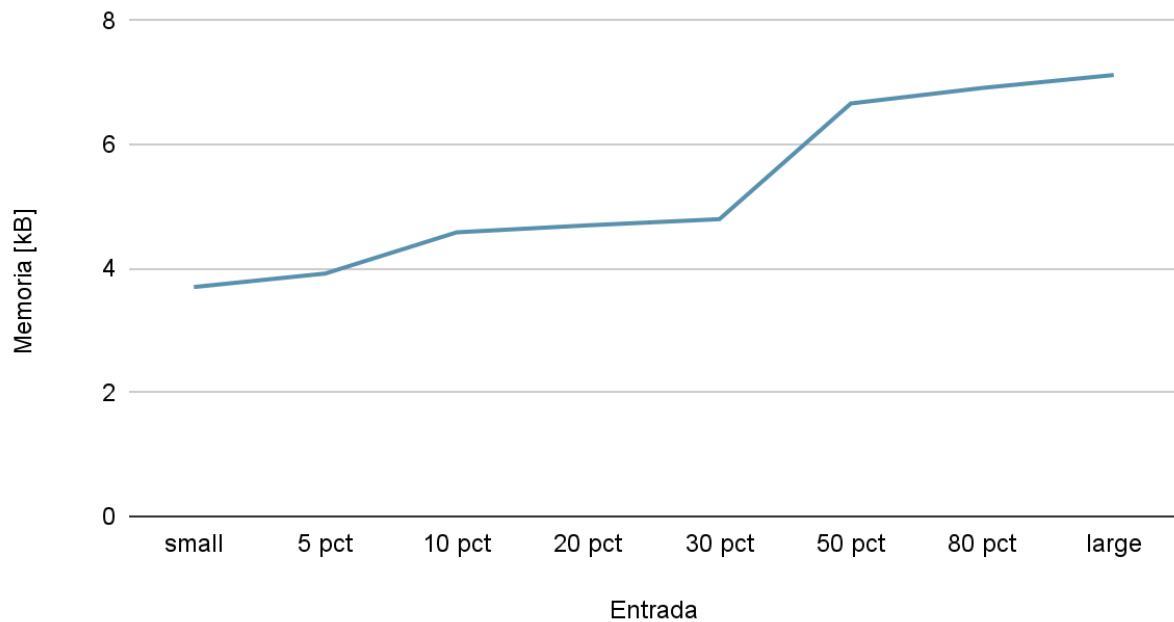
Entrada	Memoria (kB)
small	3,701171875
5 pct	3,919140625
10 pct	4,583984375
20 pct	4,697265625
30 pct	4,796875
50 pct	6,6640625
80 pct	6,9140625
large	7,123046875

Graficas

Tiempo [ms] vs Entrada Req 3



Memoria [kB] vs. Entrada Req 3



Análisis

Podemos ver en la gráfica que el tiempo de ejecución del requerimiento 3 es bastante similar a una gráfica $O(M \log(M))$. Además podemos ver que, incluso con la entrada de large el algoritmo se demora relativamente poco. Si lo comparamos con el mismo requerimiento en el reto 1 nos podemos dar cuenta que es mucho más rápido usando tablas de hash. El requerimiento 3 del reto 1 se demoraba 141 ms en large mientras que acá se demora tan solo 27 ms. Esto se debe a que el primer filtro por equipo ya está hecho en el mapa y acá solo debemos filtrar por año, lo cual hace que M sea mucho menor que N .

Requerimiento <<4>>

Descripción

Para este requerimiento se hace uso del índice["t_req_4"]. Para encontrar el total de partidos relacionados con el torneo que entra por parámetro, se busca si el partido fue ganado por penales al usar la fecha

Entrada	Data Struct, Fecha inicial, Fecha final y el torneo deseado
Salidas	Un diccionario con el total de torneos encontrados y una lista con los partidos del torneo dentro de la fecha estipulada
Implementado (Sí/No)	Si se implementó, por Pablo Castellanos.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Buscar el total de partidos asociados con el torneo	$O(1)$
Iterar en estos partidos para ver cuales están en el periodo estipulado	$O(C)$ Siendo C el total de partidos del torneo
Para el partido que cumple los requerimientos, consulta si su date está en shootouts y en caso de estarlo saca winner	$O(1)$
Se hace sort a la lista de partidos válidos	$O(C \log C)$
Se itera en el view para pasar los datos a un lista a la que se le hace tabulate	$O(C)$
TOTAL	$O(C \log c)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procesadores	Memoria RAM (GB)	Sistema Operativo
Intel CORE I5 10th GEN	8 GB	WINDOWS 11 64 BITS

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

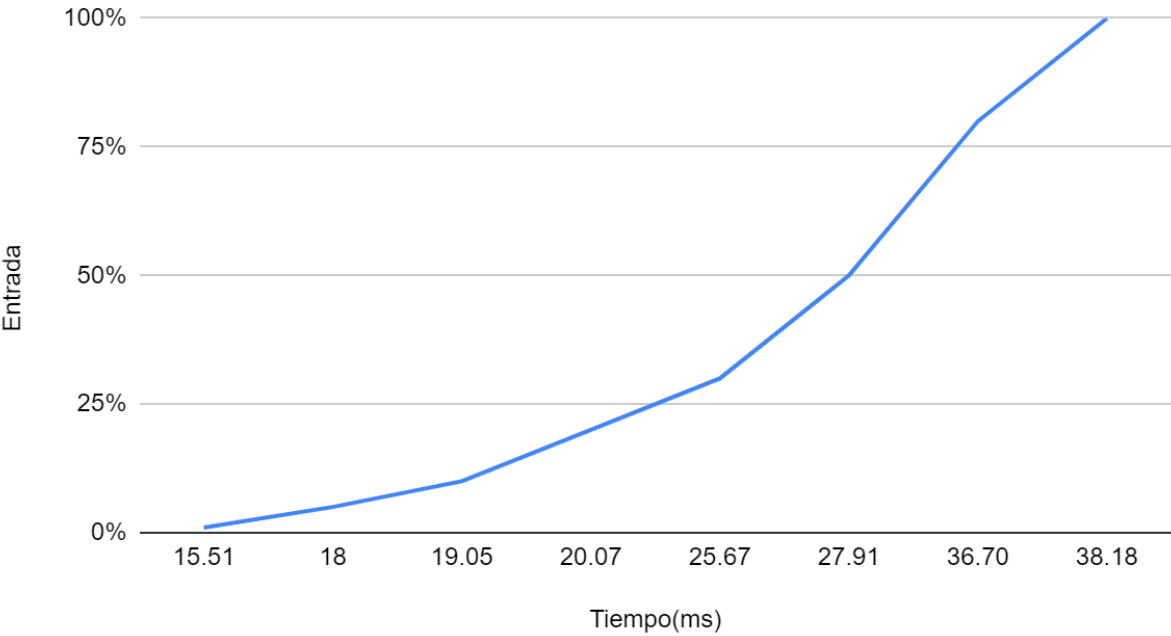
Entrada	Tiempo(ms)
small	15.51
5%	18
10%	4.06
20%	20.07
30%	25.67
50%	27.91
80%	36.70
large	38.18

Entrada	Memoria (kB)
small	4.7
5%	10.26
10%	10.06
20%	11.5
30%	9.5
50%	9.35
80%	9.24
large	9.15

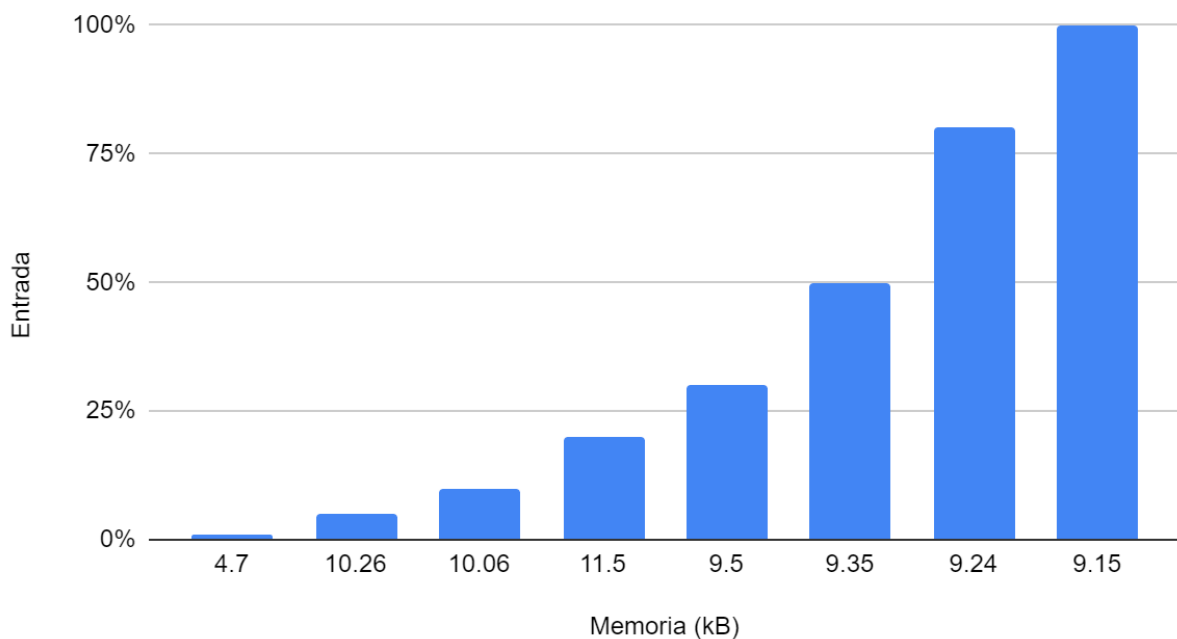
Graficas

Las gráficas con la representación de las pruebas realizadas.

Entrada frente a Tiempo(ms)



Entrada frente a Memoria (kB)



Análisis

La función cumple con los requerimientos y de igual forma tiene denota una tendencia lineal al ser que el inverso de la gráfica de entrada vs mostraría una función mucho con una tendencia logarítmica lo que hace que la complejidad sea correcta. Otra cosa a tener en cuenta es que a mayor la entrada de datos menor la memoria usada, es un fenómeno extraño pero ya se había visto antes en el Lab 7.

Requerimiento <<5>>

Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Nombre del jugador sobre el cual se quiere hacer la consulta, fecha inicial, fecha final
Salidas	Tabla con 3 partidos más recientes y 3 partidos más antiguos en donde se muestre cierta información.
Implementado (Sí/No)	Sí, Juan Esteban Salamanca

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
-------	-------------

Considerar y ubicar el nombre del jugador que se desea en mapa cuyas llaves son los nombres de los jugadores.	$O(1)$
Cuando se ubique la llave, se accede a los valores, que es una lista con todos los partidos del jugador.	$O(1)$
Se itera sobre esta lista, validando la información que se pide en el req	$O(b)$, donde b es menor a n .
Se ordena la lista cronológicamente y de acuerdo a otros parámetros dados por el enunciado.	$O(b \log b)$ puesto que se usa merge sort
TOTAL	$O(b \log b)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procesadores	Apple M2 con 8 núcleos de CPU (4 de alto rendimiento y 4 de alta eficiencia), 10 núcleos de GPU, Neural Engine de 16 núcleos y 100 GB/s de ancho de banda de memoria
Memoria RAM (GB)	8 GB
Sistema Operativo	macOS Ventura

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

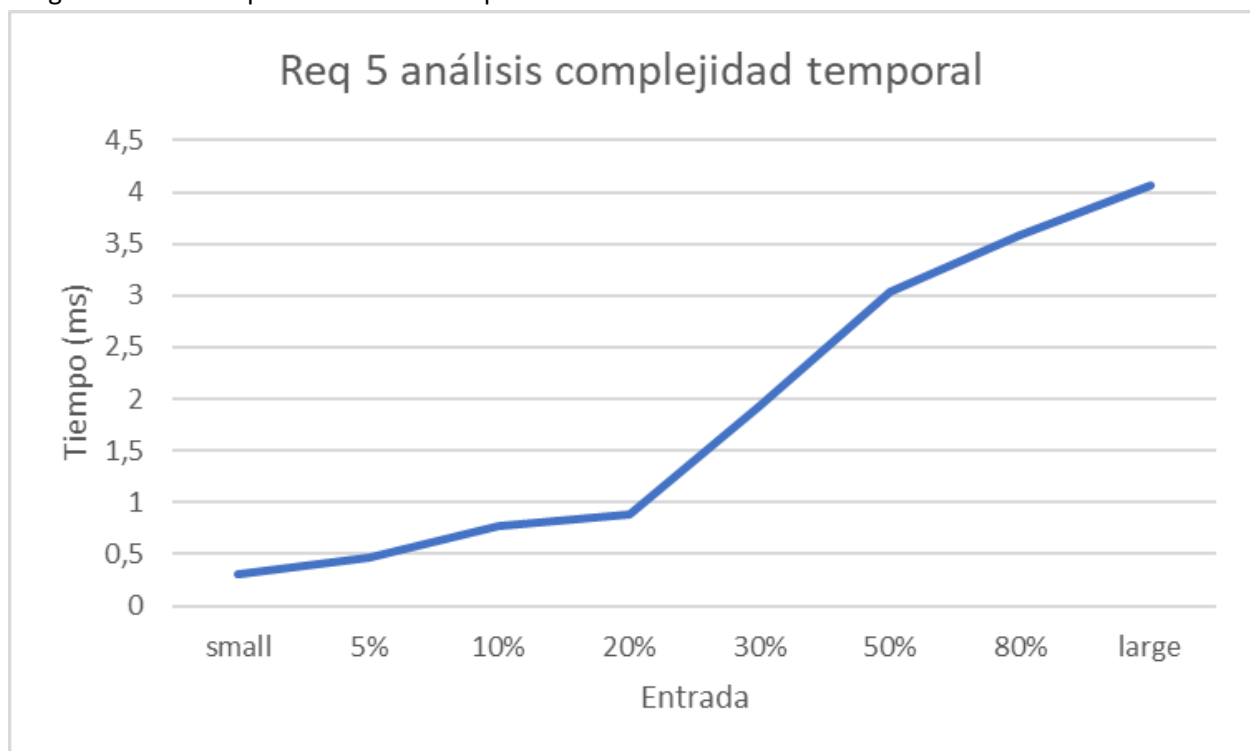
Entrada	Tiempo (ms)
small	0.31
5%	0.47

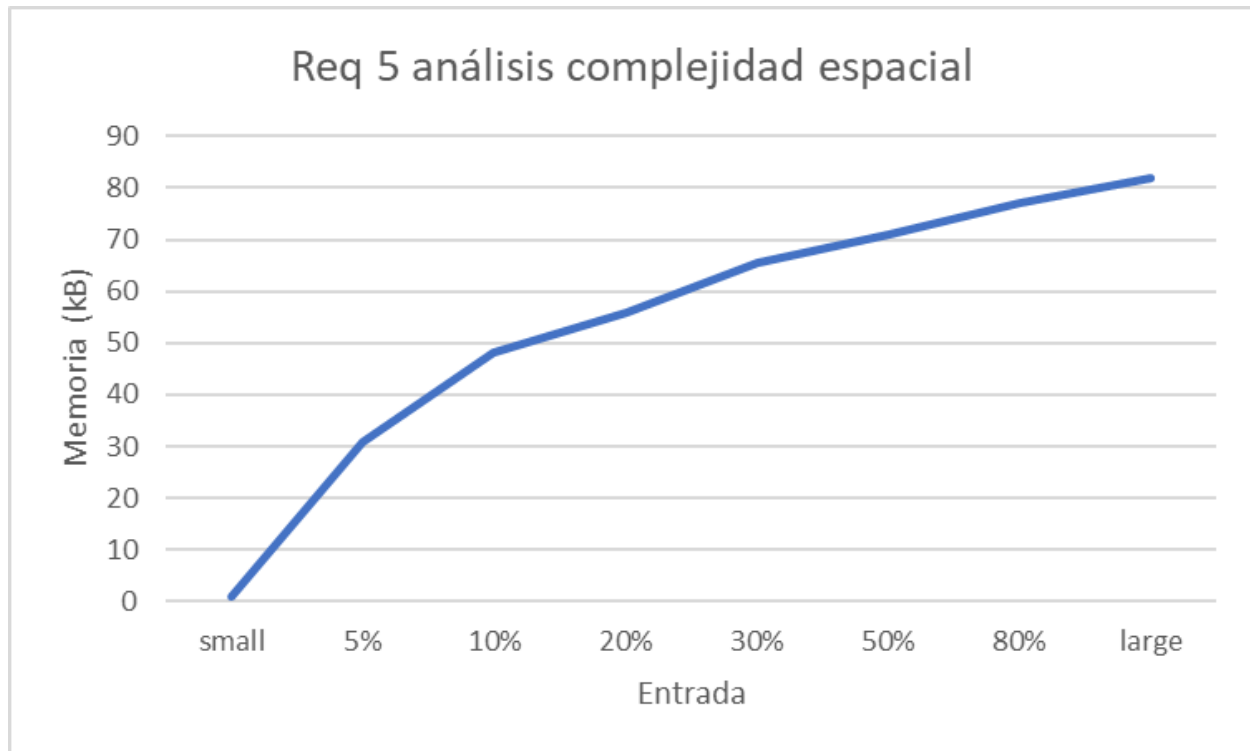
10%	0.78
20%	0.89
30%	1.93
50%	3.04
80%	3.58
large	4.06

Entrada	Memoria
small	0.85
5%	30.92
10%	48.27
20%	55.92
30%	65.47
50%	70.88
80%	77.02
large	81.84

Graficas

Las gráficas con la representación de las pruebas realizadas.





Análisis

Como se puede vislumbrar en este req, a pesar de que se presenta una implementación de una complejidad lineal en el análisis de complejidad en el primer punto de la tarea, la práctica presenta un comportamiento logarítmico. Esto puede deberse a que a diferencia de los demás requerimientos que implementé, este mantiene el uso de las listas auxiliares como mínimo, justificando que a medida que aumentan los datos que entran, no aumenta de forma tan sustancial la complejidad de memoria. Sin embargo, el proceso de validar los datos y recorrer sub listas sigue siendo el mismo, cosa que hace que la complejidad temporal sí aumente de forma significativa a partir de que se emplean el 50% de datos.

Requerimiento <<6>>

Para abordar el requerimiento se creó un mapa en `data_structs()`. El mapa tiene de llave un año y de valor tiene un diccionario con todos los torneos. Cada llave del diccionario es el nombre de un torneo y cada valor es otro diccionario con la información de los equipos que participaron en el torneo ese año, la cantidad de encuentros que hubo, la cantidad de países y ciudades donde se presentaron encuentros ese año en ese torneo, los nombres de todas las ciudades donde se presentaron encuentros y un nuevo diccionario donde está la información de cada equipo que participó en el torneo. Este último diccionario tiene por llaves los nombres de los equipos que participaron en el torneo ese año y en el valor hay un nuevo diccionario con toda la información necesaria para el requerimiento.

Luego se extrae toda la información del mapa creado y se entrega para que luego se imprima en view.

Descripción

Entrada	Año, torneo, N (Top N equipos del torneo en el año dado)
Salidas	La cantidad de torneos con información disponible en el año dado, la cantidad de equipos, partidos, países y ciudades con información disponible en el torneo y año dados, y una tabla con la información de los primeros 3 y los últimos tres equipos del top N del torneo y del año dados.
Implementado (Sí/No)	Sí se implementó por Silvana Archila

Análisis de complejidad

Pasos	Complejidad
Extraer la información del año	$O(M)$ donde $M < N$
Extraer la información del torneo	$O(1)$
Extraer la información del tamaño de varias listas usando <code>lt.size()</code>	$O(1)$
Iterar por todas las ciudades donde se disputaron partidos en el año en el torneo dado y encontrar la ciudad donde más se disputaron partidos.	$O(L)$ donde $L < N$
Ordenar la información de los equipos del torneo en el año dado usando <code>merge.sort()</code>	$O(Z \log(Z))$ donde $Z < N$
Encontrar los primeros y últimos 3 equipos con información.	$O(1)$
TOTAL	$O(Z \log(Z))$ donde $Z < N$

Pruebas Realizadas

Procesadores	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 2419 Mhz, 4 procesadores principales, 8 procesadores lógicos
Memoria RAM	8 GB
Sistema Operativo	LENOVO_MT_82H_7_BU_idea_FM_Id eaPad 3 14ITL6

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

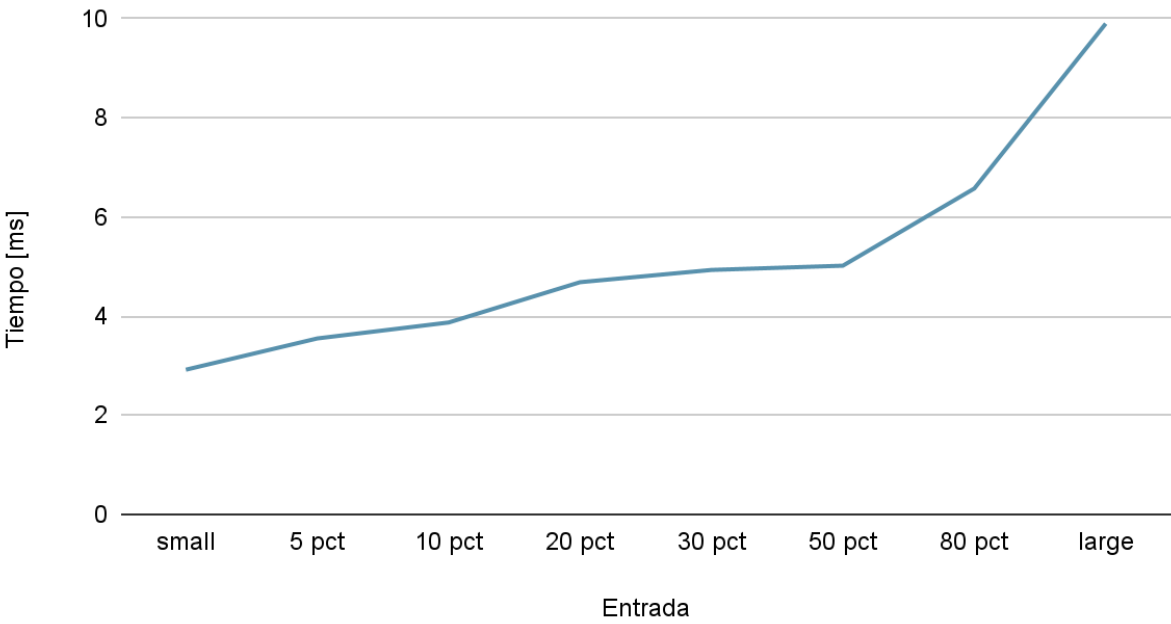
Entrada	Tiempo (ms)
small	2,918500006198883
5 pct	3,549199998378754
10 pct	3,871000051498413
20 pct	4,684099972248077
30 pct	4,930999910831451

50 pct	5,017000031471252
80 pct	6,57370001077652
large	9,8996000289917

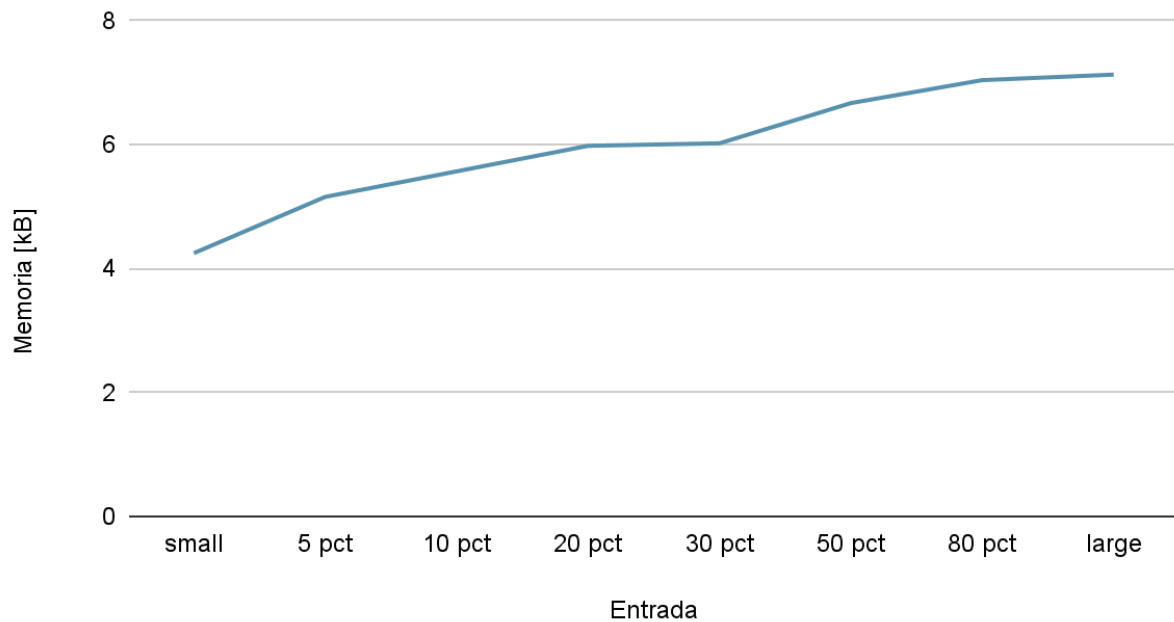
Entrada	Memoria (kB)
small	4,248046875
5 pct	5,156250010
10 pct	5,568359375
20 pct	5,980468750
30 pct	6,020703125
50 pct	6,669921875
80 pct	7,040234375
large	7,129296875

Gráficas

Tiempo [ms] vs Entrada Req 6



Memoria [kB] vs. Entrada Req 6



Análisis

En la gráfica se puede ver el comportamiento $Z\log(Z)$ de la complejidad temporal del requerimiento 6, como se predijo en el análisis de complejidad. Adicionalmente, se puede ver que el tiempo de ejecución es bastante pequeño incluso usando el archivo large. Si lo comparamos con el requerimiento 6 del reto 1 nos damos cuenta de la mejora monumental al usar tablas de hash. El requerimiento 6 en el reto 1 se demoraba 1319 ms en el archivo large, mientras que acá se demora tan solo 9,8 ms.

Esto se debe a que todos los filtros se aplican en la carga de datos al crear el mapa "years". Luego el requerimiento solo debe usar los datos filtrados. Esto significa que Z es mucho menor que N .

Requerimiento <<7>>

Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Nombre del torneo que se desea consultar, puntaje de los jugadores que se desea consultar
Salidas	Lista de estadísticas numéricas con numero de partidos, numero de goles, numero de jugadores encontrados, penales y autogoles
Implementado (Sí/No)	Sí, Juan Esteban Salamanca

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Crear variables de datos numéricos que tendre que retornar al final	$O(1)$
Ubicar los valores de torneo que debo de tener en cuenta (partidos que pertenecen al torneo introducido como parámetro) para iterar sobre ellos.	$O(1)$
Iterar sobre la lista de partidos de un torneo específico, creando listas la primera vez que se llegue a los jugadores que anotaron los goles.	$O(c)$, donde c es menor a n
En caso de que no se hayan creado las llaves y valores de estos jugadores en un mapa auxiliar, se crean. En caso de que, si estén, se actualizan estas listas. Se saben sus posiciones.	$O(1)$
Se guarda en cada iteración el último gol que anotó el jugador, para comparar con próximas iteraciones que se tengan con el mismo jugador.	$O(1)$
Finalmente, se extraen los mapas creados y se organizan de acuerdo con criterios especificados por el enunciado.	$O(c \log c)$, donde c es menor a n
TOTAL	$O(c \log c)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Procesadores	Apple M2 con 8 núcleos de CPU (4 de alto rendimiento y 4 de alta eficiencia), 10 núcleos de GPU, Neural Engine de 16 núcleos y 100 GB/s de ancho de banda de memoria
---------------------	---

Memoria RAM (GB)	8 GB
Sistema Operativo	macOS Ventura

Tablas de datos

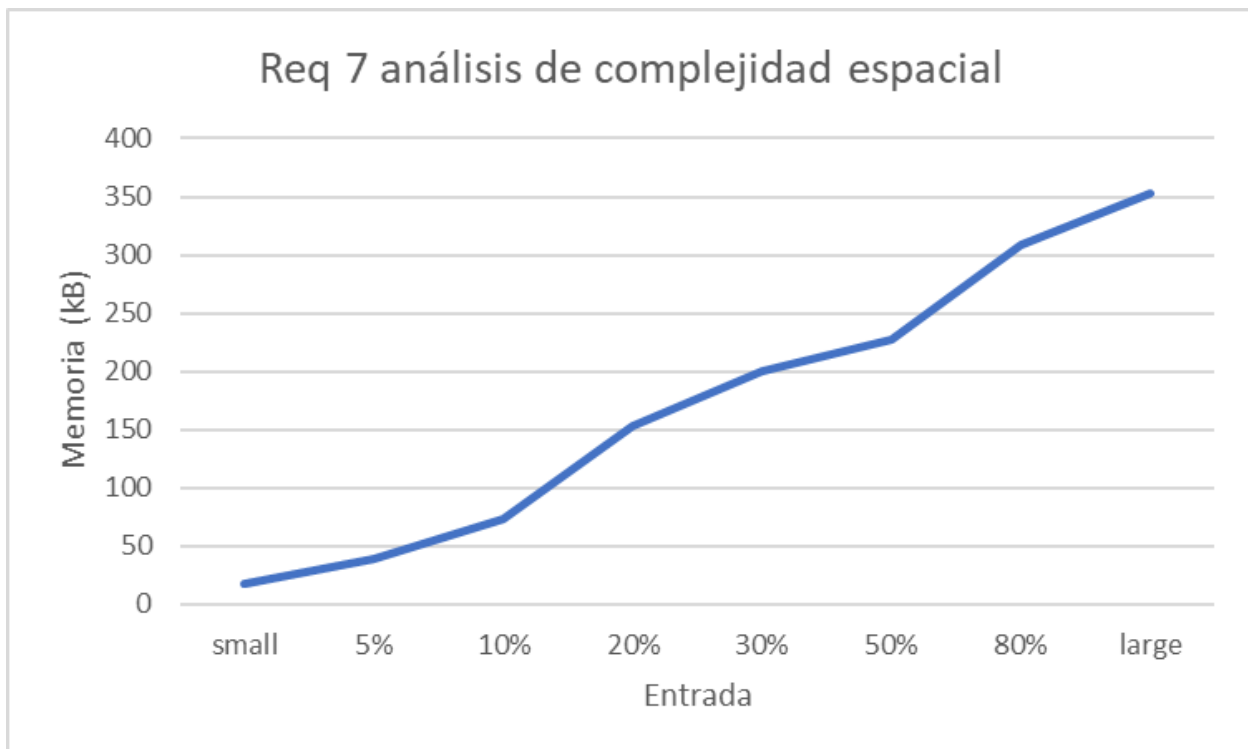
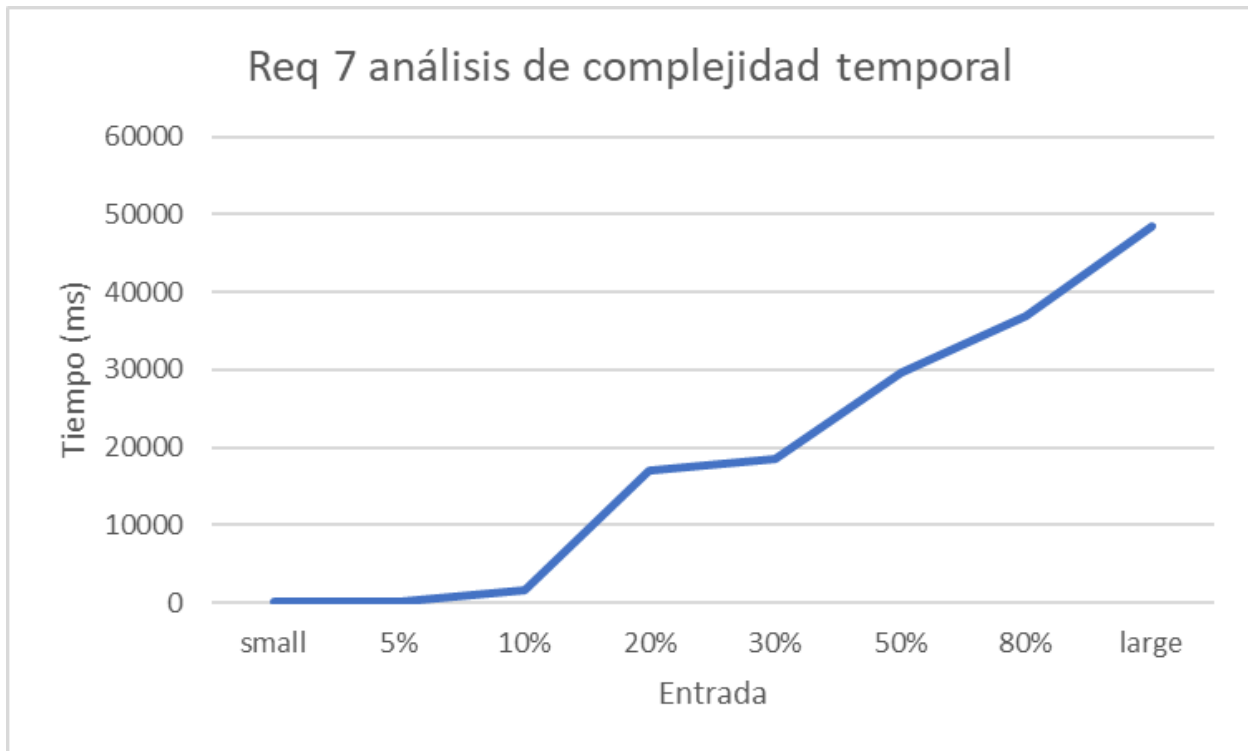
Las tablas con la recopilación de datos de las pruebas.

Entrada	Tiempo (ms)
small	8,45
5%	163,92
10%	1549,51
20%	17068,42
30%	18451,14
50%	29654,23
80%	36852,25
large	48539,8

Entrada	memoria
small	17.01
5%	38.14
10%	73.26
20%	152.78
30%	200.48
50%	226.86
80%	308.28
large	353.46

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

Como se puede apreciar en ambas gráficas de este requerimiento, la complejidad en ambos casos se dispara a partir del 30% de datos. Este comportamiento casi lineal puede deberse a la forma en la que se usan listas para crear y gestionar los datos que se van almacenando y actualizando por jugador. De igual

forma, como se explicó en el análisis de complejidad, este es uno de los pocos requerimientos que usa un mapa adicional y auxiliar dentro de la implementación del req en sí para proveer los datos que se quieren mostrar, en vez de solamente acceder a mapas ya creados anteriormente.

Requerimiento <<8>>

Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción

Breve descripción de como abordaron la implementación del requerimiento

Entrada	Parámetros necesarios para resolver el requerimiento.
Salidas	Respuesta esperada del algoritmo.
Implementado (Sí/No)	No se implemento

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1	$O(\dots)$
Paso 2	$O(\dots)$
Paso	$O(\dots)$
TOTAL	$O(\dots)$

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Entrada	Tiempo (s)

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

Análisis de resultados de la implementación, tener cuenta las pruebas realizadas y el análisis de complejidad.