



ITSRLL
INSTITUTO TECNOLÓGICO SUPERIOR
DE LA REGIÓN DE LOS LLANOS

Ingeniería Mecatrónica

SISTEMAS EMBEBIDOS CON APLICACIONES DE IOT

Enero – Junio 2025

M.C. Osbaldo Aragón Banderas

UNIDAD:

1	2	3	4	5
---	---	---	---	---

Actividad número: 2

Nombre de actividad:

Reporte de Programa en Jupyter Notebook –
Evaluación de Métodos de Ordenamiento

Actividad realizada por:

Melissa Gómez Rentería.

Guadalupe Victoria, Durango

Fecha de entrega:

12	02	2025
----	----	------

OBJETIVO GENERAL DE LA ACTIVIDAD

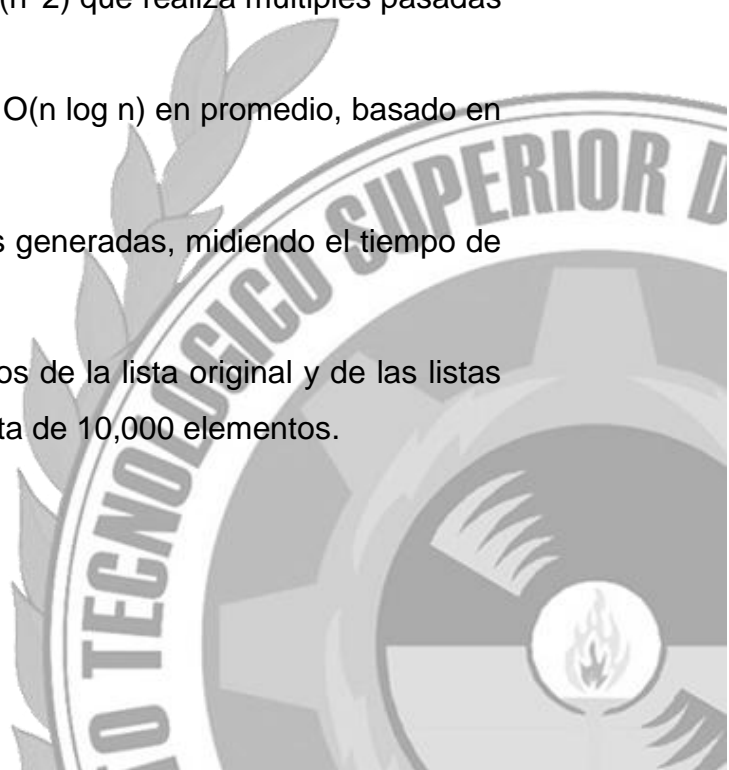
Utilizar Jupyter Notebook para implementar y evaluar dos métodos de ordenamiento (Burbuja y Quicksort) en Python, analizando su rendimiento. Esta actividad refuerza el conocimiento de algoritmos, fundamentales para la optimización del software en aplicaciones de robótica. Además, se busca que los estudiantes practiquen la creación y organización de repositorios en GitHub, con el fin de que sirvan como apoyo en su currículo profesional.

DESARROLLO DE LA ACTIVIDAD

En este programa se implementa y se compara la eficiencia de dos algoritmos de ordenamiento: Burbuja y Quicksort. Para ello, se ejecutan ambos algoritmos en listas de hasta 10,000 elementos aleatorios, esto en listas de 100, 500, 1,000, 5,000 y 10,000 elementos, para ello se mide el tiempo de ejecución, y con ello, se analiza el rendimiento de cada uno de los métodos de ordenamiento.

PROCEDIMIENTO

1. Se generaron listas de 100, 500, 1,000, 5,000 y 10,000 números aleatorios entre 0 y 10,000.
2. Se implementaron los algoritmos de ordenamiento:
 - **Burbuja:** Algoritmo de complejidad $O(n^2)$ que realiza múltiples pasadas sobre la lista.
 - **Quicksort:** Algoritmo de complejidad $O(n \log n)$ en promedio, basado en dividir y conquistar.
3. Se ejecutaron los algoritmos en las listas generadas, midiendo el tiempo de ejecución.
4. Se imprimieron los primeros 10 elementos de la lista original y de las listas ordenadas con cada algoritmo para la lista de 10,000 elementos.
5. Se analizaron los resultados obtenidos.



RESULTADOS

A continuación, se presentan los tiempos de ejecución obtenidos para las listas generadas aleatoriamente por el programa:



Figura 1: Comparación de tiempos de ordenamiento.

CONCLUSIONES

- El algoritmo de Burbuja demostró ser significativamente más lento debido a su complejidad $O(n^2)$.
- Quicksort presentó una ejecución mucho más eficiente gracias a su estructura basada en particiones.
- Para conjuntos de datos grandes, Quicksort es claramente la mejor opción, mientras que Burbuja solo es útil para listas pequeñas o cuando se requiere una implementación simple.

Las pruebas realizadas confirman la ventaja de Quicksort en términos de tiempo de ejecución y escalabilidad, reforzando la importancia de elegir el algoritmo adecuado según el contexto del problema.