

## Reflexión

En esta situación problema se nos dió un archivo con diferente información representando accesos direcciones IP 's de computadoras con el objetivo de obtener los registros correspondientes para un rango de fechas. La información dada por cada acceso incluía el mes, día, hora, dirección IP y razón de falla.

El primer paso para obtener los registros correspondientes para un rango de fechas fue ordenar la información por fecha para realización de las búsquedas. Para hacer esto creamos una clase llamada registro y guardamos la información del archivo en un vector de tipo registro. Además, para cada elemento de tipo registro creamos su clave con el método de crearClave, la cual se llama dentro del constructor de la clase registro. Finalmente, para ordenar los datos en base a sus claves, utilizamos el algoritmo de ordenamiento de tipo merge con una complejidad de tiempo de  $O(n \cdot \log n)$ . Después, para unir estos datos utilizamos un método llamado unir en nuestro programa de complejidad  $O(n)$ . Para hacer las comparaciones en la función de merge hicimos uso de operator overloading en una nuestra clase de registro con el fin de poder comparar las claves directamente. Utilizamos este algoritmo porque como estamos trabajando con una gran cantidad de datos, este es más eficiente que el algoritmo de ordenamiento de intercambio y burbuja.

Finalmente para buscar registros correspondientes en un rango de fechas le pedimos al usuario la fecha inicial y final y después llamamos a la función de búsqueda. Esta función agarra las claves de la fecha inicial y final llamando al método de orden clave. Después llama el método de busquedaBinaria utilizando como parámetro fecha inicial para obtener posición inicial y fecha final para obtener posición final respectivamente. Finalmente, el método de búsqueda checa si la posición inicial y final existen (no son -1 ) y si sí existen hace un for loop en donde despliega los registros desde la posición inicial a la posición final. Utilizamos estos dos métodos porque son eficientes, teniendo una complejidad de tiempo de  $O(n)$  y  $O(\log n)$  para los algoritmos de busqueda y busquedaBinaria respectivamente.

Me gustó mucho poder aplicar métodos de ordenamiento y búsqueda para una situación problema de verdad. Creo que es sumamente importante el saber dominar estos algoritmos al igual que entender cómo funciona para poder tener la intuición de cómo resolver este tipo de problemas de la forma más eficiente.