

David Alejandro Lozano Arreola A01722728

Programación de estructuras de datos y algoritmos fundamentales

Evidencia 2 - Reflexión

Todo software puede verse como un conjunto de algoritmos o recetas para llevar a cabo ciertas tareas. Además, es necesario optimizar dichos algoritmos para realizar las tareas de la manera más eficiente posible. Esto es importante en ámbitos como el científico, en el cual se busca encontrar los límites de la computación, lo cual, suele tardar tiempos más largos que la historia de la humanidad, en caso de no hacerse de manera eficiente. Por otra parte, en el ámbito de los negocios. Una mejora en rendimiento puede significar menos tiempo en servidores, lo cual, ahorra grandes cantidades de dinero a las empresas.

En su parte más básica, optimizar significa buscar algoritmos y estructuras de datos capaces de resolver problemas de manera eficiente. Por ejemplo, en la evidencia se pudo haber utilizado un vector en vez de una lista enlazada. Sin embargo, existen casos en donde esto no es posible. De hecho, las listas enlazadas se inventaron para resolver el problema causado por los vectores y su necesidad de tener direcciones de memoria contiguas. Esto es importante debido a que en la actualidad es muy común encontrar sistemas de hardware que implementan el procesamiento de datos en paralelo a través de varios procesadores. Además, al momento de guardar archivos en un disco duro (aka. HDD or Hard Drive Disk). Es muy común que estos queden en espacios de memoria esparcidos por el disco duro. Lo cual, causa la necesidad de utilizar una lista enlazada para navegar los datos.

Finalmente, cabe destacar que la optimización es parte fundamental del software en todos los niveles. Además de pensar en conceptos de bajo nivel como las estructuras de datos, también es importante tomar en cuenta aspectos como la experiencia del usuario, la cual tiene que ser lo más responsiva y ágil posible para promover el uso de la aplicación. Es por eso que se utilizan conceptos como el multi-threading o el non-blocking-io en Node.js.