



# Tecnológico de Monterrey

Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos  
Fundamentales (Evidencia Competencia)

Nikole Morales - A01782557

**TECNOLÓGICO DE MONTERREY**

Campus Santa Fe

Grupo: 570

Fecha de entrega: 8 de julio de 2023

Profesor: Dr. Eduardo A. Rodríguez Tello

Programación de estructuras de datos y algoritmos fundamentales

# Conceptos Básicos y Algoritmos Fundamentales

## Reflexión

Los algoritmos de ordenamiento y búsqueda tienen un rol muy importante para poder desarrollar un sistema ya que como su nombre nos indica permiten ordenar la información basándonos en un criterio. Cuando se necesita organizar grandes volúmenes de información podemos usar diferentes tipos de ordenamientos, cada uno tiene distinta eficacia y dependiendo el caso y la información hay unos que tienen mayor eficiencia ya que según el algoritmo elegido se realizan  $n$  operaciones y el tiempo de ejecución depende de dichas operaciones, es decir depende de cuantas se repita el código.

Dentro del ámbito informático se tienen que tomar en cuenta algunas limitaciones que como ya se mencionó anteriormente una de esas limitaciones podría ser el tiempo de ejecución, tomando en cuenta que en esta actividad se utilizó una cantidad de datos muy grande por lo que se tuvo que considerar cuál fue el método más rápido.

Para el programa que se realizó en esta actividad se usaron dos algoritmos de ordenamiento, bubble sort y mergesort, con el fin de comparar el rendimiento de cada uno. Los datos obtenidos por el programa fueron los siguientes:

<b>Bubble Sort</b>	El número de comparaciones en el ordenamiento Burbuja fue de:	141229221
	El número de intercambios en el ordenamiento Burbuja fue de:	7016691

Tabla 1.1: Resultado de las comparaciones e intercambios del algoritmo Bubble Sort.

<b>Merge Sort</b>	El número de comparaciones en el ordenamiento Mergesort fue de:	214693
	Merge sort no tiene intercambios, por lo tanto, el número de intercambios en el ordenamiento fue de:	0

Tabla 1.2: Resultado de las comparaciones e intercambios del algoritmo Merge Sort.

Como se puede observar en este caso en específico el método de ordenamiento más rápido fue el de merge sort ya que realizó menos comparaciones para poder ordenar la información.

También se puede determinar esto con la complejidad temporal, el ordenamiento de burbuja tiene una complejidad  $O(n^2)$ , esto significa que el tiempo de ejecución crece cuadráticamente con el volumen de datos o información por ende en casos donde se manejan grandes cantidades de datos el tiempo de ejecución es muy largo.

Mientras que la complejidad temporal del MergeSort es  $O(n \log n)$ , esto significa que el tiempo de ejecución aumenta a medida que el tamaño del conjunto de datos crece. Por esta razón cuando se trata de cantidades de datos grandes, este método se vuelve uno de los más adecuados.

Por otro lado en cuanto al algoritmo de búsqueda se utilizó el de búsqueda binaria, su funcionamiento se basa en dividir de manera repetida la sección de la lista en la cual se presume que el elemento puede estar presente, hasta que se reduce el rango de ubicaciones posibles a solo una.

La búsqueda binaria destaca porque necesita menos tiempo para obtener una respuesta, se aplica cuando se usan cantidades grandes de datos, la reducción del número de comparaciones necesarias y su requerimiento de una lista ordenada.

Estas características hacen que la búsqueda binaria sea una elección importante y eficiente en situaciones donde se necesita encontrar elementos ya ordenados, en este caso arrojó los resultados de la búsqueda dentro de un rango de fechas y horas dadas por el usuario.

En conclusión el algoritmo tanto de búsqueda como de ordenamiento más eficaz depende de la cantidad de datos que se manejan, cabe mencionar que cada algoritmo tiene sus ventajas y desventajas que pueden variar dependiendo del caso y las características de la problemática, por eso es importante hacer una comparación entre métodos y encontrar el más conveniente para el caso.

## Referencias

Tait, J., Ripke, T., Roger, L., & Matsuo, T. (2018, December). Comparing Python and C++ efficiency through sorting. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 864-871). IEEE. Recuperado el 7 de julio de 2023 en: <https://ieeexplore.ieee.org/abstract/document/8947654>

Arifin, R. W., & Setiyadi, D. (2020). Algoritma Metode Pengurutan Bubble Sort dan Quick Sort Dalam Bahasa Pemrograman C++. *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS: Journal of Information System*, 4(2), 178-187. Recuperado el 7 de julio de 2023 en: <http://www.ejournal-binainsani.ac.id/index.php/ISBI/article/view/1348>

C++ program for merge sort. (2022, diciembre 15). GeeksforGeeks. Recuperado el 7 de julio de 2023 en: <https://www.geeksforgeeks.org/cpp-program-for-merge-sort/>

Bubble sort - data structure and algorithm tutorials. (2014, febrero 2). GeeksforGeeks.  
Recuperado el 7 de julio de 2023 en: <https://www.geeksforgeeks.org/bubble-sort/>