



**CENTRO UNIVERSITARIO DE  
CIENCIAS EXACTAS E INGENIERÍAS**



**UNIVERSIDAD DE  
GUADALAJARA**  
Red Universitaria de Jalisco

***División de Tecnologías para la Integración Ciber-Humana***

***Departamento de Ciencias Computacionales***

***Carrera: Ingeniería en Computación***

Análisis de Algoritmos

Actividad 2

Visualizador de Métodos de Ordenamiento

Profesor: Lopez Arce Delgado Jorge Ernesto

Integrantes del Equipo:

Gutierrez Vazquez Axel 220575328

Quintero Arreola Laura Vanessa 220577347

Sección: D06

Fecha de entrega: 10 de septiembre de 2025

## Introducción

---

En esta práctica se desarrolló una interfaz gráfica con python y tkinter para visualizar el funcionamiento de diferentes algoritmos de ordenamiento: select sort, merge sort, bubble sort y quick sort. El programa permite generar listas de datos aleatorios, observar el proceso de ordenamiento paso a paso mediante barras gráficas, y registrar automáticamente el tiempo de ejecución de cada método. De esta forma, se facilita tanto la comparación visual como el análisis de rendimiento de los algoritmos.

## Objetivos

---

Implementar y comparar distintos métodos de ordenamiento (de la actividad pasada) en una GUI con Tkinter, permitiendo:

- Seleccionar el algoritmo desde un dropdown.
- Generar datos, Ordenar, Mezclar (shuffle), y Limpiar resaltado con botones.
- Modificar el número de barras desde un Entry.
- Ajustar la velocidad de la animación con un Scale.

## Desarrollo

---

Para desarrollar esta práctica tomamos como base la participación número dos que realizamos en clase y tal como venía en el documento de explicación sobre el código la clave para poder desarrollar esta actividad más compleja era entender cómo funcionaba yield, pues este nos permitía generar la animación de las barras acomodándose, porque pausaba la animación hasta que se volviera a llamar a la misma función haciendo de esta forma una especie de “salto” y de esta forma hacer una animación fluida en tiempo real de ejecución del código, además otro detalle importante fue adaptar los métodos de ordenamiento que ya teníamos previamente hechos a la forma en la que funcionaba la animación de las barras.

Se realizaron pruebas con listas de diferentes tamaños y se registraron tiempos aproximados de ejecución de cada método. por ejemplo, para  $N=20$ , los tiempos fueron de 0.015s a 0.020s. pero para  $N=40$  los tiempos aumentaron a 0.040s a 0.080s, ya que el tamaño de la lista era mas grande, además, Merge y Quick se mantuvieron eficientes durante todas las pruebas, siendo los primeros en terminar el ordenamiento son gran ventaja en comparación a bubble y select. Se observó que Bubble sort y select sort se vuelven imprácticos cuando la lista contiene más de 40 o 50 elementos, mientras

que mergesort y quicksort mantienen un rendimiento adecuado incluso en listas grandes, aunque quicksort puede ser más lento que merge sort en listas muy pequeñas.

Durante la práctica se tomaron capturas de la GUI en ejecución, mostrando las barras moviéndose mientras se ordenaron los datos y la actualización de la tabla con los tiempos registrados. esto nos permitió demostrar el correcto funcionamiento de los algoritmos y observar la eficiencia de cada uno en diferentes pruebas.

Evidencia



## Conclusiones Individuales

---

Gutierrez Vazquez Axel

Al realizar la práctica, pude comprender de manera más clara el funcionamiento de los distintos métodos de ordenamiento. Observé que Bubble Sort es el más lento de los utilizados, mientras que Quicksort, aunque generalmente eficiente, puede ser más lento en listas pequeñas debido a la complejidad de su proceso. Asimismo, reforcé el uso de yield, que permite pausar y retomar funciones, lo cual fue muy útil para visualizar paso a paso cómo se realiza el ordenamiento de los datos.

Quintero Arreola Laura Vanessa

En esta práctica creo que lo que pudo haber sido más complicado fue añadir varias funciones a la ventana de la GUI, es decir, la parte de agregar varios botones para limpiar, mezclar, generar, aplicar, etcétera. A grandes rasgos creo que esta práctica también fue en pro de hacer que nos familiaricemos con nuevas herramientas para una interfaz gráfica, además con esta actividad creo que fue más fácil identificar de manera visual que método de ordenamiento es mejor y bajo qué condiciones, pues dependiendo mucho del caso es cuanto tardan así que son útiles para ciertas aplicaciones.

## Bibliografía

---

Python 3.13 documentation. Python Documentation. Recuperado el 11 de septiembre de 2025, de <https://docs.python.org/3/>