

## Reflexión Actividad 1.3

Dentro del código de la actividad 1.3 optamos por usar el algoritmo de ordenamiento Quick Sort debido a que este tiene una complejidad en el peor de los casos de  $O(n^2)$  y en el mejor de los casos un  $O(n\log(n))$  por lo que es más eficiente que un algoritmo de ordenamiento que siempre es  $O(n^2)$  como el Bubble Sort. Para que el algoritmo de ordenamiento funcionara convertimos los datos del archivo en un numero entero que fuera capaz de ordenarse siendo por ejemplo Aug 28 00:20:01 a 828002001 y ya de ahí hacer las comparaciones. Como cada dato se guardaba en un struct cada vez que movía el numero todos los demás datos se movían con él.

Ahora para la manera en que buscamos los datos que nos pedían usamos la búsqueda binaria ya que su complejidad  $O(\log(n))$  es menor a una búsqueda normal que es  $O(n)$  y finalmente ese resultado lo almacenábamos en un archivo de salida con los datos pedidos