



## **Evidencia 1 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales**

Valeria Pérez Alonso

A00833973

Programación de estructuras de datos y algoritmos fundamentales

TC1031.608

David Alonso Cantú Delgado

## Estudio comparativo de la eficiencia de los métodos de ordenación

Una de las actividades más frecuentes al trabajar con datos es la búsqueda, manipulación y procesamiento con el fin de obtener información u ocupar aquellos datos para otras funcionalidades. Para ello es bien conocido el uso de métodos de ordenación y búsqueda que hacen de manera más eficiente aquellas actividades, sin embargo dependiendo del tipo dato, cantidad de registros y recursos computacionales se debe escoger los mejores métodos para hacer de manera más eficiente esos procesos.

En la Evidencia 1, que tiene como objetivo hacer un programa de búsqueda de registros dentro de un rango, se ocuparon 2 algoritmos conocidos. El primero el Quick Sort, uno de los métodos de ordenamiento más conocidos por su velocidad y baja necesidad de memoria y el Binary Search que ha mostrado una gran eficiencia con cantidades masivas de datos.

Table 1: Summary of the best-case, average-case and worst-case[2]

<i>Algorithm</i>	<i>B(n)</i>	<i>A(n)</i>	<i>W(n)</i>
HornerEval	n	n	n
Towers	$2^n$	$2^n$	$2^n$
LinearSearch	1	n	n
BinarySearch	1	logn	logn
Max, Min , MaxMin	n	n	n
InsertionSort	n	$n^2$	$n^2$
MergeSort	nlogn	nlogn	nlogn
HeapSort	nlogn	nlogn	nlogn
QuickSort	nlogn	nlogn	$n^2$
BubbleSort	n	$n^2$	$n^2$
SelectionSort	$n^2$	$n^2$	$n^2$
GnomeSort	n	$n^2$	$n^2$

Ambos algoritmos utilizados ocupan la técnica de dividir y conquistarás, que ocupa principalmente dividir la cantidad de elementos en conjuntos de elementos más pequeños para trabajar con ellos. Bajo esa propuesta el Quick Sort tiene una eficiencia  $O(n \log n)$  y el Binary Sort una eficiencia de  $O(\log n)$ .

Uno de los beneficios es que tiene la ventaja de no crear listas de elementos extras lo que lleva a que no se necesite el uso de memoria innecesaria para llevar al cabo los algoritmos. Y en el caso que el programa necesita hacer el ordenamiento y búsqueda de

## **Referencias**

Hammad, Jehad. (2015). A Comparative Study between Various Sorting Algorithms. IJCSNS International Journal of Computer Science and Network Security. 153. 11.