

## Reflexión Evidencia 1

El canal de Suez es una de las creaciones más importantes en el mundo del comercio. Con una longitud de 195 kilómetros y una anchura de alrededor de 200 metros, este estrecho de agua artificial permite una ruta de comercio más fácil entre Europa y Asia, pasando por medio de Egipto, en vez de tener que rodear toda África para poder llegar a uno de estos dos continentes.

Al día, pasan alrededor de 50 a 100 barcos, y a lo largo del tiempo, se han guardado los barcos que han pasado. Desafortunadamente, estos registros se encuentran desordenados, por lo que necesitamos ordenarlos, y ponerlos en un formato presentable para buscarlos, dependiendo de las necesidades del usuario.

Para organizar los datos, se utilizó el método Quicksort, el cual se especializa en la filosofía de “dividir y vencer” en el que de un punto pivote de una matriz, se comparan los diferentes elementos, y se encuentra el punto en el que se encuentra el elemento pivote, dividiendo en dos listas a la matriz original, por medio de la recursión. Este algoritmo tiene una eficiencia de  $n \cdot \log(n)$ , (peor caso siendo  $n^2$ ) por lo que hasta ahorita se considera el algoritmo más rápido visto en clase, siendo más rápido que merge sort. Se utilizó para organizar los datos por número de serie UBI, luego fecha de entrada al canal, y al final la hora de entrada.

Para la búsqueda de los elementos del usuario por medio del UBI, se utilizó el algoritmo binary search, en el que se busca desde la mitad de la matriz el elemento que se quiere encontrar, y subsecuentemente encuentra la mitad de la mitad, hasta encontrar el elemento necesario. Con una eficiencia de  $\log(n)$  en todos los casos, se utilizó para poder encontrar un elemento de lo que busca el usuario, ya sea por medio de 3 dígitos de la UBI, o la UBI completa de 5 dígitos. Después de encontrar uno de los datos, puede ser posible que se tengan más de una instancia de una UBI o serie de UBIs, por lo que se utilizó sequential search para poder encontrar el primer elemento, y subsecuentemente imprimir los elementos encontrados por UBI. Este algoritmo tiene una eficacia de  $O(n)$ .

Al tener tantos elementos, alrededor de 13,000, se puede tardar un ratito al trabajar con estos algoritmos, pues las limitantes del CPU y software que se tenga. Pero, con el uso de estos algoritmos, se pudo ahorrar bastante tiempo en la ordenación y búsqueda de datos, comparados con el uso de lápiz y papel.