

UNIVERSIDAD DE GUADALAJARA
CUCEI



Actividad # __2__

Tema: Visualizador de Métodos de
Ordenamiento (GUI)

Estudiantes:

- Cuéllar Hernández Cinthya Sofía
- José Eduardo Valentin Gallardo

Análisis de algoritmos, D01, 2025B

Introducción

Los algoritmos de ordenamiento son fundamentales en las ciencias computacionales, ya que permiten organizar datos de manera eficiente permitiendo optimizar búsquedas y procesamiento. Esta aplicación implementa y compara visualmente cuatro diferentes tipos de algoritmos: Quick Sort, Merge Sort, Bubble Sort y Selección Sort, los cuales proporcionan una herramienta educativa para entender su funcionamiento y su rendimiento.

Objetivos

- Implementar una interfaz gráfica interactiva para visualizar diferentes tipos de algoritmos de ordenamiento.
- Comparar el rendimiento de cada algoritmo mediante la medición de tiempo.
- Identificar los puntos donde cada algoritmo se vuelve impráctico para un volumen grande de datos.

Desarrollo

Para este proyecto se desarrolló una aplicación en la biblioteca Tkinter que genera listas aleatorias de números enteros y permite ordenarlas mediante distintos algoritmos. Este programa permite visualizar en tiempo real el proceso de ordenamiento, ajustar parámetros como el tamaño de los datos y la velocidad de la animación. Puedes seleccionar 4 diferentes tipos de algoritmos y comparar resultados a través de gráficas. Los resultados de las pruebas se grafican usando Matplotlib. Se incluyó un control de velocidad de la animación, la opción de mezclar los datos y probar todos los algoritmos de forma automática.

Evidencias

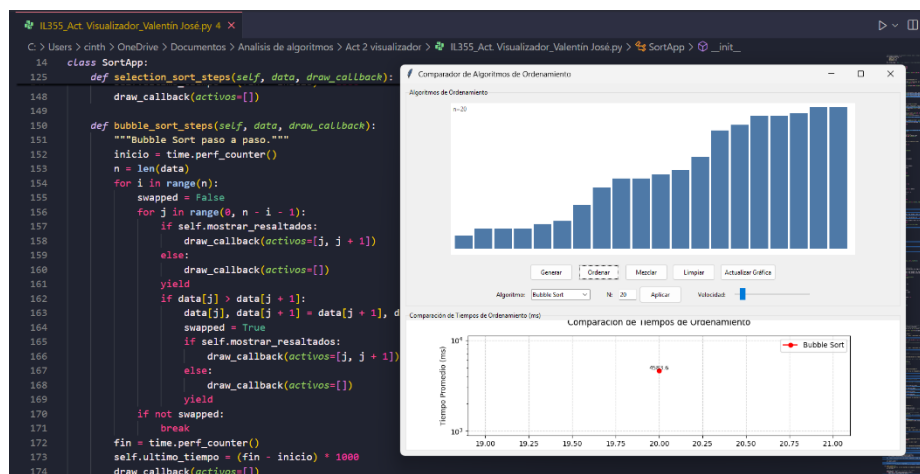
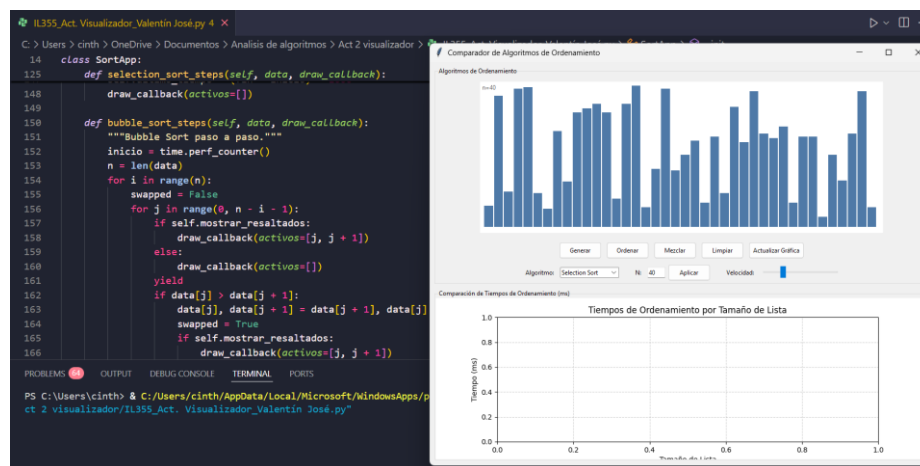


Tabla de tiempos aproximados:

N (tamaño)	Selection Sort	Bubble Sort	Merge Sort	Quick Sort
100	2-5	3-7	0.5-1	0.3-0.8
500	50-70	80-120	3-5	2-4
1000	200-300	350-500	7-10	5-8
2000	900-1200	1500-2000	15-22	10-15
5000	6000-8000	10000-14000	40-55	25-35

Cuándo cada método empieza a ser impráctico

- **Selection Sort:** Comienza a ser impráctico para $N > 1000$ elementos debido a su complejidad $O(n^2)$
- **Bubble Sort:** Impráctico para $N > 500$ elementos, siendo el menos eficiente de los cuatro
- **Merge Sort:** Se mantiene con un buen rendimiento hasta $N > 50,000$ elementos
- **Quick Sort:** El más eficiente de la práctica, manteniendo buen rendimiento incluso para $N > 100,000$ elementos.

Conclusiones

Sofía Cuéllar - Los algoritmos de ordenamiento más simples como Selection Sort y Bubble Sort son útiles para entender conceptos básicos de ordenamiento, pero no son adecuados para grandes volúmenes de datos. Por otro lado, Merge Sort y Quick Sort ofrecen un desempeño muy superior y son preferibles en aplicaciones reales. La visualización gráfica permitió comprobar el funcionamiento correcto de cada algoritmo y la comparación cuantitativa confirmó las diferencias en eficiencia.

Eduardo Valentín - Los resultados obtenidos en este trabajo permitieron confirmar que los algoritmos de ordenamiento como: Selection Sort y Bubble Sort, son de gran utilidad para comprender fundamentos de la computación, aunque presentan limitaciones importantes en listas de gran tamaño debido a su complejidad temporal. Por otro lado, los métodos Merge Sort y Quick Sort demostraron ser más eficientes y adecuados para aplicaciones prácticas, evidenciando la importancia de seleccionar el algoritmo correcto de acuerdo con la magnitud del problema. La implementación y visualización facilitaron la comprensión de estas diferencias y reforzaron la relevancia del análisis de algoritmos en la práctica.

Referencias

Universitat Oberta de Catalunya. (s.f.). *Interfaz gráfica de usuario*. Design Toolkit. Recuperado de <https://design-toolkit.recursos.uoc.edu/es/graphical-user-interface/>

Gómez, J. (2015). *Análisis comparativo de algoritmos de ordenamiento: Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Merge Sort, Heap Sort y Quick Sort*. MQL5. Recuperado <https://www.mql5.com/es/articles/3118>