

Act. Visualizador de métodos de Ordenamiento



UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias
Exactas e Ingenierías

División de Tecnologías para la
Integración Ciber-Humana

Nombre del curso: Análisis de Algoritmos

Nombre del equipo:

- Yashima Rodríguez Satomi
- Humberto de Jesús Peña Dueñas

Carrera: Ingeniería en Computación

Sección: D01

Título de la actividad: Act. Visualizador de
métodos de Ordenamiento

Fecha de elaboración: 03/09/2025

Introducción

En esta práctica se implementaron cuatro algoritmos de ordenamiento para su análisis:

SelectionSort: Recorre la lista buscando el elemento más pequeño y lo intercambia cuando lo encuentra. **BubbleSort:** Compara elementos adyacentes, intercambiándolos si están desordenados, y repite el proceso hasta ordenar toda la lista. **MergeSort:** Usa el método divide y vencerás, dividiendo la lista en sublistas que ordena recursivamente para luego combinarlas ordenadas. **QuickSort:** También divide y vence, pero elige un pivote para reorganizar los elementos y ordenar la lista de forma recursiva.

Además, se desarrolló una interfaz gráfica (GUI) para visualizar el comportamiento, eficiencia y adaptabilidad de cada algoritmo según el tamaño de la lista.

Objetivos

Implementar y comparar distintos métodos de ordenamiento 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 el desarrollo de esta actividad utilizamos como base el algoritmo que se nos dió en clase, decidimos dividirlo para que el código fuera más limpio y organizado, el archivo de los algoritmos, el de la gráfica, la interfaz y el main.

En el apartado de los algoritmos se utilizaron los previamente proporcionados por el profesor y se hicieron los respectivos draw_callback en los momentos necesarios para resaltar.

En el archivo de la gráfica, decidimos hacer una objeto con sus respectivos atributos y métodos, los cuáles de manera general hacen lo siguiente: **generar** (crea y dibuja una nueva lista con N barras.), **ordenar** (ejecuta la animación del algoritmo elegido.), **mezclar** (barajea la lista actual.), **detener** (interrumpe el ordenamiento en curso.), **resaltar** (opción de activar/desactivar resaltado de elementos activos.), **cambiar N** (el usuario ingresa un valor y genera nueva lista con esa cantidad.), **velocidad** (ajusta la rapidez de la animación.)

Por último en la interfaz se hizo todo el modelo visual, acomodo de botones y conexión con la lógica de la gráfica.

Conclusiones

La práctica nos resultó sencilla y logramos hacer el código de manera que ambos comprendieramos qué era lo que se quería lograr. Trabajar sobre un código base fue de gran ayuda, ya que antes de realizar modificaciones era necesario comprender el funcionamiento de cada una de sus partes para identificar dónde y cómo realizar los cambios necesarios para obtener el resultado esperado. Además debíamos comprender y aprovechar lo aportado por el otro para continuar y mejorar el código. Fue bastante interesante trabajar de esta manera y considero que no nos fue para nada complicado.

Referencias

Python Software Foundation. (2025). *tkinter — Python interface to Tcl/Tk*. In Python 3 Documentation. <https://docs.python.org/3/library/tkinter.html>

Geremias Baudino. (2023). MetodosDeOrdenamiento. [Código fuente]. GitHub. <https://github.com/gbaudino/MetodosDeOrdenamiento?tab=readme-ov-file&authuser=0>

Python Software Foundation. (2025). *random — Generate pseudo-random numbers*. In Python 3 Documentation. <https://docs.python.org/3/library/random.html>