



República Bolivariana de Venezuela  
Ministerio del Poder Popular Para la Educación  
Universidad Simón Bolívar

## **Reporte de laboratorio**

### **Semana 6**

**Asignatura:** CI-2692

#### **Profesor:**

Guillermo Palma

#### **Hecho por:**

Luis Isea. 19-10175

Juan Cuevas. 19-10056

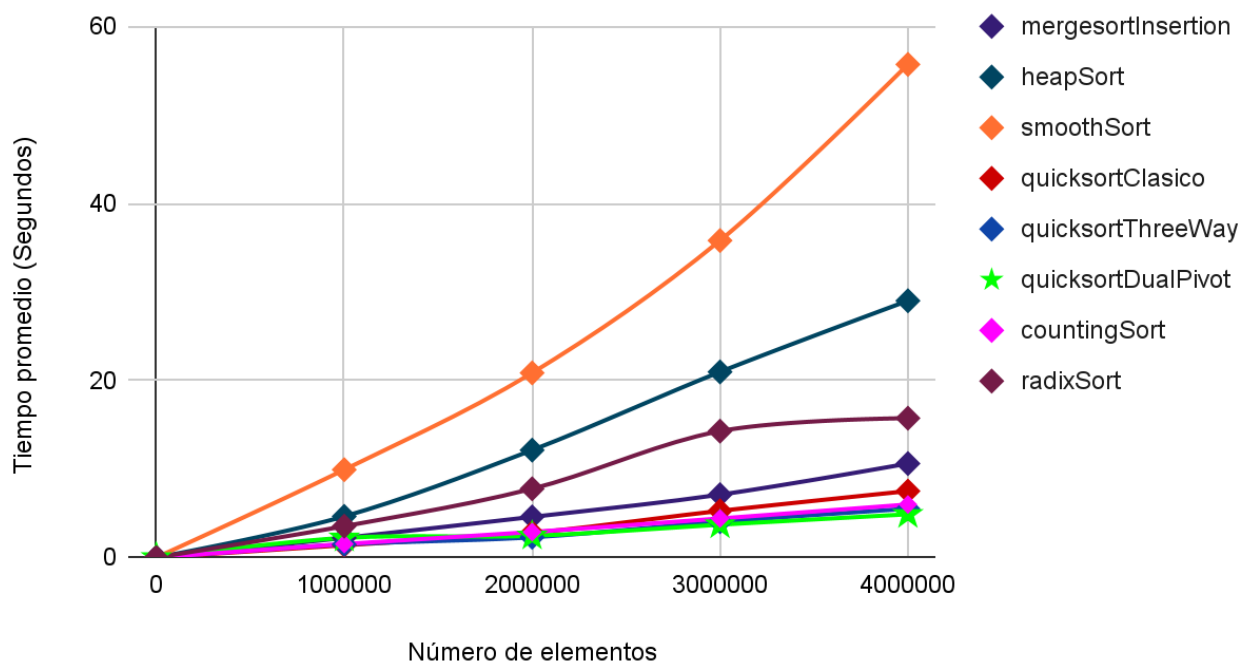
Sartenejas, junio de 2023

## Reporte

Se probó en un computador portátil con las siguientes características: Intel® Celeron® CPU N2830 @ 2.16GHz de doble núcleo (4gb de RAM), Kotlin versión 1.8.20-release-327 (JRE 17.0.6+10), OpenJDK 64-Bit Server VM Temurin-17.0.6+10. Se realizaron los ordenamientos con cada algoritmo con 3 arreglos distintos con números aleatorios. Los resultados fueron (Todos en segundos):

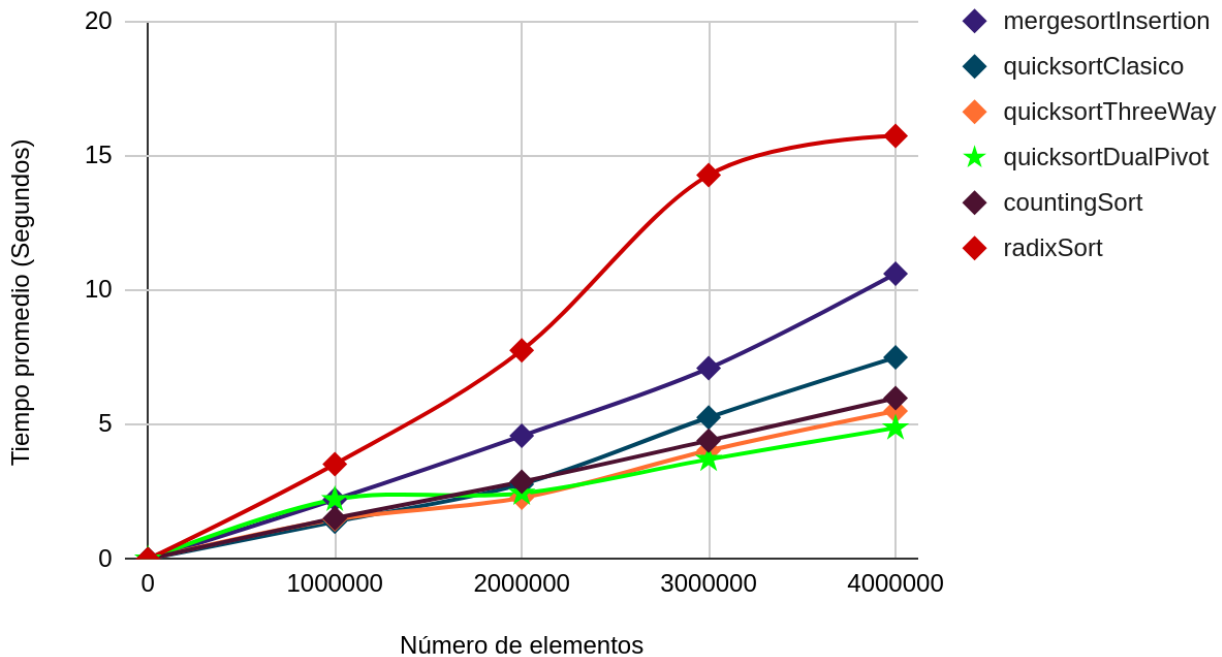
| Algoritmo          | n = 1.000.000  | n = 2.000.000   | n = 3.000.000  | n = 4.000.000  |
|--------------------|----------------|-----------------|----------------|----------------|
| mergesortInsertion | 2.213(±0.305)  | 4.582(±0.141)   | 7.093(±0.360)  | 10.607(±1.527) |
| heapSort           | 4.638(±0.419)  | 12.124(± 1.275) | 20.961(±1.498) | 29.001(±0.503) |
| smoothSort         | 9.914(± 1.336) | 20.849(±1.521)  | 35.804(±0.799) | 55.701(±3.632) |
| quicksortClasico   | 1.385(±0.132)  | 2.782(±0.111)   | 5.264(±0.570)  | 7.498(±0.562)  |
| quicksortThreeWay  | 1.476(±0.475)  | 2.279(±0.111)   | 4.034(±0.350)  | 5.507(±0.946)  |
| quicksortDualPivot | 2.213(±0.419)  | 2.445(±0.304)   | 3.705(±0.688)  | 4.875(±0.279)  |
| countingSort       | 1.516(±0.170)  | 2.876(±0.289)   | 4.404(±0.639)  | 5.984(±0.802)  |
| radixSort          | 3.532(±0.636)  | 7.758(±0.197)   | 14.270(±1.413) | 15.733(±0.431) |

## Comparación rendimiento algoritmos de ordenamiento



Uso de CPU: 96.2%    Uso de RAM: 23.6%

## Comparación rendimiento algoritmos de ordenamiento



Gráfica sin smoothSort y heapSort para apreciar mejor los algoritmos más eficientes.

Es muy importante destacar que debido a la gran cantidad de pruebas, se programó que las pruebas de todos los algoritmos se hicieran en un mismo programa que fuera imprimiendo el tiempo de ordenamiento por cada algoritmo. Es por esto que no se pudo encontrar la forma de saber cuáles algoritmos tuvieron mayor uso de CPU y cuáles tuvieron mayor uso de RAM. Simplemente se obtuvo el uso de CPU y RAM del programa que realizó todas las pruebas.