ANÁLISIS DEL RETO

William Pollock, 202221321, w.pollock@uniandes.edu.co

Juan Alejandro Hernández, 202225518, ja.hernandezg12@uniandes.edu.co

Juan Manuel Bonilla, 202121983, j.bonillac@uniandes.edu.co

Requerimiento 1

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

Descripción

```
def req_1(data_structs, matches, team, condition):
    """
    Función que soluciona el requerimiento 1
    """
    if condition == 'home':
        a = mp.get(data_structs['home'], team)
        b = me.getValue(a)['partidos']
        if lt.size(b) < matches:
            c = lt.subList(b, 1, lt.size(b))
        else:
            c = lt.subList(b, 1, matches)

    elif condition == 'away':
        a = mp.get(data_structs['away'], team)
        b = me.getValue(a)['partidos']
        if lt.size(b) < matches:
            c = lt.subList(b, 1, lt.size(b))
        else:
            c = lt.subList(b, 1, matches)
    x = lt.newList("ARRAY_LIST")
    lt.addLast(x, lt.getElement(c, 1))
    lt.addLast(x, lt.getElement(c, 2))
    lt.addLast(x, lt.getElement(c, 3))
    lt.addLast(x, lt.getElement(c, lt.size(c)-2))
    lt.addLast(x, lt.getElement(c, lt.size(c)-1))
    lt.addLast(x, lt.getElement(c, lt.size(c)))
    return x
```

Este requerimiento se encarga de listar los últimos N partidos de un equipo internacional. El usuario tiene la oportunidad de escoger cuál equipo buscar, cuántos partidos quiere ver y la condición de estos (local o visitante). Lo primero que hace es verificar si la condición es ‘home’. Si esta es la condición, entonces se escogen todos los partidos con condición ‘home’ en ‘partidos’. Después, estos son comparados con los partidos (en cuanto al tamaño); si b es menor que ‘matches’, entonces se crea una sublista con b en la posición 1 y con b elementos. Este mismo proceso ocurre con ‘away’. Al final, se agregan los 3 primeros y 3 últimos partidos jugados.

Entrada	Estructura de datos del modelo, los partidos jugados, los equipos y sus condiciones.
Salidas	Los últimos partidos jugados por el equipo escogido por el usuario (además de su condición).

Implementado (Sí/No)	Sí; grupal.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
mp.get(data_structs['home'], team)	O(1)
me.getValue(a)['partidos']	O(1)
lt.size(b)	O(1)
lt.sublist(b, 1, k)	O(N)
TOTAL	***O(N)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	
20 pct	Dato4	

30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas
Las gráficas con la representación de las pruebas realizadas.

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

Requerimiento 2
Plantilla para el documentar y analizar cada uno de los requerimientos.

```
def req_2(data_structs, scores, player_name):
    """
    Función que soluciona el requerimiento 2
    """
    a = mp.get(data_structs['scorers'],player_name)
    b = me.getValue(a)['partidos']
    lon = lt.size(b)
    x = lon - scores
    if lon < scores:
        c = b
    else:
        c = lt.subList(b, x, scores)
    if lt.size(c) >= 6:
        x = lt.newList("ARRAY_LIST")
        lt.addLast(x, lt.getElement(c, 1))
        lt.addLast(x, lt.getElement(c, 2))
        lt.addLast(x, lt.getElement(c, 3))
        lt.addLast(x, lt.getElement(c, lt.size(c)-2))
        lt.addLast(x, lt.getElement(c, lt.size(c)-1))
        lt.addLast(x, lt.getElement(c, lt.size(c)))
        return x
    else:
        return c
```

Descripción
Este requerimiento se encarga de listar los N primeros goles de un jugador. El usuario tiene la oportunidad de escoger cuál jugador buscar. Esto se hace cogiendo los datos de ‘scorers’ con el nombre del jugador (los cuales están en ‘partidos’).

Entrada	Estructura de datos del modelo, goles y nombre del jugador.
Salidas	Los primeros goles metidos por el jugador escogido por el usuario.
Implementado (Sí/No)	Sí; grupal

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	

20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas
Las gráficas con la representación de las pruebas realizadas.

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

Requerimiento 3
Plantilla para el documentar y analizar cada uno de los requerimientos.

Descripción
Breve descripción de como abordaron la implementación del requerimiento

Entrada	Datos, el nombre del equipo, fecha inicial y fecha final
Salidas	Lista que contenga la búsqueda necesaria
Implementado (Sí/No)	Si y lo implemento Juan Bonilla

Análisis de complejidad
Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB

Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	0.450
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	
20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

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

Requerimiento 4

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)	Si se implementó y quien lo hizo.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	

20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas
Las gráficas con la representación de las pruebas realizadas.

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

Requerimiento 5
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)	Si se implementó y quien lo hizo.

Análisis de complejidad
Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB

Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	
20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

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

Requerimiento 6

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)	Si se implementó y quien lo hizo.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB
Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	

20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas
Las gráficas con la representación de las pruebas realizadas.

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

Requerimiento 7
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)	Si se implementó y quien lo hizo.

Análisis de complejidad
Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Paso 1	O(...)
Paso 2	O(...)
Paso ….	O(...)
TOTAL	***O(...)***

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	**Intel ® Core ™ i3-1011U CPU @ 2.10 GHz 2.59 GHz**
Memoria RAM	8 GB

Sistema Operativo	Windows 10 Home – 64 bits

Entrada	**Tiempo (ms)**
small	
5 pct	
10 pct	
20 pct	
30 pct	
50 pct	
80 pct	
large	

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	**Salida**	**Tiempo (ms)**
small	Dato1	
5 pct	Dato2	
10 pct	Dato3	
20 pct	Dato4	
30 pct	Dato5	
50 pct	Dato6	
80 pct	Dato7	
large	Dato8	

Graficas

Las gráficas con la representación de las pruebas realizadas.

Análisis

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

Requerimiento Ejemplo

Descripción

```python
def get_data(data_structs, id):
    """
    Retorna un dato a partir de su ID
    """
    pos_data = lt.isPresent(data_structs["data"], id)
    if pos_data > 0:
        data = lt.getElement(data_structs["data"], pos_data)
        return data
    return None
```

Este requerimiento se encarga de retornar un dato de una lista dado su ID. Lo primero que hace es verificar si el elemento existe. Dado el caso que exista, retorna su posición, lo busca en la lista y lo retorna. De lo contrario, retorna None.

Entrada	Estructuras de datos del modelo, ID.
Salidas	El elemento con el ID dado, si no existe se retorna None
Implementado (Sí/No)	Si. Implementado por Juan Andrés Ariza

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	**Complejidad**
Buscar si el elemento existe (isPresent)	O(n)
Obtener el elemento (getElement)	O(1)
TOTAL	***O(n)***

Pruebas Realizadas

Las pruebas realizadas fueron realizadas en una maquina con las siguientes especificaciones. Los datos de entrada fueron el ID 1.

Procesadores	**AMD Ryzen 7 4800HS with Radeon Graphics**
Memoria RAM	8 GB
Sistema Operativo	Windows 10

Entrada	**Tiempo (ms)**
small	0.05
5 pct	0.33
10 pct	1.28
20 pct	2.54
30 pct	4.98
50 pct	7.51
80 pct	13.81
large	25.97

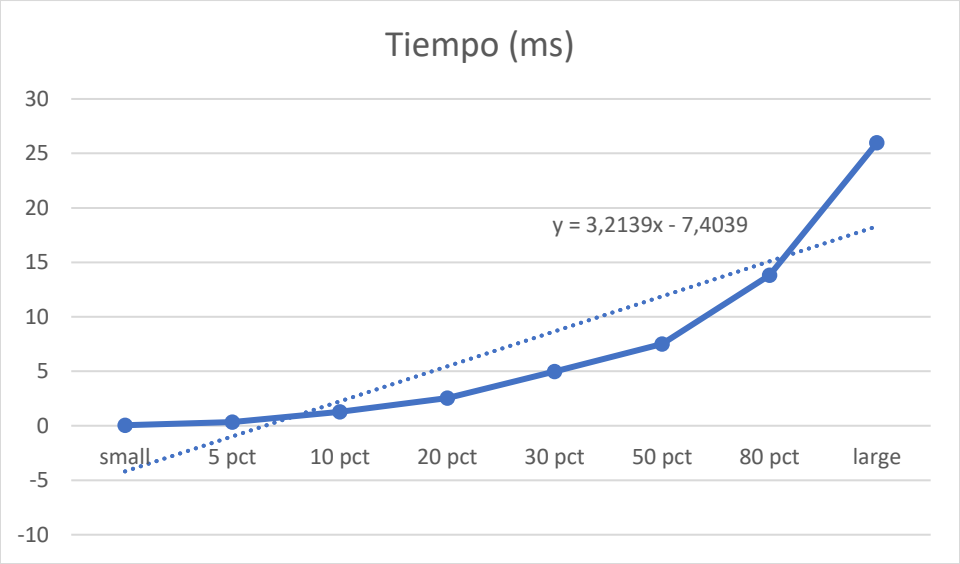
Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Muestra	Salida	Tiempo (ms)
small	Dato1	0.05
5 pct	Dato2	0.33
10 pct	Dato3	1.28
20 pct	Dato4	2.54
30 pct	Dato5	4.98
50 pct	Dato6	7.51
80 pct	Dato7	13.81
large	Dato8	25.97

Graficas

Las gráficas con la representación de las pruebas realizadas.



Análisis

A pesar de que obtener un elemento en un *ArrayList,* dada su posición, tiene complejidad constante, la implementación de este requerimiento tiene un orden lineal O(n). Esto debido a que, lo primero que se hace es verificar si el elemento hace parte de la lista. Específicamente, a la hora de buscar un elemento en una lista, en el peor de los casos es necesario recorrer toda la lista, es decir, complejidad lineal.

Este comportamiento se puede evidenciar experimentalmente en la gráfica. Ya que, gracias a que los datos no se encuentran tan dispersos con respecto a la línea de tendencia, la curva coincide con el comportamiento lineal esperado.