RETO 1

Estructuras de Datos y Algoritmos

ISIS 1225

Laura Valentina Acosta Corredor 201911225

Juanita Puentes Mozo 201814823

Andrés Julián Bolívar Castañeda 202214834

Septiembre 25, 2023

En este informe se presenta el análisis de complejidad temporal, las pruebas de tiempos de ejecución y la comparación de los resultados obtenidos con el análisis realizado para cada uno de los requerimientos realizados en el **Reto 1**. La carga de datos se realizó utilizando la estructura de datos **ARRAY LIST** y, además, tanto para ordenar la totalidad de los datos y como para ordenar los datos de cada respuesta dada se usó el algoritmo **Mergesort**.

0 Carga de datos

Definición del problema: *La carga de datos requiere recorrer cada uno de los registros en los archivos que contienen la información de interés, para, de este modo, organizar los datos en las estructuras de datos correspondientes.*

0.1 Base de datos "Results"

Análisis de complejidad ("Results")

Teniendo en cuenta el código implementado para realizar la carga de datos del archivo "results" que contiene los registros de los resultados, la complejidad temporal de esta carga de datos es O(n), ya que el código tiene un ciclo for que se encarga de recorrer todos los registros el archivo csv.

Pruebas de tiempos de ejecución ("Results")

Para la carga de datos del archivo "results", se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Los datos se presentan en la Tabla 1.

Tamaño de la muestra	**Tiempo de ejecución [ms]**
720 (small)	16,05
3463 (5pct)	50,03
6657 (10pct)	82,46
12364 (20pct)	127,24
17486 (30 pct)	180,21
26341 (50 pct)	263,53
37713 (80 pct)	353,6
44762 (large)	434,6

Table 1: Tiempos de ejecución carga de datos "results".

Con la información de la Tabla 1, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

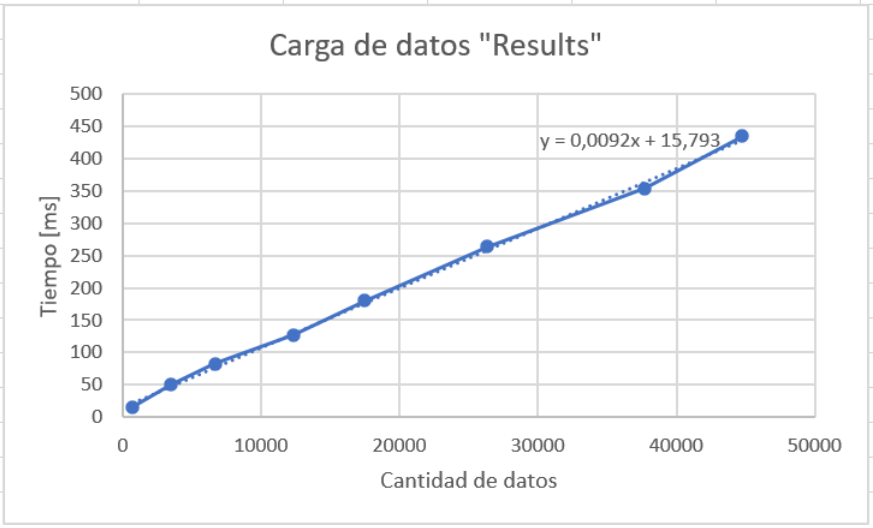

Figure 1: Tiempos de la carga de datos de "results".

Comparación de los resultados con el análisis de complejidad:
Teniendo en cuenta la linea de tendencia de la gráfica 1, la complejidad temporal de la muestra de datos está dada por el término de mayor orden que es x, por ende, la complejidad en notación O es $O(n)$ que coincide con la complejidad dada en el análisis de complejidad temporal, lo cual sustenta la afirmación establecida de que el ciclo for de la carga de datos determina una complejidad $O(n)$ para el algoritmo.

0.2 Base de datos "GoalScorers"

Análisis de complejidad ("GoalScorers"):
Teniendo en cuenta el código implementado para realizar la carga de datos del archivo "goalScorers" que contiene los registros de las anotaciones, la complejidad temporal de esta carga de datos es $O(n)$, ya que el código tiene un ciclo for que se encarga de recorrer todos los registros el archivo csv.

Pruebas de tiempos de ejecución ("GoalScorers"):
Para la carga de datos del archivo "goalScorers", se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Los datos se presentan en la Tabla 2.

Tamaño de la muestra	Tiempo de ejecución [ms]
396 (small)	7,37
2061 (5pct)	21,2
4113 (10pct)	44,53
8242 (20pct)	72,09
12316 (30pct)	110,6
20478 (50pct)	204,7
32916 (80pct)	292,47
41113 (large)	380,94

Table 2: Tiempos de la carga de datos de "goalScorers".

Con la información de la Tabla 2, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

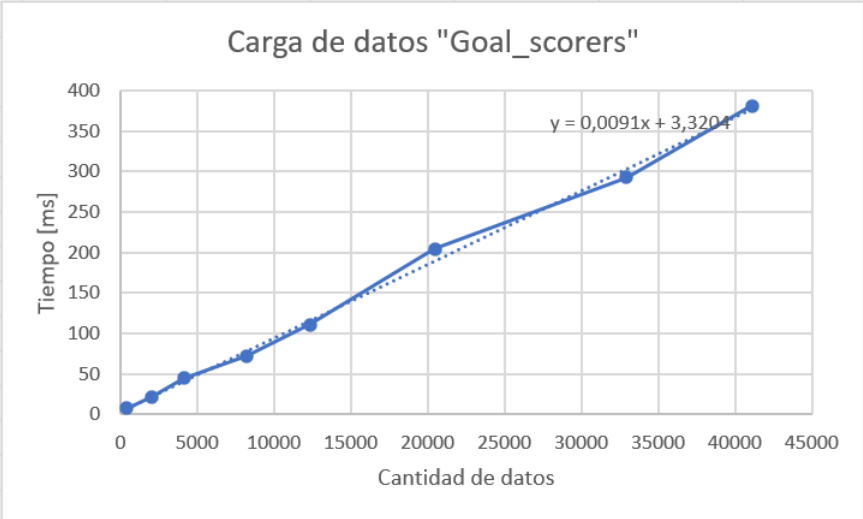

Figure 2: **T**iempos de la carga de datos de "goalScorers".

Comparación de los resultados con el análisis de complejidad:
Teniendo en cuenta la linea de tendencia de la gráfica 2, la complejidad temporal de la muestra de datos está dada por el término de mayor orden que es x, por ende, la complejidad en notación O es O(n) que coincide con la complejidad dada en el análisis de complejidad temporal, lo cual sustenta la afirmación establecida de que el ciclo for de la carga de datos determina una complejidad O(n) para el algoritmo.

0.3 Base de datos "Shootouts"

Análisis de complejidad ("Shootouts"):

Teniendo en cuenta el código implementado para realizar la carga de datos del archivo "Shootouts" que contiene los registros de las anotaciones, la complejidad temporal de esta carga de datos es $O(n)$, ya que el código tiene un ciclo for que se encarga de recorrer todos los registros el archivo csv.

Pruebas de tiempos de ejecución ("Shootouts"):
Para la carga de datos del archivo "shootoouts", se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Los datos se presentan en la Tabla 3.

Tamaño de la muestra	Tiempo de ejecución [ms]
8 (small)	0,45
39 (5pct)	0,6
70 (10pct)	0,76
123 (20pct)	1,14
205 (30pct)	1,8
316 (50pct)	2,57
468 (80pct)	3,8
555 (large)	4,44

Table 3: Tiempos de la carga de datos de "shootouts".

Con la información de la Tabla 3, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

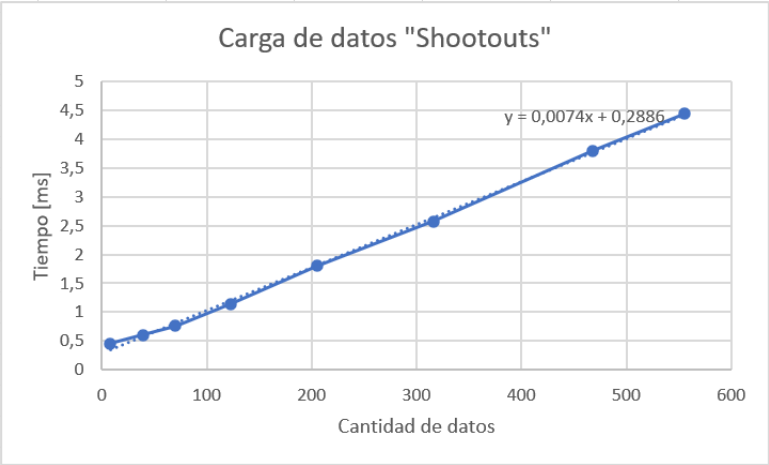

Figure 3: Tiempos de la carga de datos de "Shootouts".

Comparación de los resultados con el análisis de complejidad:
Teniendo en cuenta la linea de tendencia de la gráfica 3, la complejidad temporal de la muestra de datos está dada por el término de mayor orden que es x, por ende, la complejidad en notación O es O(n) que coincide con la complejidad dada en el análisis de complejidad temporal, lo cual sustenta la afirmación establecida de que el ciclo for de la carga de datos determina una complejidad O(n) para el algoritmo.

1 Requerimiento 1

Problem 1

En el requerimiento 1, lo que se quiere implementar es un algoritmo que le permita al usuario conocer los últimos n partidos de un equipo específico en una condición específica. De este modo, se reciben como parámetros de entrada el equipo, la cantidad de partidos que se desean consultar y la condición del equipo, ya sea "Local", "Visitante" o "Indiferente".

Análisis de complejidad:
El código implementado para solucionar este requerimiento tiene como segmento determinante un ciclo while, que en el peor de los casos deberá recorrer la totalidad de los partidos registrados en caso de que el equipo pasado por parámetro haya jugado el primer partido de la historia del fútbol. Por dicha razón, la complejidad temporal del requerimiento 1 es O(n).

Pasos	Complejidad
Creación de listas y variables	O(1)
Ciclo indeterminado sobre matches (While)	O(n)
Agregar un elemento al final (addLast)	O(1)

Table 4: Complejidad temporal para los pasos del requerimiento 1.

Pruebas de tiempos de ejecución:
Para el requerimiento 1, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Dichos registros de tiempo se realizaron ingresando como parámetros para equipo "England", ya que es uno de los equipos que jugó el primer partido de toda la historia del fútbol, cantidad de partidos "50000", esto para garantizar que se recorra la totalidad de los partidos registrados, y condición "Indiferente". Los datos se presentan en la Tabla 5.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	4,28
3463 (5pct)	10,11
6657 (10pct)	14,05
12364 (20pct)	28,30
17486 (30pct)	40,08
26341 (50pct)	59,57
37713 (80pct)	80,57
44762 (large)	104,85

Table 5: Tiempos de ejecución requerimiento 1.

Con base en la Tabla 5, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

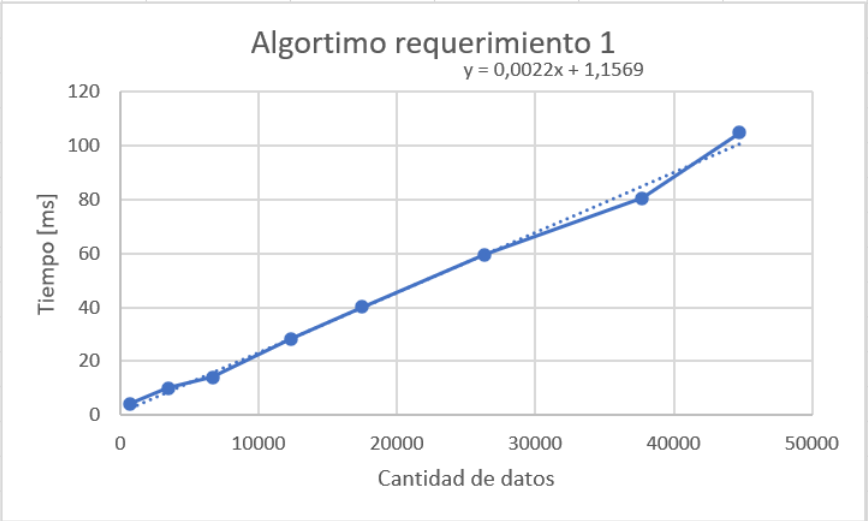

Figure 4: **T**iempos del algoritmo del requerimiento 6.

Comparación de los resultados con el análisis de complejidad:
Al establecer la línea de tendencia de la gráfica 4 y establecer la ecuación que esta línea produce, se evidencia que el término de mayor orden es x acompañado de constantes. Sin embargo, al establecer esta ecuación en notación O, se establece que la complejidad temporal dada por los resultados de las pruebas es $O(n)$. En este orden de ideas, el análisis de complejidad realizado previamente se sustenta por medio de estas pruebas, pues la complejidad que se establece por medio de los experimentos coincide con la complejidad establecida previamente. Por ende, la complejidad temporal del algoritmo implementado para el requerimiento 1 está dado por un ciclo while en el código desarrollado.

2 Requerimiento 2

Problem 2

En el requerimiento 2 se quiere implementar un algoritmo que permita conocer los primeros N goles anotados por un jugador especifico. Asi, se reciben como entrada el nombre del jugador y la cantidad de (N) primeros goles.

Análisis de complejidad:

El código implementado para solucionar este requerimiento se realiza un ciclo for, donde se recorren cada uno de los goleadores que se encuentran en el archivo *goalScorers*. Dentro de este ciclo se obtiene la información de cada goleador para determinar el total de anotaciones obtenidas y luego se ordenan. Es importante tener en cuenta que al final se ordena recursivamente la respuesta con merge sort, cuya complejidad es O(n log(n)). Por dicha razón, la complejidad temporal del requerimiento 2 es O(n log(n)). La Tabla 6 muestra las complejidades en cada uno de los pasos al implementar este requerimiento. Es importante tener en cuenta que las operaciones *getElement* y *addLast* se encuentran dentro del ciclo For.

Pasos	Complejidad
Iterar sobre goal scorers (For)	O(n)
Obtener el elemento (getElement)	O(1)
Agregar elemento al final (addLast)	O(1)
Ordenamiento recursivo (mergeSort)	O(n log(n))

Table 6: Complejidad temporal para los pasos del requerimiento 2.

Pruebas de tiempos de ejecución:

Para el requerimiento 2, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). En este caso se tomaron como parametros de entrada el nombre del jugador **Lionel Messi** considerando que es uno de los futbolistas con mayor cantidad de goles anotados, junto con un valor de N igual a 10. Los datos se presentan en la Tabla 7.

Tamaño de la muestra	Tiempo de ejecución [ms]
396 (small)	3,50
2061 (5pct)	5,84
4113 (10pct)	7.21
8242 (20pct)	11.99
12316 (30pct)	14.59
20478 (50pct)	23.49
32916 (80pct)	34,09
41113 (large)	40,28

Table 7: Tiempos de ejecución requerimiento 2.

La Figura 5 ilustra graficamente los datos descritos en la Tabla 7. En ella se puede evidenciar como cambia el tiempo de ejecución en función del tamaño de la muestra de datos.

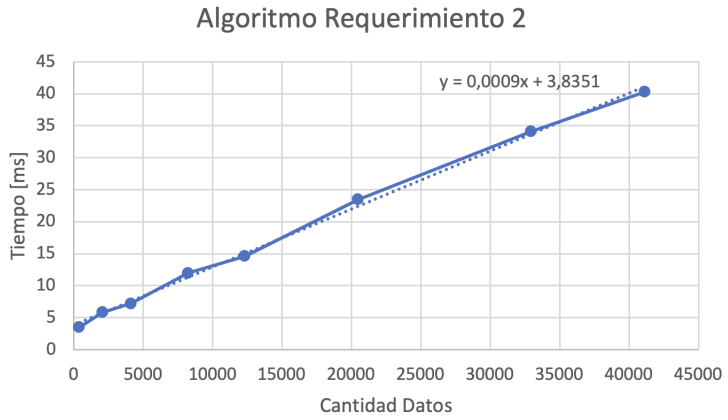

Figure 5: **T**iempos del algoritmo del requerimiento 2.

Comparación de los resultados con el análisis de complejidad:

En el análisis de la gráfica del requerimiento 2 (Figura 5), se observó que la tendencia lineal mejor se ajusta al comportamiento del algoritmo. Esto implica una relación proporcional entre el aumento del tamaño de entrada (n) y el tiempo de ejecución. Este hallazgo coincide con un polinomio de grado 1. Lo anterior, representa una complejidad temporal estimada de O(n), por lo que se estima que el termino log(n) tiene un aporte mínimo a la complejidad. Esta información resulta crucial para comprender cómo el algoritmo se comporta en función de la entrada y orienta decisiones de diseño y optimización.

3 Requerimiento 3

Problem 3

En el requerimiento 3, se espera consultar los partidos que disputo un equipo utilizando su nombre y un periodo entre dos fechas especificadas. Así, los parámetros de entrada son el nombre del equipo, la fecha inicial del periodo a consultar y la fecha final del periodo a consultar.

Análisis de complejidad

El algoritmo planteado para la resolución del requerimiento 3 comprende tres grandes partes: un primero ciclo **for** que se encarga de recorrer la información proveniente de *results* **(tamaño n)** para filtrar los partidos que se encuentran entre el rango de fechas que ingresan por parámetro. Por otra parte, hay un segundo ciclo **for** que se encarga igualmente de filtrar las anotaciones del archivo *goalScorers* por las fechas **(tamaño m)**. Finalmente, el tercer bloque de código es un ciclo anidado, mas especificamente un doble **for**. En este fragmento, se recorre la sublista filtrada de partidos **(tamaño k)** y para cada partido, se recorre la sublista de goleadores **(tamaño p)**. Esto se hizo con el objetivo de poder verificar si hubo goles desde el punto penal o autogoles. La complejidad de el ciclo de dos for anidados que recorren la sublista de partidos y de anotaciones respectivamente es O(k*p). Es importante tener en cuenta que, en la mayoría de casos, el tamaño de los partidos (n) es mayor que el tamaño de los goleadores (m) y que las dos sublistas filtradas (k y p). Por tanto, la complejidad del requerimiento 3 es O(n).

Pasos	**Complejidad**
Iterar sobre matches (For)	O(n)
Agregar un elemento al final (addLast)	O(1)
Iterar sobre goal scorers (For)	O(m)
Agregar un elemento al final (addLast)	O(1)
Doble ciclo anidado	O(k*p)
Obtener el elemento (getElement)	O(1)

Table 8: Complejidad temporal para los pasos del requerimiento 3.

Pruebas de tiempos de ejecución

Los tiempos de ejecución se registraron de manera similar a requerimientos anteriores, variando los tamaños de muestra de datos según los archivos disponibles en *Data*. Se tomaron 8 tiempos distintos, cada uno asociado a un tamaño de muestra: small (1pct), 5pct, 10pct, 20pct, 30pct, 50pct, 80pct y large (100pct). Como parametros de entrada se empleo "Colombia" como pais y el rango de fechas fue 1965-12-04 hasta 2010-11-06. Los datos obtenidos para el tiempo de ejecución se presentan en la Tabla 9.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	5,05
3463 (5pct)	11,22
6657 (10pct)	12,37
12364 (20pct)	31,59
17486 (30pct)	47,93
26341 (50pct)	62,28
37713 (80pct)	114,82
44762 (large)	139,19

Table 9: Tiempos de ejecución requerimiento 3.

Con base en esta tabla se construyó la gráfica de comportamiento de la Figura 6, en la que se muestra el cambio en el tiempo de ejecución según el tamaño de la muestra, y la linea de tendencia que mejor se ajusta a los datos.

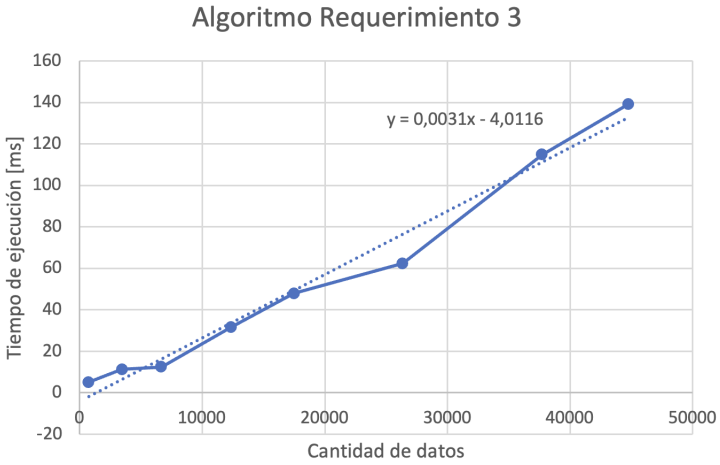

Figure 6: Tiempos del algoritmo del requerimiento 3.

Comparación de resultados con los análisis de complejidad

La gráfica representada en Figura 6 muestra la ecuación de la línea de tendencia que mejor se ajusta al comportamiento observado. Esta ecuación se corresponde con un polinomio de primer grado. En términos de notación "O" (Big O), este término representa la parte del algoritmo que contribuye de manera más significativa a la complejidad temporal general. Sin embargo, es evidente que el crecimiento se asemeja más una tendencia lineal con pendiente cercana a 1. Esto se puede explicar por el hecho de que los bucles anidados no necesitan recorrer la misma cantidad de elementos dado que fueron filtrados anteriormente.

4 Requerimiento 4

Problem 4

El requerimiento 4 consiste en la consulta de los partidos pertenecientes a un torneo de interés dentro de un rango de tiempo especificado. El programa entonces debe recibir por parámetro la información del torneo a buscar, la fecha inicial y la fecha final del periodo de tiempo de consulta.

Análisis de complejidad

El algoritmo planteado para la resolución del requerimiento 4 comprende dos grandes partes: un primero ciclo **for** que se encarga de recorrer la información proveniente de *results* para filtrar los partidos pertenecientes al torneo ingresado que se encuentran dentro del rango; y un segundo recorrido compuesto por **dos for anidados** que realizan una validación final al comparar la información con la contenida en *shootouts*. Además de esto, el algoritmo hace uso del Mergesort integrado en DISClib para ordenar la lista resultante por fecha, país y ciudad. La complejidad esta entonces regida por el ciclo de dos for anidados que recorren la lista de torneo y la de shootouts respectivamente. Teniendo un **tamaño n** para la lista de los partidos del torneo, y un **tamaño m** para la lista de shootouts, la complejidad total es entonces O(n*m). La tabla a continuación registra los ordenes de complejidad de los pasos principales empleados para la extracción de la información necesaria.

Pasos	**Complejidad**
Iterar sobre matches (For)	O(k)
Agregar un elemento al final (addLast)	O(1)
Verificar la presencia de un elemento (IsPresent)	O(k)
Iterar sobre sublista de torneos (For)	O(n)
Iterar sobre shootouts (For)	O(m)
Doble ciclo anidado	O(n*m)
Ordenar sublista de torneos (getElement)	O(n log(n))

Table 10: Complejidad temporal para los pasos del requerimiento 4.

Pruebas de tiempos de ejecución

Los tiempos de ejecución se registraron de manera similar a requerimientos anteriores, variando los tamaños de muestra de datos según los archivos disponibles en *Data*. Se tomaron 8 tiempos distintos, cada uno asociado a un tamaño de muestra: small (1pct), 5pct, 10pct, 20pct, 30pct, 50pct, 80pct y large (100pct). Como parámetros de entrada, se utilizaron los sugeridos en las instrucciones del reto, usando entonces "Copa América" como nombre del torneo, "1955-06-01" como fecha inicial, y "2022-06-30" como fecha final. Los datos obtenidos para cada tamaño de muestra pueden observarse en la Tabla 11.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	3,39
3463 (5pct)	9,22
6657 (10pct)	18,57
12364 (20pct)	42,38
17486 (30pct)	58,79
26341 (50pct)	110,56
37713 (80pct)	139,01
44762 (large)	162,03

Table 11: Tiempos de ejecución requerimiento 4.

Con base en esta tabla se construyó la gráfica de comportamiento de la Figura 7, en la que se muestra el cambio en el tiempo de ejecución según el tamaño de la muestra, y la linea de tendencia que mejor se ajusta a los datos.

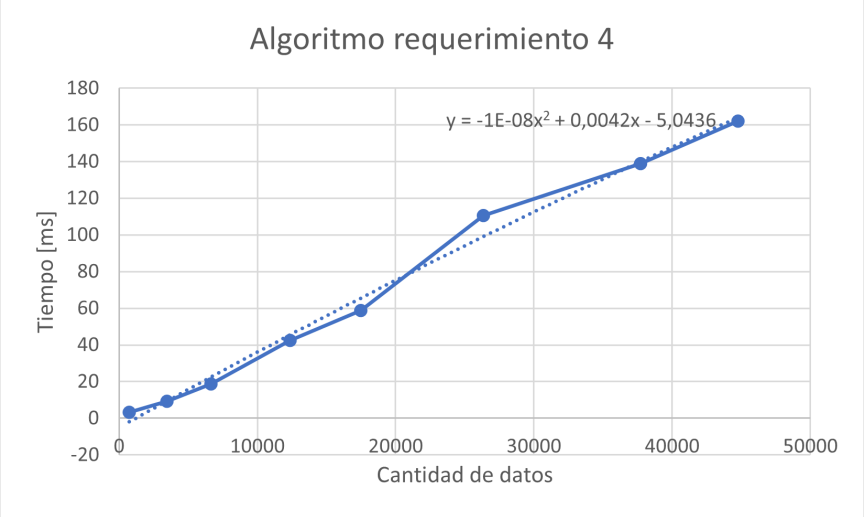

Figure 7: Tiempos del algoritmo del requerimiento 4.

Comparación de resultados con los análisis de complejidad

La gráfica de la Figura 7 presenta la ecuación de la linea de tendencia con mejor ajuste al comportamiento presentado $R^2 = 0.911$. Se observa que esta corresponde a una ecuación polinomial de grado 2, en la que predomina el término x^2 al ser el de mayor orden. Este, en notación O, es el término asociado a la sección del algoritmo que tiene un aporte más significativo a la complejidad temporal general. Sin embargo, se observa que el crecimiento se comporta más de forma casi lineal que polinomial, lo cual puede explicarse con el hecho de que el for externo y el for interno no requieren recorrer la misma cantidad de elementos. Por este motivo, el crecimiento no se da como un $O(n^2)$, sino como un $O(n * m)$ donde n y m representan las dimensiones de la lista de partidos y la lista de shootouts respectivamente.

October 12, 2023

5 Requerimiento 5

Problem 5

El requerimiento 5 consiste en consulta las anotaciones de un jugador en un periodo de tiempo específico. En este orden de ideas, los parámetros de entrada de este requerimiento son el nombre del jugador, la fecha inicial y la fecha final del periodo de tiempo a consultar.

Análisis de complejidad:

Con base en el análisis del algoritmo creado para la solución del requerimiento 5, su complejidad temporal es O(n^2). Esta conclusión se establece porque en el código implementado para dar solución a este problema, se hace presente un ciclo for anidado con un ciclo while. Para ambos ciclos, el peor de los casos es que el jugador haya anotado todas los goles de la historia del fútbol y que haya anotado en todos los partidos registrados en la historia respectivamente. Teniendo en cuenta el peor caso para ambos ciclos, el algoritmo debe recorrer la totalidad de los registros de resultados y, además, por cada resultado debe revisar la totalidad de registros de anotaciones. Por ende, la complejidad del algoritmo se puede establecer como O($n * m$), ya que los archivos de resultados y anotaciones contienen cantidades de registros similares.

Pasos	Complejidad
Creación de listas y variables	O(1)
Iterar sobre matches (For)	O(n)
Agregar un elemento al final (addLast)	O(1)
Iterar sobre goal scorers (For)	O(m)
Agregar un elemento al final (addLast)	O(1)
Iterar sobre sublista de goles	O(m)
Agregar elemento al final (addLast)	O(1)
Ciclo for anidado con ciclo while	O(n*m)
Agregar información a la respuesta (addLast)	O(1)
Iterar sobre sublista final de goles (For)	O(m)
Agregar información a la respuesta (addLast)	O(1)

Table 12: Complejidad temporal para los pasos del requerimiento 5.

Pruebas de tiempos de ejecución:

Para el requerimiento 5, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Dichos registros de tiempo se realizaron ingresando como parámetros para jugador "Lionel Messi", fecha inicial "1872-11-30" y fecha final "2023-07-16". Estas fechas se ingresaron como parámetro para que el algoritmo revise todos los registros cargados. Los datos se presentan en la Tabla 13.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	3,45
3463 (5pct)	9,47
6657 (10pct)	20,63
12364 (20pct)	43,08
17486 (30pct)	70,32
26341 (50pct)	101,76
37713 (80pct)	186,46
44762 (large)	290,23

Table 13: **T**iempos de ejecución requerimiento 5.

Con base en la Tabla 13, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

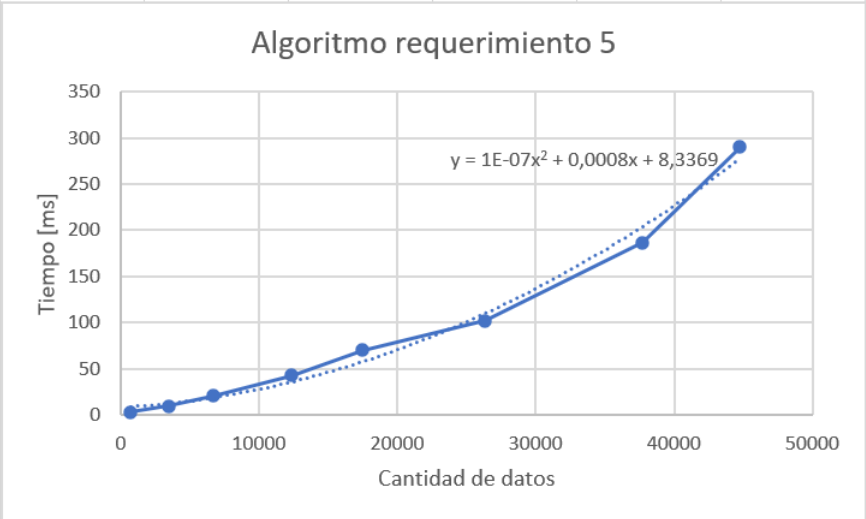

Figure 8: **T**iempos del algoritmo del requerimiento 5.

Comparación de los resultados con el análisis de complejidad:
Al establecer la línea de tendencia de la gráfica 8 y establecer la ecuación que esta línea produce, se evidencia que el término de mayor orden es x^2 acompañado de otros términos de menor orden (los cuales representan otros ciclos for usados en el algoritmo) y de constantes. Sin embargo, al establecer esta ecuación en notación O, se establece que la complejidad temporal dada por los resultados de las pruebas es O(n^2). En este orden de ideas, el análisis de complejidad realizado previamente se sustenta por medio de estas pruebas, pues la complejidad que se establece por medio de los experimentos coincide con la complejidad establecida previamente. Por ende, la complejidad temporal del algoritmo implementado para el requerimiento 5 está dado por un ciclo for anidado

con un ciclo while en el código desarrollado.

6 Requerimiento 6

Problem 6

Para el requerimiento 6, se recibe como parámetros de entrada una cantidad n de equipos, un nombre de un torneo de fútbol, una fecha inicial y una fecha final para establecer un periodo de tiempo específico. Con estos parámetros de entrada se debe retorna al usuario los n mejores equipos de dicho torneo en el periodo de tiempo dado.

Análisis de complejidad:

Al analizar el algoritmo implementado para la solución del requerimiento 6, el principal segmento de código que determina la complejidad temporal de esta solución es un ciclo for anidado con otro ciclo for, que en el peor de sus casos es que deban recorrer todos los partidos registrados y todas las anotaciones registradas en caso de que el usuario establezca como periodo de tiempo toda la historia del fútbol. En este orden de ideas, ya que los archivos de resultados y anotaciones tienen tamaños similares, se puede establecer que la complejidad temporal de esta solución es O($n * m$).

Pasos	Complejidad
Creación de listas y variables	O(1)
Iterar sobre goal scorers (For)	O(m)
Agregar un elemento al final (addLast)	O(1)
Doble ciclo anidado sobre matches y goal scorers	O(n*m)
Verificar presencia de un elemento (isPresent)	O(k)
Agregar información a la lista de respuesta	O(1)
Ordenamiento de la lista de equipos	O(k log(k))
Otros recorridos sobre listas secundarias	O(list size)

Table 14: Complejidad temporal para los pasos del requerimiento 6.

Pruebas de tiempos de ejecución:

Para el requerimiento 6, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Dichos registros de tiempo se realizaron ingresando como parámetros para cantidad de equipos "10", torneo "fifa world cup", fecha inicial "1872-11-30" y fecha final "2023-07-16". Estas fechas se ingresaron como parámetro para que el algoritmo revise todos los registros cargados para los partidos. Los datos se presentan en la Tabla 15.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	10,01
3463 (5pct)	69,52
6657 (10pct)	202,84
12364 (20pct)	673,17
17486 (30pct)	1689,04
26341 (50pct)	4556,65
37713 (80pct)	10080,43
44762 (large)	14014,72

Table 15: Tiempos de ejecución requerimiento 6.

Con base en la Tabla 15, se construyó la siguiente gráfica que representa de forma visual el comportamiento del algoritmo en función del tamaño de la muestra de datos. Dicha gráfica se presenta a continuación.

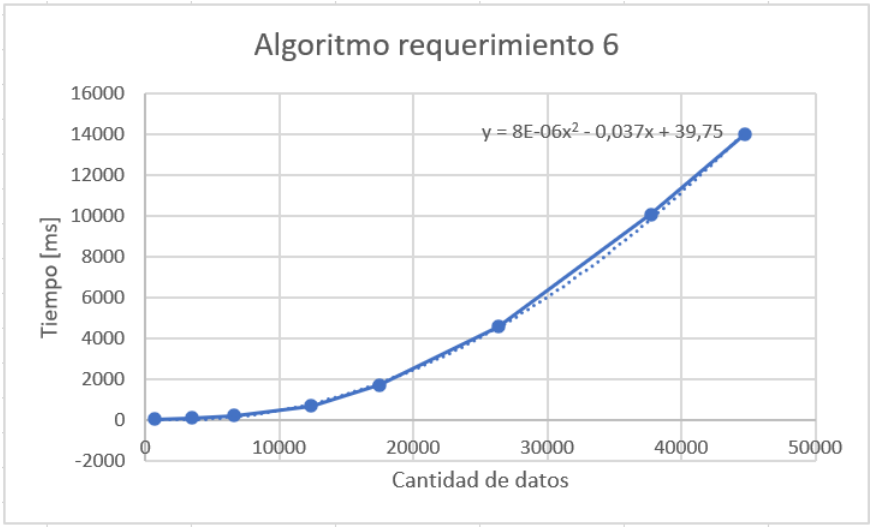

Figure 9: **T**iempos del algoritmo del requerimiento 6.

Comparación de los resultados con el análisis de complejidad:
Al establecer la línea de tendencia de la gráfica 9 y establecer la ecuación que esta línea produce, se evidencia que el término de mayor orden es x^2 acompañado de otros términos de menor orden (los cuales representan otros ciclos for usados en el algoritmo) y de constantes. Sin embargo, al establecer esta ecuación en notación O, se establece que la complejidad temporal dada por los resultados de las pruebas es $O(n^2)$. En este orden de ideas, el análisis de complejidad realizado previamente se sustenta por medio de estas pruebas, pues la complejidad que se establece por medio de los experimentos coincide con la complejidad establecida previamente. Por ende, la complejidad temporal del algoritmo implementado para el requerimiento 6 está dado por un ciclo for anidado con otro ciclo for en el código desarrollado.

October 12, 2023

7 Requerimiento 7

Problem 7

El requerimiento 7 busca clasificar los N mejores anotadores en partidos oficiales dentro de un periodo especifico. Esto puede entenderse como el TOP ranking TOP de ciertos jugadores en partidos oficiales. Así, los parametros de entrada de este requerimiento son el número (N) de jugadores para consulta, la fecha inicial del periodo a consultar y la fecha final del periodo a consultar.

Análisis de complejidad:

Al analizar el algoritmo implementado para la solución del requerimiento 7, el principal segmento de código que determina la complejidad temporal de esta solución es un triple ciclo for anidado. En este triple for es necesario recorrer la sublista de partidos filtrada **(tamaño k)**, luego la sublista de anotadores **(tamaño p)** y finalmente, las distintas anotaciones de cada uno de los goleadores **(tamaño j)**. En este orden de ideas, ya que los archivos de resultados y anotaciones tienen tamaños similares, se puede establecer que la complejidad temporal de esta solución es O(k^2). En este caso, se desprecia la contribución del tercer ciclo anidado teniendo en cuenta que solo recorre las anotaciones por jugador que generalmente son 1 o 2. Por lo que este termino es poco significativo en la complejidad temporal del algoritmo. Por otro lado, para solucionar este requerimiento se implementar otros ciclos independientes, donde las complejidades son O(n). Finalmente, se ordena la respuesta con un algoritmo recursivo (merge sort) el cual tiene una complejidad 0(nlog(n)).

Pasos	**Complejidad**
Iterar sobre matches (For)	O(n)
Agregar un elemento al final (addLast(sub$_matches$))	O(1)
Agregar un elemento al final (addLast(torneosUnicos))	O(n)
Iterar sobre goal scorers (For)	O(m)
Agregar un elemento al final (addLast)	O(1)
Triple for anidado	O(k*p*j)
Otros ciclos independientes (For)	O(n)
Ordenamiento recursivo (merge sort)	O(n log(n))

Table 16: Complejidad temporal para cada uno de los pasos del requerimiento 7.

Pruebas de tiempos de ejecución:

Para el requerimiento 7, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small(1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large(100pct). Dichos registros de tiempo se realizaron ingresando como parámetros para cantidad de jugadores que se desean consultar "18", fecha inicial "2000-12-03" y fecha final "2023-01-02". Estas fechas se ingresaron como parámetro para que el algoritmo revise los jugadores TOP en este rango de tiempo. Los resultados de tiempos de ejecución se resumen en la Tabla 17.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	193.81
3463 (5pct)	3872.04
6657 (10pct)	13971,48
12364 (20pct)	45100,28
17486 (30pct)	99313.31
26341 (50pct)	196648,37
37713 (80pct)	389862,50
44762 (large)	537351,73

Table 17: Tiempos de ejecución requerimiento 7.

Con base en la Tabla 17, se tomaron los tiempos de ejecución y tamaños de la muestra para visualizar el comportamiento de manera gráfica. La Figura 10 muestra esta tendencia.

Figure 10: Tiempos del algoritmo del requerimiento 7.

Comparación de los resultados con el análisis de complejidad:
Con base a la linea de tendencia sobre los puntos de la grafica que describen el tiempo de ejecución en función del tamaño de la muestra, concluimos que el orden mayor del polinomio es x^2. La ecuación de esta curva tambien esta acompañada de otros términos de menor orden. Por tanto, se establece que la complejidad temporal según los resultados de las pruebas es O(n^2). En este orden de ideas, el analisis teorico O($k * p$) coincide con la complejidad encontrada de manera practica: O(n^2). Es importante tener en cuenta que si bien se esta haciendo un triple for, el tercer ciclo anidado solo recorre las anotaciones de cada jugador, donde en la mayoria de casos son iguales a 1 o 2. Por lo que en realidad este ciclo se vuelve despreciable al aporte de la complejidad temporal total.

8 Requerimiento 8

Problem 8

El requerimiento 8 busca comparar el desempeño histórico anual (año por año) de dos selecciones en torneos oficiales, excluyendo los juegos amistosos, hasta una fecha específica. Los parámetros de entrada de este requerimiento son los nombres de los equipos a comparar (por separado), la fecha inicial del periodo a consultar y la fecha final del periodo a consultar. Se busca retornar las estadísticas individuales de los partidos en los años comprendidos entre las fechas ingresadas, y los datos de los enfrentamientos entre ambos equipos, si los hay.

Análisis de complejidad:

El código desarrollado para solucionar este requerimiento se compone de múltiples recorridos que buscan calcular y extraer los datos necesarios para las estadísticas individuales y en común de los equipos ingresados. Sin embargo, la complejidad de este algoritmo esta determinada por el recorrido que se hace sobre la estructura más grande (matches), en conjunto con el ordenamiento por Merge Sort que se hace sobre la misma. A continuación se presenta una síntesis de los principales pasos de ejecución, con su complejidad correspondiente:

Pasos	**Complejidad**
Ordenar matches con Merge Sort	O(n log(n))
Iterar sobre matches (For)	O(n)
Agregar un elemento al final (addLast)	O(1)
Verificar la presencia de un elemento (IsPresent)	O(n)
Iterar sobre lista de años (For)	O(k)
Doble ciclo anidado sobre goalscorers	$O(m^2)$
Otros recorridos sobre listas secundarias	O(list size)

Table 18: Complejidad temporal para los pasos del requerimiento 8.

Teniendo en cuenta lo expuesto en la tabla 18, la complejidad para este requerimiento, en notación O, es de $O(n^2 logn)$.

Pruebas de tiempos de ejecución:
Para el requerimiento 8, se tomó el tiempo de ejecución del algoritmo que soluciona este requerimiento para cada uno de los tamaños de la muestra de los de datos, es decir, para small (1pct), 5pct, 10pct, 20,pct, 30pct, 50pct, 80pct y large (100pct). Dichos registros de tiempo se realizaron ingresando como parámetros "Argentina" y "Chile" como selecciones a comparar; "1952-03-25" como fecha inicial y "2021-11-23" como fecha final. Los resultados de tiempos de ejecución se resumen en la Tabla 19.

Tamaño de la muestra	Tiempo de ejecución [ms]
720 (small)	1606.35
3463 (5pct)	126367.84
6657 (10pct)	989502.50
12364 (20pct)	7095464.48
17486 (30pct)	∞
26341 (50pct)	∞
37713 (80pct)	∞
44762 (large)	∞

Table 19: Tiempos de ejecución requerimiento 8.

Con base en la Tabla 19, se tomaron los tiempos de ejecución y tamaños de la muestra para visualizar el comportamiento de manera gráfica. La Figura 11 muestra esta tendencia.

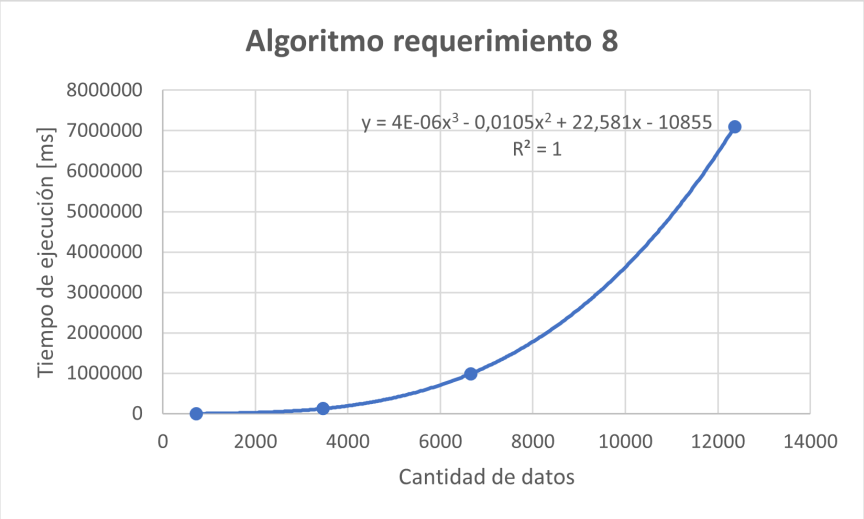

Figure 11: Tiempos del algoritmo del requerimiento 8.

Comparación de los resultados con el análisis de complejidad:

Tras realizar la gráfica de tiempos de ejecución para distintos tamaños de muestra (Figura 11), se evidenció que la línea de tendencia de mejor ajuste corresponde una polinomial cúbica, con un ajuste de $R^2 = 1$. Sin embargo, es necesario tener en cuenta que para este requerimiento no fue posible realizar la toma completa de los datos, dado que a partir de 30 pct el algoritmo toma un tiempo mayor a 6h. Se esperaría entonces que el ajuste fuera menor, lo cuál a su vez podría indicar que el orden de complejidad si estaría cercano al estimado teóricamente, es decir, O($n^2 logn$).