|  |  |
| --- | --- |
| Proyecto 1 | **Requerimientos Funcionales** |
| Autores: | Jaime Carvajal Melendez  Juan Pablo Romero Pinzón |
| Fecha: | 17/02/2020 |

**Listado de Requerimientos**

|  |  |
| --- | --- |
| **Nombre** | R1 – Cargar información |
| **Resumen** | Solamente una vez se puede cargar la información de los comparendos sin requerir una entrada. La complejidad temporal esperada es de n, donde n es la cantidad de comparendos, ya que todo lo que se quiere utilizar se puede inicializar para cada comparendo al mismo tiempo. |
| **Entradas** | |
| Ninguna | |
| **Resultados** | |
| |  | | --- | | Se inicializa una pila con los comparendos | | Se inicializa una cola con los comparendos | | Se muestra el número de comparendos | | Se muestra el comparendo con el mayor objectID | | Se inicializa y se muestra las esquinas del rectángulo minimax | | |

|  |  |
| --- | --- |
| **Nombre** | R2 – Consultar primer comparendo con una localidad dada |
| **Resumen** | Se recibe un string localidad, se compara con las localidades de cada comparendo, y se devuelve el primer comparendo con la localidad dada que se encuentre. Se maneja también el caso de que no se encuentre una localidad. Complejidad esperada de n, donde n es la cantidad de comparendos, ya que solo se debería tener que recorrer una estructura una vez como máximo |
| **Entradas** | |
| La localidad que se quiere buscar. | |
| **Resultados** | |
| |  | | --- | | Si existen comparendos con la localidad pedida, se devuelve el primero de estos | | Si no existen comparendos con la localidad pedida, se deja saber al usuario. | | |

|  |  |
| --- | --- |
| **Nombre** | R3 – Consultar todos los comparendos con una fecha dada |
| **Resumen** | El usuario entrega una fecha, para la cual se buscan todos los comparendos que tengan la fecha dada y se le devuelven al usuario en orden por el código de infracción. Se muestra la información de los resultados. Se espera una complejidad de n, donde n es la cantidad de comparendos, ya que se pueden ordenar los comparendos mientras se van escogiendo. |
| **Entradas** | |
| Fecha | |
| **Resultados** | |
| |  | | --- | | Se muestra una lista con los resultados. | | |

|  |  |
| --- | --- |
| **Nombre** | R4 – Comparar comparendos por código INFRACCIÓN en dos fechas dadas. |
| **Resumen** | Se reciben dos strings fecha en el formato año/mes/día, se comparan todas las infracciones para esas fechas y se devuelven los números de cada tipo de infracción. Complejidad esperada de n, donde n es la cantidad de comparendos, ya que todo lo que se pide se puede actualizar para cada comparendo pasando por él solamente una vez. |
| **Entradas** | |
| |  | | --- | | Fecha1 | | Fecha2 | | |
| **Resultados** | |
| |  | | --- | | Se devuelve una tabla comparando la cantidad de cada tipo de infracción para cada fecha. | | |

|  |  |
| --- | --- |
| **Nombre** | R5 – Consultar primer comparendo con una infracción dada |
| **Resumen** | Se recibe un string infracción, se compara con las infracciones de cada comparendo, y se devuelve el primer comparendo con la infracción dada que se encuentre. Se maneja también el caso de que no se encuentre una infracción. Complejidad esperada de n, donde n es la cantidad de comparendos, ya que solo se debería tener que recorrer una estructura una vez como máximo. |
| **Entradas** | |
| La infracción que se quiere buscar. | |
| **Resultados** | |
| |  | | --- | | Si existen comparendos con la infracción pedida, se devuelve el primero de estos | | Si no existen comparendos con la infracción pedida, se deja saber al usuario. | | |

|  |  |
| --- | --- |
| **Nombre** | R6 – Consultar los comparendos con una infracción dada |
| **Resumen** | Se recibe un string infracción, se compara con las infracciones de cada comparendo, y se devuelven los comparendos con la infracción dada que se encuentre. Se maneja también el caso de que no se encuentre una infracción. Complejidad esperada de n+n\*lg n, donde n es la cantidad de comparendos, ya que se debería tener que recorrer una estructura una vez como máximo para la lectura y luego n\*lg n en organizarlo usando mergesort. |
| **Entradas** | |
| La infracción que se quiere buscar. | |
| **Resultados** | |
| |  | | --- | | Si existen comparendos con la infracción pedida, devuelve todos los que cumplan esa condición organizados por fecha de menor a mayor. | | Si no existen comparendos con la infracción pedida, se deja saber al usuario. | | |

|  |  |
| --- | --- |
| **Nombre** | R7 – Comparar comparendos por tipo servi “Particular” y “Público” |
| **Resumen** | Se genera una tabla en la que para cada infracción que presenta al menos uno de los dos tipos de servicio y la cantidad de servicios encontrados para esta infracción. La tabla debe estar organizada en orden alfabético según la infracción. Complejidad esperada n^2 + n\*lg n, porque se debe recorrer cada infracción dos veces, para leerlo y para saber si ya se había leído y sumar su tipo de servicio; además, hay que organizarlo usando mergesort. Seguir el siguiente ejemplo:  Comparación de comparendos por Infracción en servicio Particular y servicio Público  Infracción | Particular | Público  C19 | 2 | 12  C38 | 69 | 0  H02 | 0 | 17 |
| **Entradas** | |
| Ninguna. | |
| **Resultados** | |
| |  | | --- | | Tabla en la que se muestra el número de tipos de servicio “Particular” y “Público” para cada infracción. | | |

|  |  |
| --- | --- |
| **Nombre** | R8 – Consultar los comparendos con una localidad dada y bajo cierto rango de fechas. |
| **Resumen** | Se recibe un string localidad, y dos de fecha. Se cuenta el número de comparendos encontrados para cada infracción donde su comparendo sea de la localidad entregada y entre las fechas dadas. Se debe entregar en orden alfabético según la infracción. Complejidad esperada n + n\*lg n porque hay que recorrer todos los comparendos para identificar cuales requerimos extraer y luego organizarlos usando mergesort. Seguir el siguiente ejemplo:  Comparación de comparendos en Teusaquillo del 2018/06/01 al 2018/06/30  Infracción | # Comparendos  C19 | 14  C38 | 69  H02 | 17 |
| **Entradas** | |
| Localidad, y dos fechas para filtrar los comparendos a escoger. | |
| **Resultados** | |
| Tabla organizada alfabéticamente por infracción donde se da el número de veces que se encontró cada infracción donde su comparendo coincidía con las condiciones dadas por parámetro. | |

|  |  |
| --- | --- |
| **Nombre** | R9 – Consultar los n comparendos con mayor número de infracciones y bajo cierto rango de fechas. |
| **Resumen** | Se recibe un int n, y dos de fecha. Se encuentran las n infracciones que más se repiten y que estén entre las fechas dadas. Se debe entregar en orden de mayor número de comparendos encontrados a menor. Complejidad esperada n + n\*lg n porque hay que recorrer todos los comparendos para identificar cuales requerimos extraer y luego organizarlos usando mergesort. Seguir el siguiente ejemplo:  Ranking de las 3 mayores infracciones del 2018/06/01 al 2018/06/30  Infracción | # Comparendos  C38 | 69  H02 | 17  C19 | 14 |
| **Entradas** | |
| Número de comparendos a mostrar, y dos fechas para filtrar los comparendos a escoger. | |
| **Resultados** | |
| Tabla organizada por número de comparendos encontrados donde se da el número de veces que se encontró cada infracción donde su comparendo coincidía con las condiciones dadas por parámetro. | |

|  |  |
| --- | --- |
| **Nombre** | R10 –Generar histograma con número total de comparendos por localidad. |
| **Resumen** | Cada comparendo se representa con un asterisco, los nombres de las localidades aparecen alfabéticamente y se justifican cada 16 caracteres. Cada asterisco implica mínimo un comparendo y máximo 50. Tiene complejidad esperada de n+n\*log(n), para poder leer todos los datos y luego ordenarlos. |
| **Entradas** | |
|  | |
| **Resultados** | |
| |  | | --- | | Se muestra el número de comparendos por localidad en un histograma. | | Si no hay comparendos en la localidad, se le hace saber al usuario. | | |