

### Reflexión Actividad 1.3

Cristóbal Alberto Escamilla Sada - A00827074

El objetivo de esta actividad fue organizar un archivo de entrada "bitacora.txt" conteniendo más de 16,000 datos, buscar y desplegar el rango de datos proporcionado por el usuario, así como transferir los datos organizados a un archivo nuevo. Para lograr esto, lo primero que se consideró fue el formato del archivo bitacora.txt. Este está organizado por renglones, cada uno conteniendo un registro diferente. Cada renglón cuenta con 5 atributos: mes, día, hora, dirección IP, y razón de falla. Para facilitar el uso de vectores decidimos crear una clase "Registro.h" con dichos atributos. Adicionalmente, creamos un atributo entero llamado "claveOrden" en el que se almacena una clave dependiendo del mes y día del mismo registro. Por ejemplo, el registro con fecha "Aug 1" tiene una clave con valor de 801 debido a que este se refiere al mes 8 y día 1.

Se hizo la lectura del archivo y los datos fueron colocados en un vector de apuntadores de tipo Registro. Para el ordenamiento de los datos, se utilizó la función "ordenaMerge". Esta opera mediante el famoso método del emperador Julio César, "divide y vencerás". La función se encarga de partir el conjunto de  $n$  datos a la mitad, repitiéndolo  $\log_2 n$  veces recursivamente hasta tener  $n$  datos individuales. Posteriormente los une a través del llamado a la función "unir", que se encarga de la comparación de los datos. En este caso se comparan los valores de claveOrden en cada Registro para ordenarlos de forma ascendente (por fecha). La función "unir" compara los datos linealmente por lo que tiene una complejidad de  $O(n)$ . Debido a que la función "ordenaMerge" hace dos llamadas recursivas con un tiempo de ejecución de  $\log_2 n$  y una llamada a la función "unir"— de un lineal, se concluye que la complejidad computacional de "ordenaMerge" es de  $O(n \log n)$ .

Una vez organizados los datos, se le solicita al usuario que tecleé dos fechas; una de inicio y una de fin. A partir de este rango se hará una búsqueda para posteriormente desplegar los registros correspondientes. La búsqueda así como el despliegue de los datos se hacen a través de la función "busqueda". Esta hace dos llamados a la función "busqBinaria"— una para encontrar el dato de inicio y otra para encontrar el dato de fin. Posteriormente despliega los datos a través de un for loop iterando dentro de este rango. La búsqueda funciona de manera binaria, dividiendo el conjunto de datos (a través de las variables enteras ini, fin y mit) y comparando sus claves de ordenamiento de fecha hasta encontrar el valor deseado. A esta se le agregó un parámetro de entrada booleano llamado "inicio", que determina si el dato que se está buscando es uno de inicio o de fin. Debido a que existen fechas repetidas dentro del registro, la búsqueda se tiene que encargar de desplegar todos los datos correspondientes a dichas fechas. Ya que la función binaria haya encontrado la fecha solicitada, revisa si el dato es de inicio y en este caso, si la fecha en el índice anterior es igual, el valor de fin se modifica a  $mit-1$ . De lo contrario, se regresa el valor de mit, representando el registro con la fecha solicitada. Se repite la búsqueda binaria sucesivamente hasta que el dato solicitado haya sido encontrado. Si el valor booleano de "inicio" es falso y el algoritmo encuentra la fecha solicitada, revisa si la fecha del índice posterior es la misma. En caso de que sí, el valor de ini se modifica a  $mit+1$ . De lo contrario, se regresa el valor de mit, representando el registro con la fecha solicitada. Debido a que la función divide los datos  $n$  veces, este tiene una complejidad de  $O(\log n)$ . Debido a que la función "busqueda" hace dos llamadas a la búsqueda binaria y una iteración lineal para el despliegue de los datos, esta tiene una complejidad de  $O(n)$ .

Este proyecto me pareció bastante interesante debido a que es una introducción al tipo de programas que estaremos desarrollando a lo largo del semestre. Cada vez siento que mis habilidades de programación y mi pensamiento computacional se desarrollan más. Conforme hago los trabajos me doy cuenta que cada vez pienso en soluciones más eficientes y profesionales para resolver los problemas requeridos. Me siento motivado para seguir aprendiendo sobre los conceptos de las estructuras de datos y sus algoritmos. Sé que esta clase es fundamental para mi desarrollo como Ingeniero en Tecnologías Computacionales.