Lista de Requerimientos

|  |
| --- |
| 1A- Obtener el ranking de las N franjas horarias que tengan más infracciones. El valor N es un dato de entrada. |
| Entradas |
| 00:00:00 - 00:59:59 01:00:00 - 01:59:59 02:00:00 - 02:59:59 ... 21:00:00 - 21:59:59 22:00:00 - 22:59:59 23:00:00 - 23:59:59 |
| Caso Especial: 24:00:00 equivale a 00:00:00 |
| Resultados |
| Ranking de las N franjas horarias que tengan más infracciones |

|  |
| --- |
| 2A- Realizar el ordenamiento de las infracciones por Localización Geográfica (Xcoord, Ycoord). |
| Entradas |
| La llave de búsqueda es la tupla (Xcoord, Ycoord) |
| Resultados |
| El ordenamiento de las infracciones en una Tabla de Hash por localización geográfica. |

|  |
| --- |
| 3A- Buscar las infracciones por rango de fechas [Fecha Inicial (Año/Mes/Día), Fecha Final (Año/Mes/Día)]. |
| Entradas |
| Rango de fechas [Fecha Inicial (Año/Mes/Día), Fecha Final (Año/Mes/Día)]. |
| Resultados |
| Un Árbol Balanceado por Fecha (TicketIssueDate). |

|  |
| --- |
| 1B- Obtener el ranking de las N tipos de infracción (ViolationCode) que tengan más infracciones. El valor N es un dato de entrada. |
| Entradas |
| N tipos de infracción (ViolationCode) que tengan más infracciones. |
| Resultados |
| En el ranking con los tipos de infracción de mayor a menor número de infracciones. |

|  |
| --- |
| 2B- Realizar el ordenamiento de las infracciones por Localización Geográfica (Xcoord, Ycoord). |
| Entradas |
| Tupla (Xcoord, Ycoord) |
| Resultados |
| Realiza el ordenamiento de las infracciones en un Árbol Balanceado por localización geográfica. |

|  |
| --- |
| 3B- Buscar las franjas de fecha-hora donde se tiene un valor acumulado de infracciones en un rango dado [US$ valor inicial, US$ valor final]. |
| Entradas |
| [US$ valor inicial, US$ valor final] |
| Resultados |
| Para cada fecha en el semestre seleccionado, se calcula el valor acumulado (US$) de las infracciones para cada franja de hora completa. |

|  |
| --- |
| 1C- Obtener la información de una localización dada |
| Entradas |
| Localización dada (AddressID) |
| Resultados |
| El total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, el valor total a pagar por las infracciones, y el Id de la calle (StreetSegId). |

|  |
| --- |
| 2C- Obtener las infracciones en un rango de horas [HH:MM:SS inicial, HH:MM:SS final]. |
| Entradas |
| Rango de horas [HH:MM:SS inicial, HH:MM:SS final]. |
| Resultados |
| Información general: el total de infracciones, el porcentaje de infracciones sin accidente, el porcentaje de infracciones con accidente, y el valor total a pagar por las infracciones. Información por código (ViolationCode): Para las infracciones resultantes son agruparlas por código. Informar el detalle de código y el total de infracciones. |

|  |
| --- |
| 3C – Obtener el ranking de las N localizaciones geográficas (Xcoord, Ycoord) con la mayor cantidad de infracciones. |
| Entradas |
| Las N localizaciones geográficas. |
| Resultados |
| El ranking de las N localizaciones geográficas (Xcoord, Ycoord) con la mayor cantidad de infracciones. |

|  |
| --- |
| 4C – Mostrar una gráfica ASCII con la información de las infracciones por código (ViolationCode). |
| Entradas |
| La información de las infracciones por código (ViolationCode). |
| Resultados |
| Una gráfica ASCII con la información de las infracciones por código (ViolationCode). |

|  |
| --- |
| Restricciones |
| • Los datos contenidos en los archivos sólo se pueden leer una vez |
| • Se deberá trabajar en Java 8 |
| • El proyecto se debe implementar en Eclipse |
| • La entrada/salida de información adicionales se debe realizar por consola |
| • No usar las colecciones del API Java. |