

Instituto Tecnológico de Estudios Superiores de Monterrey



**Tecnológico
de Monterrey**

Campus Monterrey

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

Programación de estructuras de datos y algoritmos fundamentales (Gpo 610)

Eduardo López Benítez

Fidel Morales Briones A01198630

Monterrey, Nuevo León, a 7 de septiembre 2023

Reflexión personal:

La importancia de los algoritmos de búsqueda y ordenamiento en la programación para ordenar datos es muy alta, ya que sin ellos no sería posible manejar y procesar grandes bases de datos. Por ejemplo, gracias al algoritmo de **búsqueda binaria** (*que tiene una complejidad en tiempo de $O(\log(n))$ y espacio de $O(1)$*), es posible encontrar datos de una manera fácil y rápida sin tener que procesar todos los datos y sin tardar mucho tiempo. Los algoritmos de ordenamiento también son esenciales cuando se manejan grandes cantidades de datos, ya que si se quiere tener una mejor visualización o si se requiere hacer una búsqueda, es necesario tener los datos ordenados de manera descendente, ascendente, etc. Por ejemplo, el algoritmo **merge sort** (*que tiene una complejidad en tiempo de $O(n\log(n))$ y espacio de $O(n)$*) sirve para ordenar un arreglo o vector n elementos de manera ascendente o descendente, depende de cómo se programe. Este algoritmo es uno de los mejores en ordenamiento, lo malo de este es que utiliza espacio adicional en la memoria. Otra cosa muy importante de estos algoritmos es su complejidad de tiempo, y su complejidad en el espacio. Existen algoritmos que tienen una baja complejidad de tiempo y espacio, es posible verificar esto con la notación Big-O, por lo general mientras menos espacio y tiempo tome un algoritmo, mejor es. Cada base de datos y problema tiene maneras diferentes de solucionarse por lo que hay que saber las ventajas y desventajas de usar cada algoritmo para aplicar el mejor y más útil en cada caso específico.

Durante esta actividad, aprendí a aplicar algoritmos de búsqueda y ordenamiento en un caso práctico. También aprendí sobre las estructuras en C++ y me di cuenta que son muy útiles para ordenar información, me recordaron a las clases pero pienso que son más sencillas. La notación de Big-O también me sirvió para darme una mejor idea de cómo funcionan los métodos que son propios de nuestro código.

Bibliografia:

Zaveri, M. (2022, 4 mayo). An intro to algorithms: searching and sorting algorithms.
codeburst.io.

<https://codeburst.io/algorithms-i-searching-and-sorting-algorithms-56497dbaef20>