

# Informe Tarea N°2 EDD

Luis Donoso Carimán

El objetivo de este informe es realizar un análisis comparativo sobre el orden de complejidad de las diversas funciones requeridas para el correcto funcionamiento de la base de datos pedida, tomando en cuenta para cada caso el tipo de estructura que se está utilizando.

Para la realización de esta tarea se solicitó utilizar tres tipos distintos de estructuras de datos, los cuales fueron Árbol Binario de Búsqueda, Árbol AVL y Árbol 2-3, con sus respectivas funciones de Insertar, Buscar y Eliminar un Anime dentro de la base de datos. En el primer caso, en el Árbol Binario de Búsqueda, la función de insertar es recursiva de orden constante, al igual que su función buscar y eliminar, por otro lado con el Árbol AVL la complejidad en la inserción aumenta ya que debe rebalancearse constantemente a medida que cada nodo es agregado al sistema, pero al momento de la búsqueda el tiempo de procesamiento es mucho menor, ya que la profundidad del árbol se mantiene siempre en el mínimo, otorgando así la complejidad de  $n \log(n)$ , en el caso del Árbol 2-3, la inserción es también más compleja debido a que también la profundidad del árbol se mantiene en el mínimo y se rebalancea a medida que entran datos de manera similar al caso anterior, pero en cambio para la búsqueda y eliminación el tiempo se reduce incluso más ya que al momento de comparar los datos se realiza con los valores extremos de los nodos para así hallar en primer lugar el nodo en el cual el dato está alojado, por lo cual el desempeño es similar al de una skiplist.

Por lo anterior mencionado, la estructura más adecuada para las funciones más pedidas (hallar el anime con mayor puntuación y el de menor puntuación), es más efectivo utilizar un árbol 2-3 por su capacidad de procesar una mayor cantidad de datos en tiempo de orden logarítmico. El árbol AVL es la segunda opción más conveniente ya que su orden es de  $n \log(n)$  y el peor caso posible sería el de ABB por su posibilidad de comportamiento de orden  $n$ , que sería prácticamente el comportamiento de una lista.