Santiago Martin del Campo Soler – A01713396

20 - 01 - 2026 - Reflexión sobre Eficiencia de Algoritmos de ordenamiento y búsqueda.

En esta Actividad 1.4, el equipo desarrolló un programa para analizar una bitácora de accesos a un sistema. El objetivo principal fue procesar una gran cantidad de registros de manera correcta y en un tiempo razonable, utilizando algoritmos de ordenamiento y búsqueda eficientes, específicamente algoritmos de ordenamiento para organizar la información y búsqueda binaria para realizar consultas por rangos de fechas.

Debido a la gran cantidad de registros contenidos en la bitácora,al momento de elegir los algoritmos a utilizar elegimos uno de alta y uno de baja eficiencia. A lo largo del desarrollo de la aplicación se aplicaron conceptos de análisis de algoritmos y complejidad temporal, lo que permitió comprender de forma práctica la importancia de seleccionar algoritmos adecuados según el problema a resolver.

Algoritmos de Ordenamiento

Durante el desarrollo de la aplicación estuvimos discutiendo sobre cuáles algoritmos usar y, finalmente, nos decidimos por implementar Bubble Sort y Quicksort, ya que representan dos enfoques muy distintos. Bubble Sort siendo un algoritmo de baja eficiencia y Quick Sort siendo un algoritmo de alta eficiencia, lo cual nos permitió comparar directamente su desempeño utilizando los mismos datos.

Bubble Sort

Bubble Sort es un algoritmo de ordenamiento que es simple de entender e implementar, pero con poca eficiencia, ya que tiene una complejidad temporal de $O(n^2)$. Esto se debe a que realiza comparaciones repetitivas entre elementos adyacentes, lo que provoca que el número de operaciones aumente demasiado cuando el tamaño de los datos es grande. En esta aplicación, el uso de Bubble Sort fue meramente académico, con el objetivo de compararlo con el desempeño de un algoritmo de alta eficiencia como lo es Quick Sort.

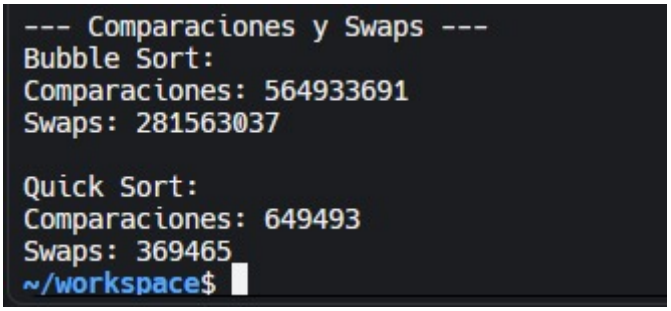
Quick Sort

Quicksort, por otro lado, es un algoritmo mucho más eficiente que Bubble Sort, con una complejidad temporal promedio de O(n log n). Mediante su estrategia de dividir el problema en segmentos más pequeños, logra ordenar grandes cantidades de datos de forma considerablemente más rápida.

Comparación en Trabajo

Tras desarrollar por completo el programa, se logró comparar el desempeño de ambos algoritmos utilizando la misma base de datos.

Para esta comparación se trabajó con una base de datos de aproximadamente 16,000 registros, realizando la consulta dentro de un rango de fechas específico, comprendido entre Octubre 26 13:37:41 como fecha inicial y Octubre 30 23:48:41 como fecha final. Los resultados obtenidos mostraron que Bubble Sort realizó una cantidad extremadamente alta de comparaciones e intercambios, mientras que Quick Sort realizó una cantidad mucho menor de operaciones.



Esto confirmó que Bubble Sort no es una opción viable para manejar grandes volúmenes de datos, mientras que Quick Sort ofrece un mejor desempeño y es más adecuado para situaciones como el análisis de una bitácora. Los resultados obtenidos fueron consistentes con la complejidad temporal de ambos algoritmos.

Fuentes Bibliográficas:

Miñarro, A. (2025, enero 15). *Algoritmos en informática: Eficiencia e Innovación*. Initium blog. https://www.initiumsoft.com/blog_initium/algoritmos-en-informatica/

Sorting algorithms. (2024, enero 24). GeeksforGeeks. https://www.geeksforgeeks.org/dsa/sorting-algorithms/

Busqueda Binaria – (s/f). Khanacademy.org. Recuperado el 21 de enero de 2026, de https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search