

**Act 6.2 - Reflexión Final de Actividades Integradoras de la Unidad de  
Formación TC1031 (Evidencia Competencia)**



**Tecnológico  
de Monterrey**

Alumno: Ricardo González Leal

Matrícula: A01639036

Materia: Programación de estructuras de datos y algoritmos fundamentales  
(TC1031)

Grupo: 13

Profesor: Luis Ricardo Peña Llamas

Fecha: 26/11/21

¿Cuáles son las más eficientes?

Considero que las actividades integradoras más eficientes que realicé en esta unidad de formación son las siguientes:

Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales: Debido a que esta fue la primer actividad integradora, considero que es una de las cuales apliqué más conocimientos, iniciando desde el tener que aplicar lectura de archivos para leer la bitácora de accesos que se nos proporcionó, posteriormente se utilizó un método de ordenamiento para ordenar los registros por fecha (en nuestro caso hicimos uso de QuickSort por ser de los más eficientes), y también utilizamos el método de búsqueda secuencial para poder buscar registros en cierto rango de fechas para posteriormente imprimirlas en orden descendente y guardarlas en un archivo de texto; cabe mencionar que fue necesario hacer uso de un diccionario para relacionar un valor entero para cada mes del año, asimismo se utiliza una búsqueda secuencial para encontrar los accesos que se encuentran en el rango. Además, es importante mencionar que esta actividad fue la base para las siguientes dos actividades.

Act 2.3 - Actividad Integral estructura de datos lineales: Esta actividad retoma mucho de lo utilizado en la actividad anterior, sin embargo, se agrega la funcionalidad de que al imprimir un registro, se imprima también el número de registros encontrados con esa IP en particular. Para lograr esto lo que hicimos fue implementar un diccionario que toma como llave una tupla conformada por la fecha y por la ip, mientras que en el value se almacena el número de IPs coincidentes. Al imprimir cada registro, se verifica si en el diccionario existe algún otro registro con la misma IP y con la misma fecha, de ser así se incrementa el value. Esta implementación fue compleja y quedamos muy satisfechos con los resultados.

Act 3.4 - Actividad Integral de BST:

Esta actividad fue algo diferente a las demás, ya que en esta se cambió la manera en la que se almacenaba la información de los registros a un árbol binario. En este arbol binario, se almacena en cada nodo el número de IPs coincidentes como la llave, mientras que como valores se almacenaron la dirección IP y la fecha de acceso. Una vez que se tuvo el arbol binario creado, se realizó una función para obtener a los 5 nodos con más accesos, para lo cual se implementó un inorder inverso.

¿Cuáles podrías mejorar y argumenta cómo harías esta mejora?

Considero que todas las actividades tienen áreas de oportunidad, esto debido a que estas actividades fueron realizadas al poco de tiempo de al

haber adquirido los conocimientos requeridos, sin embargo conforme fuimos avanzando en la unidad de formación, estos conocimientos se fuerzon reforzando. Algunos de los puntos notorios que creo se pueden mejorar son los siguientes:

En la Actividad 1.3, para poder realizar la búsqueda de los accesos en fechas delimitadas, lo que se hace es almacenar en una variable entera un valor representativo a la fecha para cada registro, de manera la que la fecha es igual al siguiente formato: MM-DD-HH-MM-SS

Ejemplo:

Jun 01 00:00:00 = 0601000000

Una vez teniendo este formato, lo que se hace es realizar una búsqueda secuencial entre todos los accesos partiendo del int representativo a la fecha. Esto considero que fue una buena solución para términos de lo que se nos pedía, sin embargo no es la más eficiente ya que la búsqueda secuencial tiene que recorrer todos los elementos hasta encontrar los coincidentes. Para poder mejorar esto, lo que se me ocurre es aprovechar las ventajas que tienen los árboles binarios, teniendo una estructura en la cual los nodos de la izquierda de un nodo sean menores a este y los de la derecha sean mayores, para lo cual, el valor representativo de la fecha obtenido previamente se puede utilizar como la llave de cada nodo. De esta manera es mucho más eficiente la búsqueda de accesos, teniendo un tiempo de complejidad mucho menor que la solución implementada.

**Conclusión:**

La unidad de formación de Estructura de Datos y Algoritmos hizo que aprendiera distintos Métodos de Ordenamiento, Métodos de Búsqueda, Estructuras de Datos Lineales (Pilas, Colas, Listas Ligadas), Estructuras de Datos No Lineales (Árboles Binarios, Árboles AVL, Heaps, Grafos), Técnicas de Hashing, así como el implementar conceptos para utilizar estos como la recursividad, el manejo de apuntadores, memoria dinámica, características de los árboles, recorridos en árboles, recorridos en grafos. Asimismo fue de gran importancia conocer qué es la Complejidad Computacional y la Notación Asintótica, ya que como programador, esto es algo que considero muy importante saber, ya que para todos los programas que diseñe, el tiempo de complejidad da una idea sobre qué tan eficiente es este. Finalmente quiero mencionar que esta unidad de formación definitivamente es una de las más importantes de la carrera ya que, además de ser completamente necesaria para las entrevistas laborales, también las estructuras de datos se utilizan en todos los programas que realizamos, aunque

quizás muchos de los lenguajes de programación que utilizamos ya tienen algunas de estas estructuras de datos incorporadas, y para las que no están implementadas, es vital saber cómo hacerlas y cómo funcionan.

Repositorio de los trabajos:

[https://github.com/RicardoGLEal/TC1031\\_Data-Structures-Algorithms](https://github.com/RicardoGLEal/TC1031_Data-Structures-Algorithms)