UNIVERSIDAD DEL VALLE DE GUATEMALA

ALGORITMOS Y ESTRUCTURAS DE DATOS

SECCIÓN 20

Hoja de Trabajo 3

Ordenamientos

Cristopher Javier Chávez Toc - 25199

Diego Fabricio Estrada Girón - 25230

10 de febrero de 2026

1. Herramientas utilizadas para medición de rendimiento

Para analizar el tiempo de ejecución de los algoritmos de ordenamiento se utilizaron dos métodos complementarios:

1. System.nanoTime() del lenguaje Java.
2. VisualVM 2.2 como herramienta de profiling.

El uso combinado de ambas herramientas permitió obtener tanto datos numéricos precisos para análisis comparativo como información detallada sobre el consumo de CPU de cada algoritmo.

2. Uso de System.nanoTime()

La función System.nanoTime() fue utilizada para medir el tiempo exacto de ejecución de cada algoritmo de ordenamiento.

La medición se realizó capturando el tiempo inmediatamente antes y después de ejecutar el método sort():

```
long inicio = System.nanoTime();
algoritmo.sort(numeros);
long fin = System.nanoTime();
```

El tiempo total se calculó restando ambos valores

```
long tiempoNanos = fin - inicio;
long tiempoMilis = tiempoNanos / 1_000_000;
```

Los resultados fueron almacenados en un archivo csv, el cual posteriormente fue utilizado para generar las gráficas comparativas en Excel.

Para permitir una captura adecuada de los datos, fue necesario modificar el método main agregando pausas controladas mediante Thread.sleep() y una espera final antes de cerrar el programa, lo cual facilitó la captura de información con VisualVM.

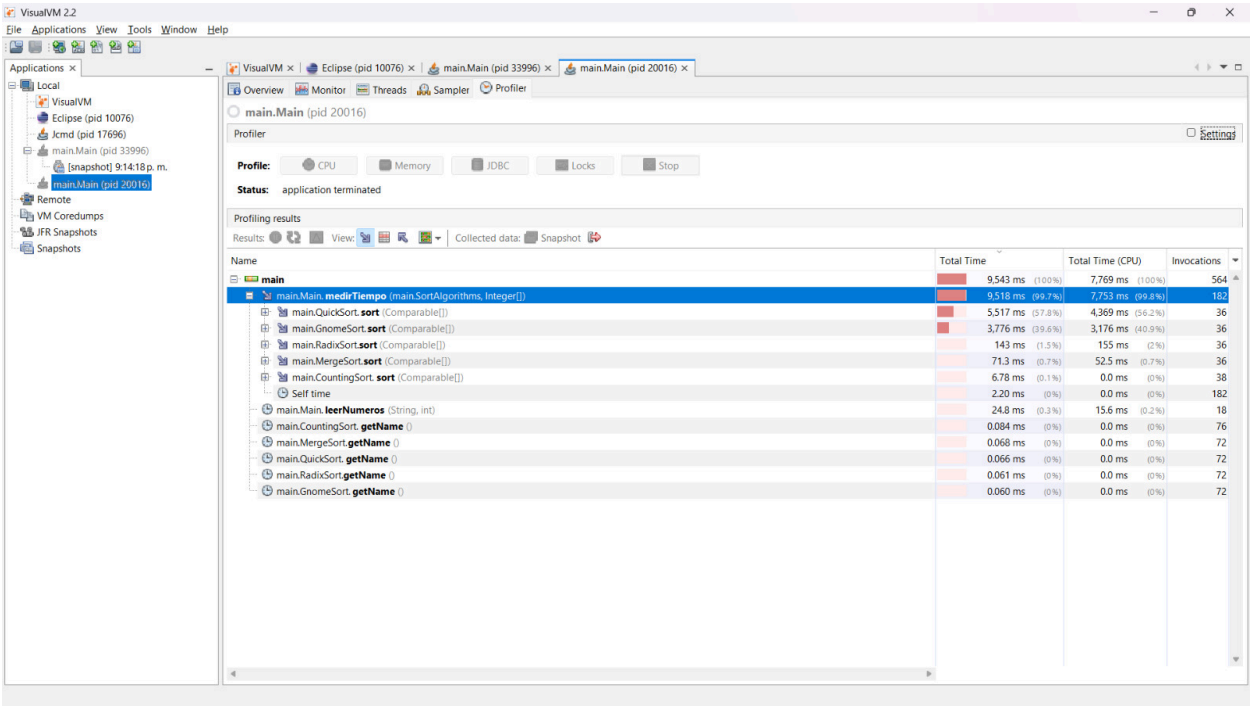
3. Uso de VisualVM como Profiler

Se utilizó VisualVM versión 2.2 para analizar el comportamiento interno de los algoritmos durante su ejecución. El procedimiento fue el siguiente:

1. Ejecutar la aplicación Java.
2. Abrir VisualVM y seleccionar el proceso correspondiente a la clase Main.
3. Activar el CPU Profiler.
4. Ejecutar los algoritmos de ordenamiento.
5. Analizar los resultados obtenidos en la tabla de métodos.

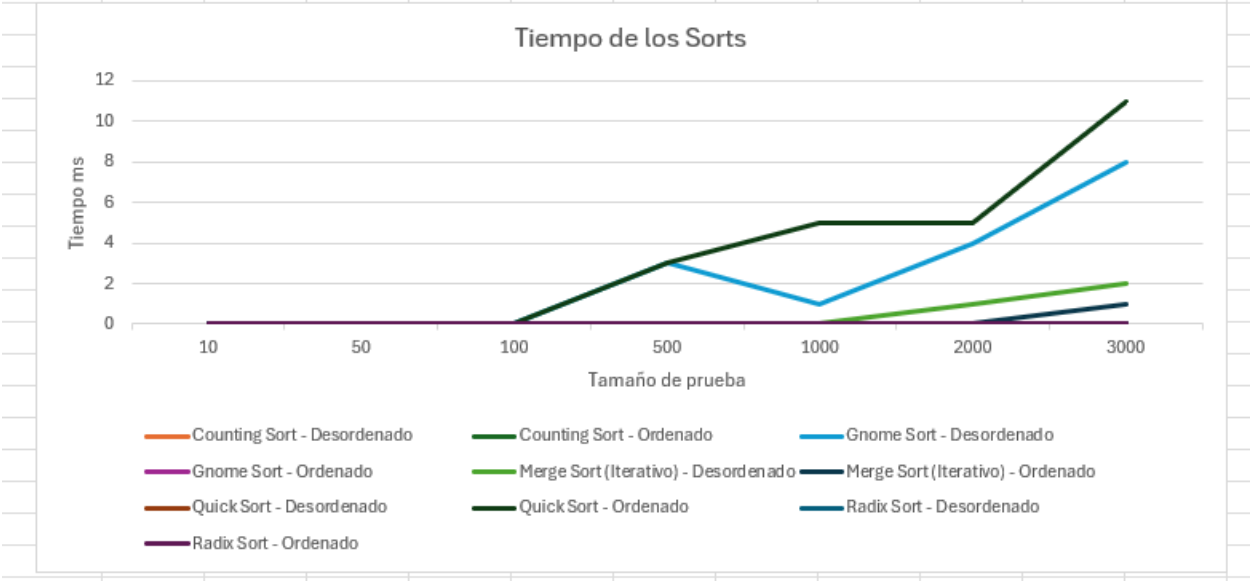
VisualVM permitió observar:

- El porcentaje de uso de CPU por método.
- El tiempo total de ejecución por algoritmo.
- Los métodos internos que consumieron mayor cantidad de recursos.

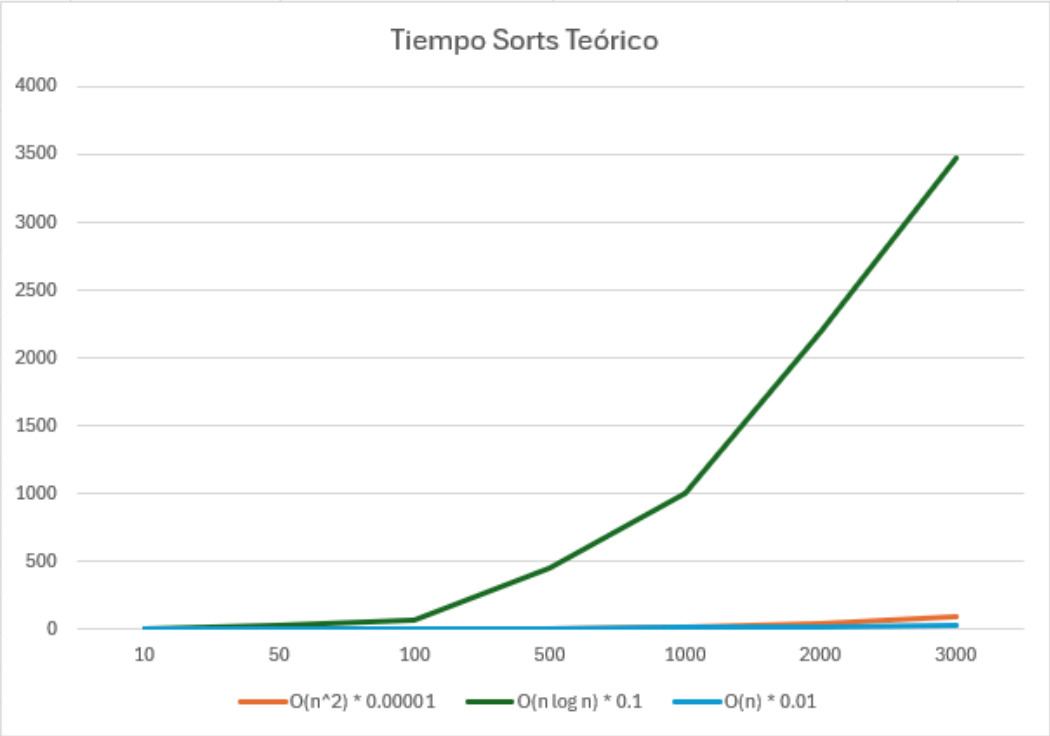


Excel:

Tamaño	Counting S	Counting S	Gnome So	Gnome So	Merge Sor	Merge Sor	Quick Sor	Quick Sor	Radix Sort	Radix Sort - Ordenado
10	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0
500	0	0	3	0	0	0	0	3	0	0
1000	0	0	1	0	0	0	0	5	0	0
2000	0	0	4	0	1	0	0	5	0	0
3000	0	0	8	0	2	1	0	11	0	0



Tamaño (i	O(n^2) * 0.00001	O(n log n) * 0.1	O(n) * 0.01
10	0.001	3.3	0.1
50	0.025	28.2	0.5
100	0.1	66.4	1
500	2.5	448.3	5
1000	10	996.6	10
2000	40	2193.1	20
3000	90	3468.9	30



Se puso una distinta escala a cada uno para que pudiera apreciar de alguna manera, ya que los tiempos son muchos mayores en O(n^2) que en O(n) y O(n log n).

Algoritmo	Complejida
Gnome Sort	O(n^2)
Merge Sort	O(n log n)
Quick Sort	O(n log n)
Radix Sort	O(n)
Counting Sort	O(n)