**Diseño de proyecto**

**Carga de datos**

* Implementación: Utilizar lectura de archivos Json para cada comparendo y almacenar la información recibida en objetos Comparendo (objeto referente a un comparente) que se irán añadiendo a una cola de prioridad, una tabla hash. A medida que se van creando los comparendos se van comparando para encontrar aquel con el mayor OBJECT\_ID mediante una variable y se usa un contador para saber el número de comparendos.
* Parámetros de entrada: ruta del archivo Json a leer
* Retorno: Comparendo can mayor OBJECT\_ID
* Orden del algoritmo: N

**Parte A**

1A.

* Implementación: Utilizar la cola de prioridad en que se cargaron los comparendos, configurada para que la prioridad esté dada por la gravedad de la infracción. De esta cola se elimina M veces y así se obtienen los comparendos de mayor gravedad en orden descendente.
* Parámetros de entrada: Entero M del número de comparendos a retornar
* Retorno: Lista Encadenada con los M comparendos de mayor prioridad
* Orden del algoritmo: M

2A.

* Implementación: Se utiliza la tabla de hash de encadenamiento separado en la cual las llaves serán el mes y el día, y el valor será el conjunto de comparendos registrados para ese mes y ese día. Así solo se debe utilizar el método get de la tabla.
* Parámetros de entrada: String del mes y el día de la semana a buscar
* Retorno: Lista Encadenada con los comparendos correspondientes
* Orden del algoritmo: N²

3A.

* Implementación: Convertir la fecha mínima en mes y día y extraer de la tabla hash ya utilizada los comparendos con esa llave. De ahí buscar N comparendos con la localidad ingresada. Si los comparendos en esa llave no alcanza a completar N, se aumenta el día y se sigue buscando.
* Parámetros de entrada: Fechas máximo y mínimo, String de la localidad y entero del N a buscar
* Retorno: Lista encadenada con los comparendos
* Orden del algoritmo: N

**Parte C**

1C.

* Implementación: Se copian los comparendos a un arreglo y se ejecuta un heapsort por fecha. Posteriormente se cuentan el numero de comparendos por intervalo.
* Parámetros de entrada: Entero del intervalo a buscar
* Retorno: Lista encadenada con la información del histograma
* Orden del algoritmo: N²logN

2C.

* Implementación: Se utiliza una tabla hash donde las llaves son el mes y el día del mes, y el valor es una cola con el conjunto de comparendos para ese mes y día. Así se hace un conteo de comparendos por día y se puede saber cuándo pueden ser procesados y cuantos quedan en espera para el próximo día y así obtener los datos.
* Parámetros de entrada: Ninguno
* Retorno: Arreglo de listas encadenadas con los datos para el histograma y la tabla
* Orden del algoritmo: N²logN

3C.

* Implementación: Se realiza el mismo procedimiento que en requerimiento anterior pero se usa una cola de prioridad para los comparendos que quedan en espera donde la prioridad es el mayor valor del tipo de infracción.
* Parámetros de entrada: Ninguno
* Retorno: Arreglo de listas encadenadas con los datos para el histograma y la tabla
* Orden del algoritmo: N²logN