**ANÁLISIS DEL RETO 2**

Juliana Rodríguez Morales – 202421552 – js.rodriguezm1234

Maria Clara Quijano - 202420069 - m.quijanoa

Juan Andrés Lozada - 202510410-j.lozadab

# **Requerimiento <<n>>**

## **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(...)*** |

## **Análisis**

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

# **Requerimiento Ejemplo**

## **Descripción**



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)*** |

## **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.

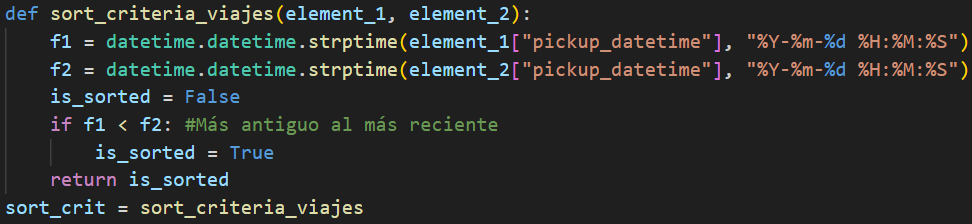
# **Requerimiento 1**

## **Descripción**

Función req\_1Texto

El contenido generado por IA puede ser incorrecto.

Función auxiliar de sort\_criteria para el quick\_sort del req\_1



A partir de un rango de fechas dado (fecha inicial y final de recogida) se deben mostrar los primeros n viajes y últimos n viajes previamente ordenados del más antiguo al más reciente.

La función de sort criteria viajes compara fecha y hora de recogida para organizar del más antiguo al más reciente al momento de utilizar quick\_sort.

|  |  |
| --- | --- |
| **Entrada** | Fecha y hora inicial de recogida  Fecha y hora final de recogida  Tamaño de muestra |
| **Salidas** | Tiempo de ejecución  # de trayectos dentro del rango de fecha/hora  N primeros y N últimos:  Fecha y hora de inicio  Coordenadas iniciales  Fecha y hora final  Coordenadas finales  Distancia recorrida (millas)  Costo total pagado |
| **Implementado (Sí/No)** | Sí, Juliana Rodríguez |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo. Solo se analizará el req\_1, no sort\_crit.

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

## **Análisis**

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

# **Requerimiento 2**

## **Descripción**

Breve descripción de como abordaron la implementación del requerimiento

|  |  |
| --- | --- |
| **Entrada** | Parámetros necesarios para resolver el requerimiento. |
| **Salidas** | Tiempo de ejecución  # de trayectos dentro del rango de fecha/hora  N primeros y N últimos:  Fecha y hora de inicio  Coordenadas iniciales  Fecha y hora final  Coordenadas finales  Distancia recorrida (millas)  Costo total pagado |
| **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(...)*** |

## **Análisis**

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

# **Requerimiento 3**

## **Descripción**

Breve descripción de como abordaron la implementación del requerimiento

|  |  |
| --- | --- |
| **Entrada** | Parámetros necesarios para resolver el requerimiento. |
| **Salidas** | Tiempo de ejecución  # de trayectos dentro del rango de fecha/hora  N primeros y N últimos:  Fecha y hora de inicio  Coordenadas iniciales  Fecha y hora final  Coordenadas finales  Distancia recorrida (millas)  Costo total pagado |
| **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(...)*** |

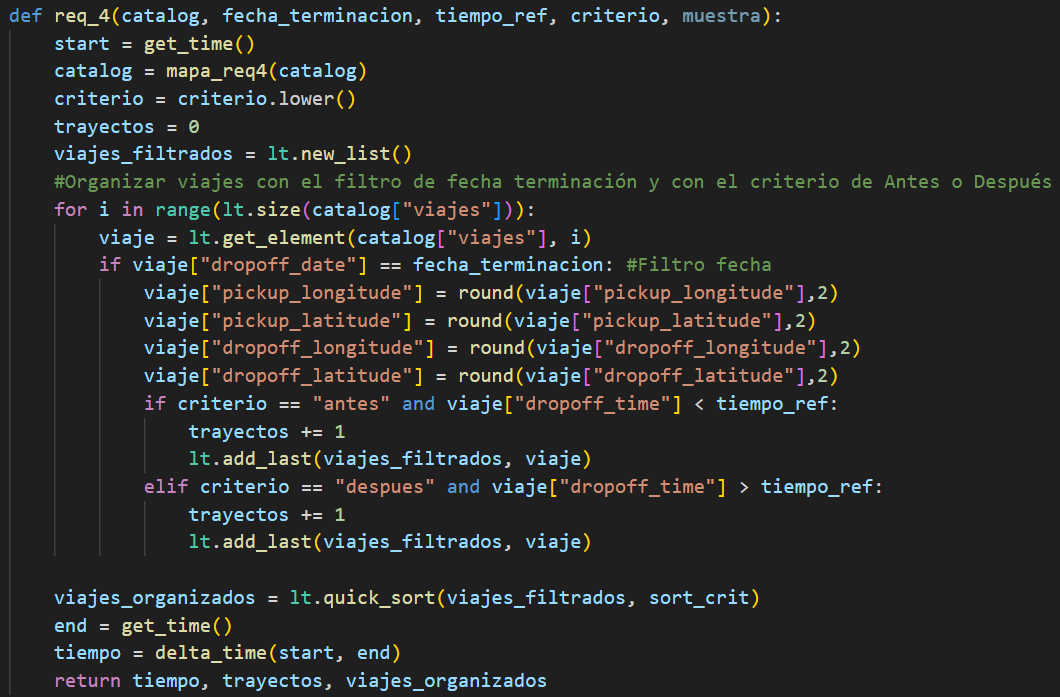
## **Análisis**

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

# **Requerimiento 4**

## **Descripción**

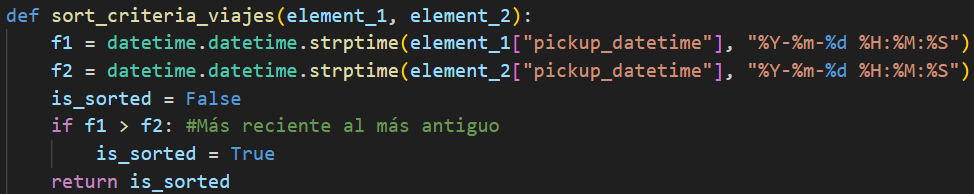
Función req\_4



Función creación de la Tabla Hash para el req\_4

Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.

Función sort\_crit para la organización en el quick\_sort del req\_4

A partir de una fecha de terminación de trayecto dada debo filtrar todos los viajes que hayan terminado en esa fecha y mostrar todos los trayectos antes o después (criterio dado por el usuario) de una hora de terminación dada. Deben mostrarse los primeros n trayectos y últimos n trayectos previamente ordenados del más reciente al más antiguo.

La función mapa\_req4 crea la tabla de hash a utilizar para el cumplimiento del req\_4. La llaves son las fechas de terminación de los trayectos, y su valor una array\_list con todos los viajes que terminaron en esa fecha.

La función de sort criterio viajes compara fecha y hora de recogida para organizar del más antiguo al más reciente al momento de utilizar quick\_sort.

|  |  |
| --- | --- |
| **Entrada** | Fecha de terminación  Criterio de comparación (ANTES, DESPUES)  Tiempo de terminación  Tamaño de muestra |
| **Salidas** | Respuesta Tiempo de ejecución  # de trayectos dentro del rango de fecha/hora  N primeros y N últimos:  Fecha y hora de inicio  Coordenadas iniciales  Fecha y hora final  Coordenadas finales  Distancia recorrida (millas)  Costo total pagado |
| **Implementado (Sí/No)** | Sí, Juliana Rodríguez |

## **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(...)*** |

## **Análisis**

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

# **Requerimiento 5**

## **Descripción**

Breve descripción de como abordaron la implementación del requerimiento

|  |  |
| --- | --- |
| **Entrada** | Parámetros necesarios para resolver el requerimiento. |
| **Salidas** | Tiempo de ejecución  # de trayectos dentro del rango de fecha/hora  N primeros y N últimos:  Fecha y hora de inicio  Coordenadas iniciales  Fecha y hora final  Coordenadas finales  Distancia recorrida (millas)  Costo total pagado |
| **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(...)*** |

## **Análisis**

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

# **Requerimiento 6**

## **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(...)*** |

## **Análisis**

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