**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 una cola de prioridad donde queden organizados por fecha. Posteriormente se cuentan cuantos comparendos hay por cada intervalo y se genera el histograma
* Parámetros de entrada: Ninguno
* Retorno: Cola de prioridad de los comparendos ordenas por fecha
* Orden del algoritmo: N²logN

2C.

* Implementación: Se utiliza el mismo método del punto 1C pero esta vez los comparendos de la cola se van contando por día y si el número sobrepasa los 1500 se cambia el conteo.
* Parámetros de entrada: Ninguno
* Retorno: Arreglo de Strings con la información del histograma
* Orden del algoritmo: N²logN

3C.

* Implementación:
* Parámetros de entrada:
* Retorno:
* Orden del algoritmo: