



# Tecnológico de Monterrey

## **Actividad Integradora 1.**

Yael García Morelos | A01352461

**Programación de estructuras de datos y algoritmos fundamentales  
(Gpo 850)**

Profesor. Eduardo Arturo Rodríguez Tello.

24 de marzo del 2025

Desde el inicio de la actividad, se presentó un gran desafío, pues considero que realmente no era consciente del tiempo que había pasado y que en esta actividad debía de mostrar las habilidades aprendidas durante las primeras cinco semanas, cabe mencionar que gracias a esta actividad me percate del importante papel que tienen los algoritmos de búsqueda y ordenamiento, sobre todo, en casos donde se manejan grandes cantidades de información. Al inicio de la actividad me imaginé que mi computadora, al no tener especificaciones técnicas sobresalientes, iba a tardar en procesar los datos, sin embargo, fue ahí donde me di cuenta de la importancia de tener un código eficiente.

Mientras realizaba la actividad, al inicio opté por implementar *quick sort*, ya que me siento más cómodo y hasta cierto punto me parece más sencillo de aplicar, sin embargo, me percate que realmente no era tan eficiente, pues a pesar de que *quick sort* y *merge sort* tengan la misma complejidad de  $O(n \log n)$ , *quick sort* puede llegar a ser poco estable en caso de no elegir un pivote correcto, pudiendo tener una complejidad de  $O(n^2)$  en el peor de los casos. Fue por esta razón que al final decidí cambiar el método de ordenamiento a *merge sort*.

Otro punto positivo que le veo a *merge sort* es que este método está más enfocado en dividir para luego fusionar el orden, lo cual evita la necesidad de mover datos de una posición a otra.

La búsqueda binaria se fundamenta en la división reiterada del conjunto de datos por la mitad, contrastando la fecha que se busca con la fecha en el centro del rango existente. Si la fecha es inferior, se elimina la mitad más alta, mientras que si es superior, se elimina la mitad más baja. Esto disminuye la cantidad de comparaciones a cerca de  $\log(n)$ . Fue por esto, que no se dudó en implementar el método de búsqueda binaria en vez de una búsqueda recursiva, siendo poco eficaz gracias a su complejidad de  $O(n)$ .

Desde mi punto de vista, la parte más complicada de esta actividad, fue la búsqueda de registros con diferencia de D días. A pesar de que cuando leí las instrucciones, me pareció lo más sencillo de aplicar, al final fue lo que consumió más de mi tiempo, tuve complicaciones con la lógica a la hora de calcular la diferencia de días, pues entre que tenía que convertir de *string* a *int*, realizar los cálculos, volver a pasar a *string* y realizar comparaciones. También, intenté aplicar punteros, esto con la finalidad de únicamente iterar una vez los datos ordenados para mantener una complejidad temporal de  $O(n)$ , siendo lo más eficiente posible.

En caso de no haber utilizado los punteros, la otra solución que se me había ocurrido era comprobar las posibles combinaciones, sin embargo, por la cantidad de datos y una complejidad de  $O(n^2)$  resultaba el peor método a usar, pues los recursos y tiempo no se utilizarían de manera eficaz.

### **Referencias.**

De La Cueva, V. M., González, L. H., & Salinas, E. G. (2020). Estructura de datos y algoritmos fundamentales. (1.a ed.) [eBook]. Editorial Digital Tecnológico de Monterrey.