

# Laboratorio 2: Sequential File vs AVL File

Integrantes:

- Mikel Bracamonte
- Gino Daza

**Github:** <https://github.com/GinoDazaU/base-de-datos-2-labs/tree/main/lab2>

## 1. Preparación de Datos:

- Se utilizó el archivo sales\_dataset.csv como fuente común para ambas pruebas.
- Los datos fueron previamente desordenados para evitar sesgos, ya que:
  - El archivo original estaba ordenado por ID.
  - Algunas estructuras de datos (como AVL) pueden mostrar peor rendimiento con entradas ordenadas.
- Ambos sistemas procesan exactamente los mismos datos (en idéntico orden aleatorio), esto garantiza una comparación justa de rendimiento.

## 2. Implementación de Pruebas:

Para la comparación, se utilizó un nuevo script en Python llamado p1\_p2\_testing.py, en el cual se realizaron las pruebas con los mismos datos para ambos métodos.

## 3. Metodología de Pruebas:

Para comparar el rendimiento de ambas estructuras, se ejecutaron pruebas idénticas bajo las mismas condiciones:

100,000 búsquedas aleatorias de IDs existentes,

100,000 búsquedas por rangos de tamaño variable (5-50 IDs), y

500 eliminaciones aleatorias,

En la misma laptop.

Todos los tiempos se midieron con `time.perf_counter()`, y se calculó el promedio para estimar cuánto demoró cada operación por sí sola.

## 4. Resultados:

### === Sequential File ===

-> Inserción

Tiempo total: 16.7564s | Avg: 0.167232ms

-> Búsqueda

Tiempo total: 6.1581s | Avg: 0.061416ms

-> Búsqueda por rango

Tiempo total: 14.3646s | Avg: 0.143392ms

-> Eliminación

Tiempo total: 1.7460s | Avg: 3.491209ms

### === AVL File ===

-> Inserción

Tiempo total: 11.9065s | Avg: 0.118811ms

-> Búsqueda

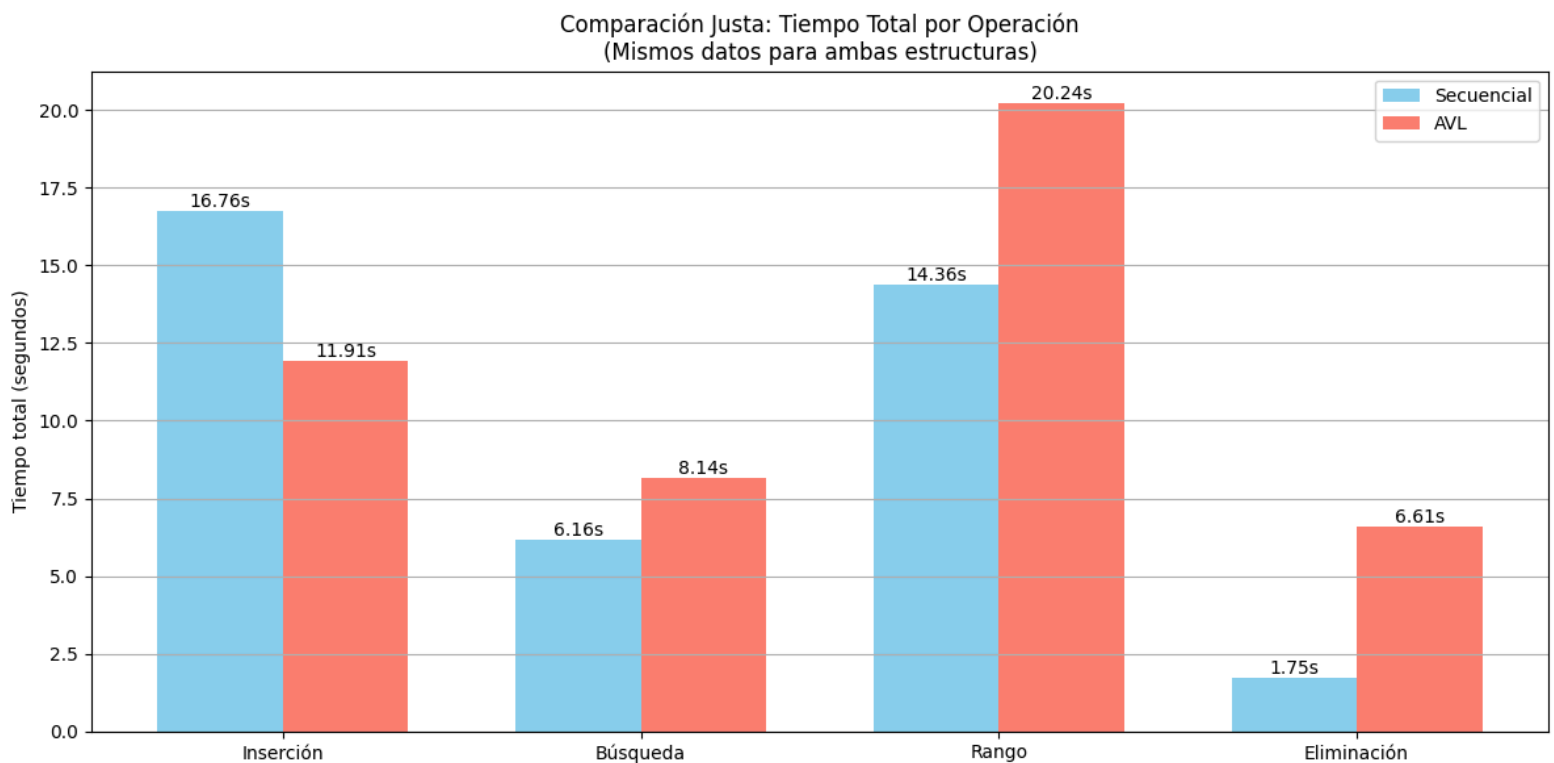
Tiempo total: 8.1406s | Avg: 0.081232ms

-> Búsqueda por rango

Tiempo total: 20.2361s | Avg: 0.202119ms

-> Eliminación

Tiempo total: 6.6120s | Avg: 13.222114ms



## 5. Análisis:

La gráfica presentada muestra el tiempo total requerido por ambas estructuras (archivo secuencial y árbol AVL) para realizar las cuatro operaciones principales: inserción, búsqueda, búsqueda por rango y eliminación, utilizando los mismos datos para asegurar una comparación justa.

### Inserción:

El archivo AVL presenta un mejor rendimiento en la etapa de inserción, con un tiempo total de 11.91 segundos, frente a los 16.76 segundos del archivo secuencial. Esto indica que, a pesar de la complejidad estructural del árbol, su capacidad para mantener orden optimiza el proceso de inserción a gran escala.

### Búsqueda específica

En este caso, el árbol AVL (8.14s) es ligeramente más lento que el archivo secuencial (6.16s). Aunque ambos realizan búsquedas logarítmicas, el AVL incurre en más accesos a disco, ya que cada comparación puede requerir la lectura de un nodo completo (para obtener sus dos punteros) desde el archivo. Esto incrementa el tiempo total, a pesar de la eficiencia teórica del algoritmo.

### Búsqueda por rango

La búsqueda por rango en el AVL (20.24s) es menos eficiente que en el archivo secuencial (14.36s). Esto se debe a que recorrer un rango en un árbol implica múltiples llamadas recursivas y accesos dispersos al archivo, en comparación con el recorrido lineal contiguo que permite el archivo secuencial en su zona principal.

### Eliminación

La operación de eliminación en el AVL (6.61s) es significativamente más costosa que en el archivo secuencial (1.75s). Esto se debe a que eliminar un nodo en un árbol AVL requiere primero ubicarlo, luego posiblemente reemplazarlo (si tiene dos hijos), y finalmente reequilibrar el árbol mediante rotaciones. Cada uno de estos pasos implica múltiples lecturas y escrituras en disco.

## 6. Discusión:

Los resultados obtenidos muestran que no existe una estructura claramente superior en todos los casos, sino que el rendimiento depende del tipo de operación predominante. El árbol AVL, al mantener un orden balanceado, facilita inserciones más rápidas a gran escala, lo cual es beneficioso en sistemas donde los registros se agregan constantemente. Sin embargo, el costo de mantener este equilibrio se refleja en

operaciones como la eliminación y la búsqueda por rango, donde el archivo secuencial, por su estructura lineal y sin reequilibrios, resulta más eficiente.

Por lo tanto, la elección entre un archivo secuencial y un árbol AVL debe hacerse considerando los requerimientos específicos del sistema: si se prioriza la velocidad de inserción y la organización automática, el árbol AVL es adecuado; si en cambio se requiere rapidez en búsquedas o eliminaciones masivas, el archivo secuencial puede ser la mejor opción.