**Análisis de Resultados**

**Reto 3 Grupo 4**

**Integrantes**

Ashlee Yin Romero, a.yin@uniandes.edu.co 202421132

Daniel Galindo, d.galindot@uniandes.edu.co, 202414673

Tomás Lozano, t.lozanoc@uniandes.edu.co 202422837

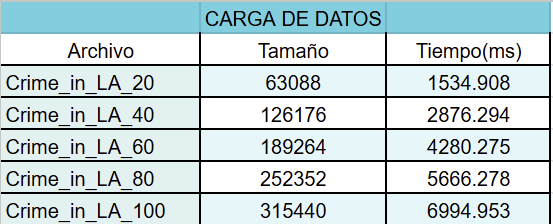
**Especificaciones**

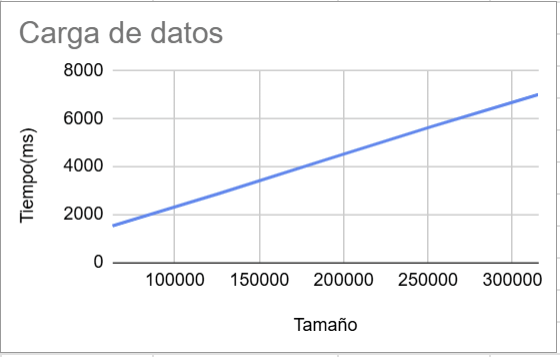
La medición del tiempo de las funciones fueron tomadas en un computador MacBook Air 2015, viejita pero bien cuidada, con un procesador 1,6 GHz Dual-Core Intel Core i5 y memoria de 8 GB 1600 MHz DDR3. Todos los tiempos fueron tomados en milisegundos.

### **Carga de Datos**

**Descripción:** Cargar y almacenar los datos en estructuras eficientes.  
**Complejidad:** O(n log n) (por uso de RBT y ordenamientos).

**Análisis:**Durante la carga, los datos son insertados en listas, mapas hash y árboles RBT para facilitar búsquedas eficientes. Las fechas se convierten a objetos datetime, y se organizan en estructuras categorizadas por fecha de ocurrido, fecha de reporte, edad, y área. Esta organización permite el uso posterior de búsquedas por rango con complejidades controladas. Aunque la inserción en listas y mapas tiene costo lineal, el ordenamiento aplicado posteriormente implica un costo dominante de O(n log n).

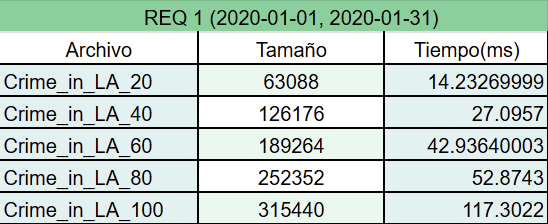


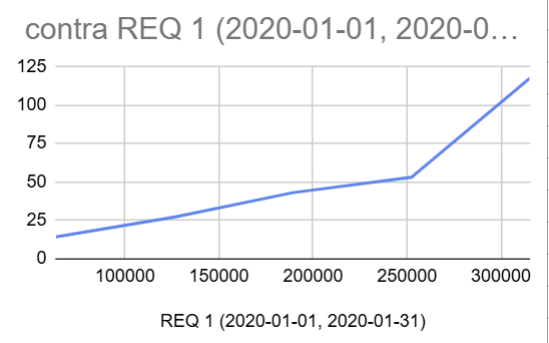


### **Requerimiento 1 (Grupal)**

**Descripción:** Listar los crímenes ocurridos entre dos fechas.  
**Complejidad:** O(n) (recorrido lineal de los nodos del árbol en el rango)

**Análisis:**El requerimiento se ejecuta en tiempo lineal respecto al número de registros recorridos, ya que se hace un recorrido por el árbol binario balanceado (RBT) entre las fechas dadas. El uso de estructuras eficientes permite mantener un buen rendimiento.

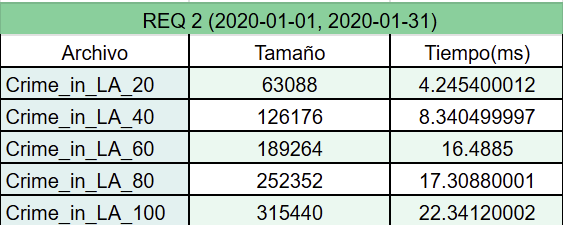


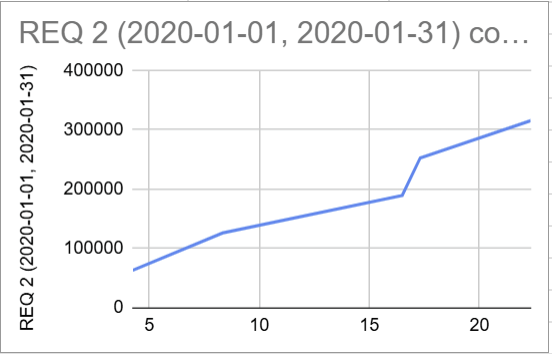


### **Requerimiento 2 (Grupal)**

**Descripción:** Listar los crímenes graves resueltos reportados en un rango de fechas.  
**Complejidad:** O(n log n) (filtrado + ordenamiento por fecha y área)

**Análisis:**Aunque se recorre el árbol como en el req 1, se aplica un ordenamiento adicional descendente por fecha y suborden por área, lo que agrega una complejidad logarítmica adicional. Se aplica slicing para mostrar los 5 primeros y 5 últimos.



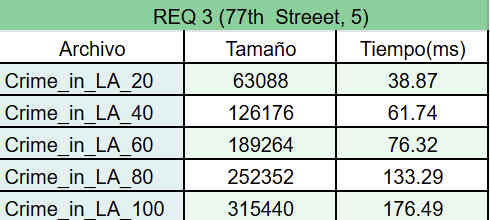


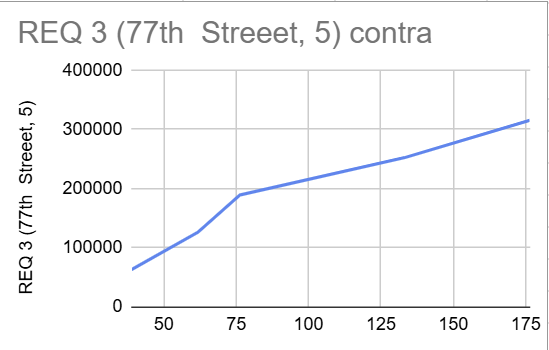
### 

### **Requerimiento 3 (Individual)**

**Descripción:** Consultar los N crímenes reportados más recientemente para un área.  
**Complejidad:** O(n log n) (heap de máximos por fecha de reporte)

**Análisis:**Se filtran los registros por área desde un mapa hash y se construye un heap con clave inversa para obtener los N crímenes más recientes. El ordenamiento es eficiente para este caso.

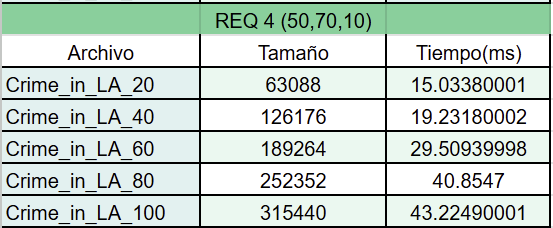


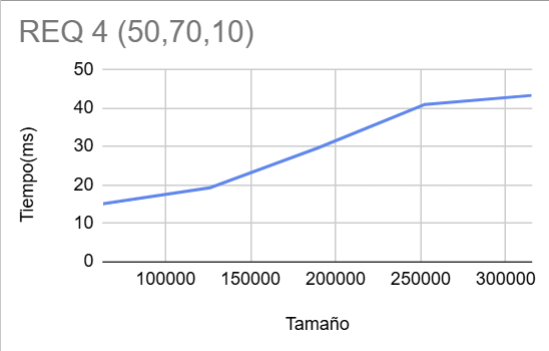


### **Requerimiento 4 (Individual)**

**Descripción:** Consultar los N crímenes con víctimas en un rango de edad y clasificarlos por gravedad.  
**Complejidad:** O(n log n)

**Análisis:**Se hace una búsqueda por rango de edad en el árbol por edades. Luego, se clasifican por gravedad y se ordenan por edad y fecha. El procedimiento cumple con las recomendaciones del enunciado.

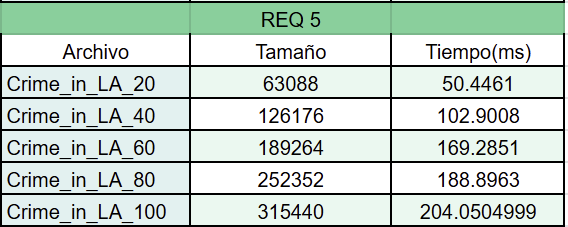


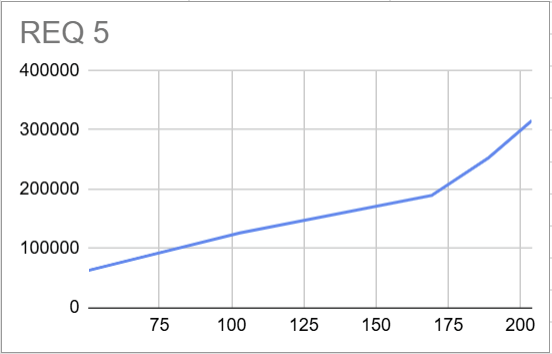


### **Requerimiento 5 (Individual)**

**Descripción:** Consultar las N áreas con mayor cantidad de crímenes no resueltos en un rango de fechas.  
**Complejidad:** O(n log n) (agrupación + heap por cantidad)

**Análisis:**Se recorren los crímenes por fecha y se agrupan por área con un contador. Luego, se aplica un heap para determinar las áreas con mayor número de casos.



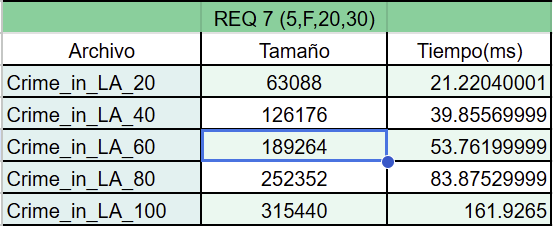


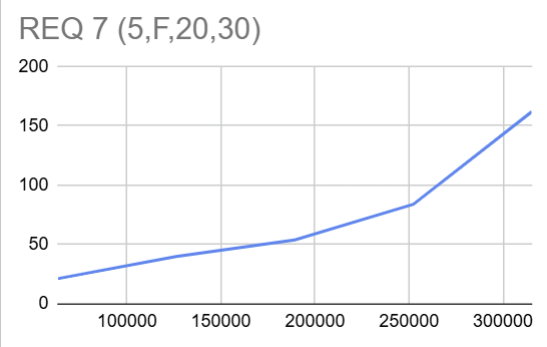
**Requerimiento 6**

### **Requerimiento 7 (Grupal)**

**Descripción:** Determinar los N crímenes más comunes para víctimas de un sexo en un rango de edad.  
**Complejidad:** O(n log n) (filtrado + conteo + ordenamiento)

**Análisis:**Se realiza una búsqueda en el árbol por edad, se filtran por sexo y se agrupan los delitos por código. Luego se cuentan por edad y por año para cada crimen. Los datos se presentan como lo exige el requerimiento.





**Requerimiento 8 (BONO)**