

# Análisis de Algoritmos

Tarea Corta 1

Gustavo Alvarado Orozco  
*Ingeniería en computación*  
*Tecnológico de Costa Rica sede Cartago*  
luiguillog2@gmail.com

Gustavo Blanco Alfaro  
*Ingeniería en computación*  
*Tecnológico de Costa Rica sede Cartago*  
tavoba2015@gmail.com

**Abstract**—El siguiente trabajo consiste en tomar dos algoritmos de ordenación diferentes y comparar el tiempo en que duran para ordenar listas de diferentes tamaños con datos aleatorios, mostrando una gráfica con los tiempos que le toma a cada algoritmo ordenar las diferentes listas, esto utilizando unity para mostrar la interfaz.

**Index Terms**—análisis, comparación, algoritmo

## I. INTRODUCCIÓN

El presente trabajo pretende ayudar al estudiante a iniciar en el manejo y conocimiento de Unity, a su vez, ayudarlo a dar los primeros pasos en los temas introductorios del curso como es, analisis de algoritmos.

## II. INSTRUCCIÓN DE USO

### A. Todo el código se encuentra dentro de una misma escena

Precionar el boton "START", utilizar WASD para desplazarse por el entorno y poder apreciar de mejor manera la posición de los tiempos.

## III. DESARROLLO

### A. Quick Sort

El algoritmo consta de los siguientes pasos:

1. Elegir un elemento de la lista de elementos a ordenar, al que llamaremos pivote.
  2. Resituar los demás elementos de la lista a cada lado del pivote, de manera que a un lado queden todos los menores que él, y al otro los mayores. Los elementos iguales al pivote pueden ser colocados tanto a su derecha como a su izquierda, dependiendo de la implementación deseada. En este momento, el pivote ocupa exactamente el lugar que le corresponderá en la lista ordenada.
  3. La lista queda separada en dos sublistas, una formada por los elementos a la izquierda del pivote, y otra por los elementos a su derecha.
  4. Repetir este proceso de forma recursiva para cada sublista mientras éstas contengan más de un elemento. Una vez terminado este proceso todos los elementos estarán ordenados.
- [1]. Origen del código: <http://csharpexamples.com/c-quick-sort-algorithm-implementation/> [3]

### B. Bubble Sort

Burbuja(Bubble Sort en inglés) Es un sencillo algoritmo de ordenamiento. Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran pequeñas burbujas. [2] Origen del código: <https://www.w3resource.com/csharp-exercises/searching-and-sorting-algorithm/searching-and-sorting-algorithm-exercise-3.php> [4]

### C. Resultados del Analisis Empirico

Con la realizacion de este trabajo se aprendieron herramientas básicas de Unity y se dieron a conocer las diferencias entre los dos algoritmos, como la diferencia en eficiencia entre ambos, como que el quick sort tiene un tiempo de ejecución lineal mientras que el bubble sort mantiene un tiempo cuadrático.

## IV. GIT

<https://github.com/Tavo506/An-lisis—Tarea-1.git>

## REFERENCES

- [1] Ecured.cu. (n.d.). QuickSort - EcuRed. [online] Available at: <https://www.ecured.cu/QuickSort> [Accessed 25 Feb. 2020].
- [2] Ecured.cu. (n.d.). Ordenamiento de burbuja - EcuRed. [online] Available at: [https://www.ecured.cu/Ordenamiento de burbuja](https://www.ecured.cu/Ordenamiento-de-burbuja) [Accessed 25 Feb. 2020].
- [3] Csharpexamples.com. (n.d.). C Quick Sort Algorithm Implementation — C sharp Examples. [online] Available at: <http://csharpexamples.com/c-quick-sort-algorithm-implementation/> [Accessed 19 Feb. 2020].
- [4] w3resource. (n.d.). C Sharp exercises: Bubble sort - w3resource. [online] Available at: <https://www.w3resource.com/csharp-exercises/searching-and-sorting-algorithm/searching-and-sorting-algorithm-exercise-3.php> [Accessed 19 Feb. 2020].