

## Documento de reflexión

Considero que los algoritmos de ordenamiento y búsqueda binaria son bastante útiles en este tipo de casos y problemas que se nos presenta en esta situación problema. Esto debido a que tenemos una manera más sencilla de visualizar y acceder a los datos y registros que estamos analizando en este caso, por ser fechas, de la más antigua a la más reciente. De este modo se vuelve bastante sencillo para nosotros seres humanos acceder y manipular la información a nuestro gusto de una manera más sencilla y cómoda. Para realizar el ordenamiento de todos los logs provenientes del archivo bitacora.txt implementé el algoritmo de ordenamiento quickSort. Esto debido a que, hay un escenario en el que el peor caso su complejidad temporal puede llegar a ser de  $O(n^2)$ , este es bastante más rápido en comparación a otros vistos en clase como BubbleSort o InsertionSort porque en promedio y, en la mayoría de los casos incluyendo este, su complejidad temporal es  $O(n \log n)$  y una complejidad espacial  $O(\log n)$ . Es de ese modo que pude implementar ese algoritmo para que pudiera leer el archivo de texto proporcionado y realizar el ordenamiento. Igualmente, el algoritmo binarySearch es bastante eficiente, teniendo una complejidad temporal promedio  $O(\log_2 n)$ .

Esta evidencia fue bastante retadora ya que al principio es un poco abstracto tratar de aprender sobre y entender los algoritmos que se han estado viendo en esta clase. Considero también que ha sido una muy buena actividad y práctica ya que es una manera muy interesante de seguir desarrollando nuestras competencias como ingenieros en software.