

Escribir en forma individual un documento llamado "**reflexFinTC1031.pdf**" en donde expongas tu reflexión sobre el desarrollo de subcompetencias sobre las soluciones que presentaste en las diferentes actividades integradoras:

1. ¿Cuáles son las más eficientes?
2. ¿Cuáles podrías mejorar y argumenta cómo harías esta mejora?

Subcompetencia SICT0301 nivel B

- Será capaz de integrar conocimiento de las ciencias de ingeniería para reconocer las variables que tienen desviaciones sobre el estado normal de funcionamiento de sistemas computacionales.

Subcompetencia SICT0302 nivel B

- Sabe integrar información disponible, modelos seleccionados y herramientas de ingeniería para obtener soluciones de problemas asociados con tecnologías de información.
- Identifica información que cumpla con criterios de calidad para la generación de soluciones al problema.

Subcompetencia SICT0303 nivel B

- Será capaz de integrar conscientemente en las acciones el análisis de priorización de soluciones basadas en criterios de significancia, completitud y optimalidad.

Considero que este curso de estructura de datos es uno de los más retadores que he tenido en la carrera, ya que contó con actividades que requirieron que desarrollara al máximo las subcompetencias, gracias a estas subcompetencias desarrolladas fui capaz de resolver cada una de las actividades.

En la subcompetencia SICT0302 nivel B, pude aplicar en todas las actividades de evidencia, ya que tuve que ser capaz de integrar la información disponible y herramientas de ingeniería como los algoritmos computacionales adecuados para dar solución a los archivos de bitácora, tuve que ser capaz de aplicar algoritmos de ordenamiento, punteros, almacenar datos en diferentes tipos de estructuras de datos como lo son listas ligadas, fila, stacks, listas ligadas, grafos, arboles binarios y tablas hash y construir funciones para manipular los datos de cada una de ellas. Tuve que ser capaz de identificar el problema dado, hacer una investigación,

consultar distintas fuentes dando pie a la subcompetencia SICT0303 nivel B y SICT0301 nivel B, fui capaz de integrar conocimientos de las ciencias de la ingeniería para reconocer las variables que tienen desviaciones sobre el sistema normal del funcionamiento de sistemas computacionales. Pude realizar análisis, priorizar algunas soluciones de otras mejorando eficiencia del algoritmo computacional y tomando las decisiones adecuadas.

Considero que las estructuras de datos más eficientes son los grafos y las hash tables, ya que las estructuras de grafos se pueden representar de distintas maneras para una infinidad de soluciones y aplicaciones, pienso que los grafos representan muy bien la realidad, esto debido a que se comporta como una red y así son los datos en la vida real, muchas veces son desordenados, de unos parten otros y no siempre se lleva la misma dirección. A diferencia de un árbol binario, los grafos pueden llegar a tener infinidad de salidas y los árboles binarios solamente cuentan con 2.

Las hash tables considero que son muy eficientes esto debido a que para la búsqueda de datos no se necesita recorrer todo el arreglo, si no que es bastante fácil ubicarla con una complejidad de $O(1)$, sin embargo tenemos que llevar a cabo un buen análisis para evitar las colisiones y si no se crea una función de hashing adecuada para el conjunto de datos, esto puede ocupar demasiada memoria y tener muchas colisiones. Por lo que considero que si se sabe aplicar bien con un conjunto de datos que le favorezca a esta técnica se puede obtener mucho provecho.

Algunas restricciones que tenemos al momento de utilizar árboles binarios es que solamente puede contar con 2 nodos emergentes, es muy rápido al momento de hacer las búsquedas, ya que en cada iteración se reduce a la mitad, quedando en un orden logarítmico. Creo que este tipo de estructura podría mejorarse trabajando en conjunto con otro tipo de estructuras, por ejemplo, ocupar un BST para algoritmos que impliquen búsqueda y una lista ligada o doble lista ligada para programas de datos que requieran más inserción y eliminación de datos.

Y hablando de listas ligadas tenemos algunas limitantes en cuanto a la búsqueda y al acceso, teniendo una complejidad de $O(n)$, esto quiere decir que tiene que pasar por toda la estructura de datos uno por uno antes de encontrar el deseado, esta sería buena ocuparla cuando se tienen conexiones lineales directas entre 2 elementos, de esta manera se podría mejorar su desempeño en algún programa computacional.

Es por esto que es importante analizar la naturaleza de los datos antes de elegir una estructura de datos adecuada para la solución del programa.