



# Tecnológico de Monterrey

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

**Programación de estructuras de datos y algoritmos fundamentales**

**Profesor:**

Dan Perez

**“Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales -  
Reflexión”**

**Estudiante:**

Luis Fernando Cruz Flores

A01738347

**09 de septiembre del 2024**

## Investigación:

El uso de algoritmos de ordenamiento y búsqueda es fundamental en programas que deben manejar grandes líneas de datos, en este caso las bitácoras o logs, los cuales cuentan con más de 1000 líneas de texto, cada una conteniendo información como fecha, hora, dirección IP y puerto. Por lo que el manejo eficiente de la información es de suma importancia para el desarrollo de nuestro programa.

Existen diversos métodos de ordenamiento y de búsqueda de datos, cada uno mejor o peor optimizado según sea su utilidad. Por lo que la correcta elección de estos algoritmos es de suma importancia, no solo por afectar al tiempo de procesamiento del programa, también el uso eficiente de la memoria.

A continuación algunos de los algoritmos de ordenamiento más utilizados junto a su complejidad:

- Bubble sort  $O(n^2)$
- Insertion sort  $O(n^2)$
- Merge sort  $O(n \log n)$
- Selection sort  $O(n^2)$
- Quick sort  $O(n \log n)$

Tras analizar los distintos tipos de algoritmos de ordenamiento, podemos destacar que los dos más eficientes son tanto Mergesort como Quicksort, ya que estos cuentan con una complejidad  $O(n \log n)$ . Debido a esto, cualquiera de estos dos algoritmos serían apropiados para un programa de consultas de bitácoras o logs. En nuestro caso, debido a que ya habíamos trabajado con el método de merge sort, optamos por usar uno nuevo (Quicksort), de tal manera que podemos aprovechar y aprender sobre este nuevo método.

De la misma forma, los algoritmos de búsqueda tienen distintas ventajas y desventajas cuando se comparan entre sí, principalmente debido a su complejidad, los dos algoritmos principales son: Búsqueda lineal  $O(n)$  y Búsqueda binaria  $(\log n)$ , el último mencionado tiene la mejor eficiencia debido a su naturaleza de ir dividiendo el arreglo o lista en 2 dependiendo si el número es mayor o menor.

## Reflexión:

Tras investigar sobre la importancia de los algoritmos a la hora de desarrollar herramientas o programas computacionales, pude notar que unos simples cambios en la elección de algoritmos puede tener consecuencias gigantescas, por ejemplo, si se trabajara con bitácoras de plataformas gigantescas como google o microsoft, entonces la complejidad haría que un trabajo que se podría hacer en cuestión de minutos, se tenga que hacer en horas o incluso días debido a la diferencia de operaciones que el programa tendría que

realizar. Otro ejemplo sería el programa de fibonacci con recursión, el cual, al usar valores que se aproximan a 100, el número de operaciones aumenta significativamente, es por eso que la elección y correcta implementación de los algoritmos de búsqueda y ordenamiento se vuelven de suma importancia, no solo para esta práctica, sino también para el desarrollo de cualquier programa que tenga que trabajar con grandes volúmenes de datos.