UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías C.U.C.E.I.

Análisis de algoritmos

Actividad 1

Arroyo Moreno Elizabeth.

Mtro. Jorge Ernesto López Arce Delgado.

20/08/2025

Descripción de la actividad.

El objetivo de esta práctica fue comparar el rendimiento de los algoritmos de búsqueda lineal y búsqueda binaria en listas de diferentes tamaños.

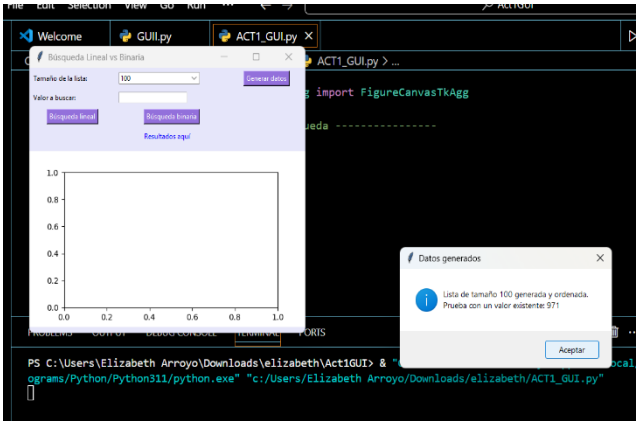
Para ello, se implementó una interfaz gráfica en Python con Tkinter, que permite:

- Generar listas ordenadas de tamaño 100, 1000, 10,000 y 100,000 elementos.
- Ingresar un valor a buscar (se selecciona un número que garantice estar en la lista).
- Ejecutar búsqueda lineal y búsqueda binaria, midiendo el tiempo en milisegundos.
- Mostrar los resultados en una gráfica comparativa.

Cada experimento fue repetido 5 veces para calcular el tiempo promedio de ejecución y reducir la variabilidad.

Tamaño de lista	Búsqueda Lineal (ms)	Búsqueda Binaria (ms)
100	0.0032	0.0012
1,000	0.0060	0.0023
10,000	0.2560	0.1591
100,000	1.2317	0.0058

La interfaz funcionando:

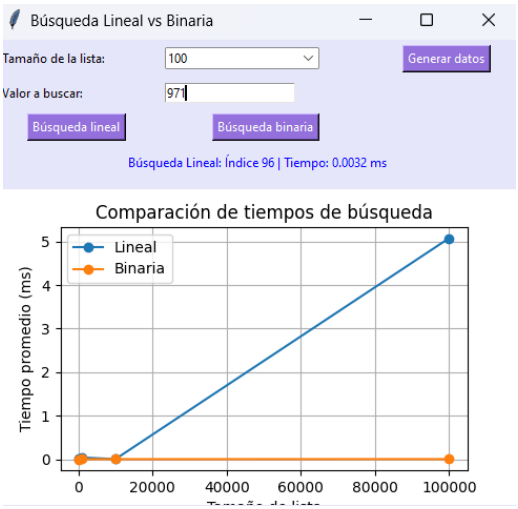


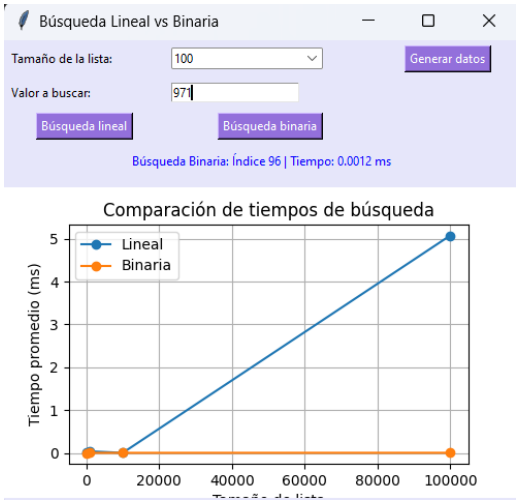
Para su uso primero se selecciona el tamaño de la lista y el programa genera datos aleatorios ordenados. Se puede ingresar un valor a buscar y el programa ejecuta la búsqueda lineal o binaria. Con la librería time, se mide el tiempo que tarda cada búsqueda en milisegundos, repitiéndose varias veces para obtener un promedio más confiable.

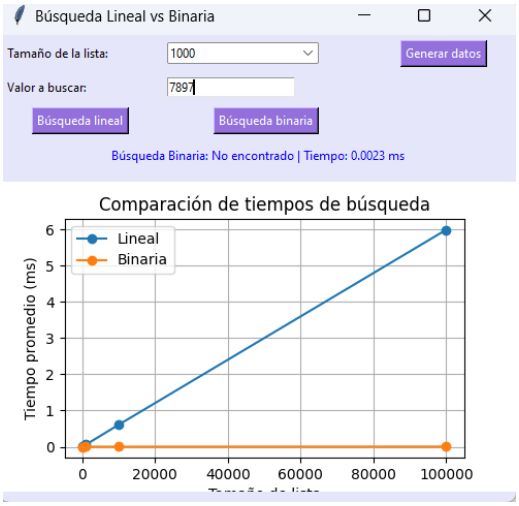
Los resultados se muestran en pantalla y en una gráfica comparativa generada con matplotlib.

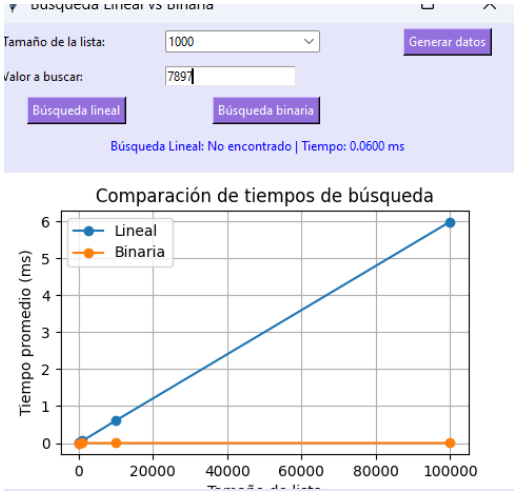
Gráficas comparativas

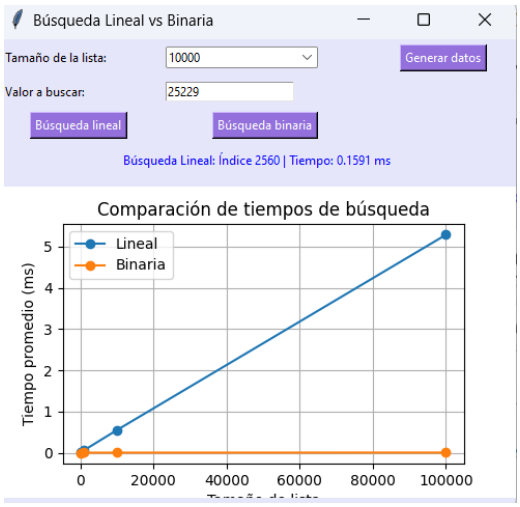
Las siguientes imágenes muestran los resultados de las gráficas generadas por la aplicación:

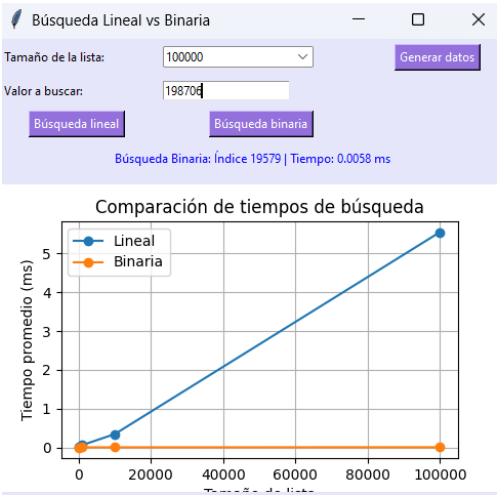


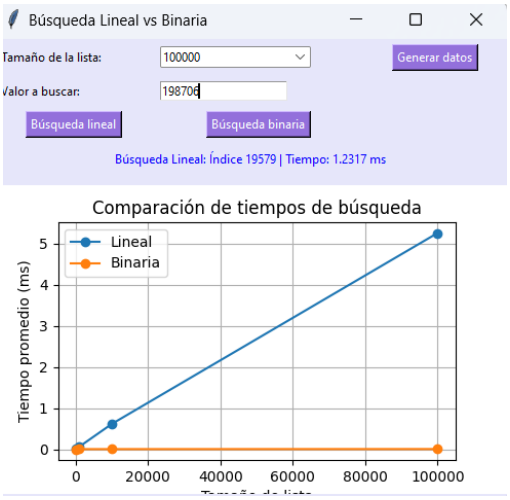












Conclusión

Con esta actividad descubrí nuevas herramientas para la creación de la interfaz, como Tkinter para diseñar ventanas y botones, NumPy para generar datos aleatorios y ordenados, Time para medir los tiempos de ejecución y Matplotlib para graficar y comparar visualmente los resultados.

Asimismo, pude analizar el comportamiento de los algoritmos de búsqueda. En el caso de la búsqueda lineal, observé que presenta un crecimiento proporcional al tamaño de la lista, ya que revisa elemento por elemento hasta encontrar el valor buscado o llegar al final, lo que implica una complejidad de O(n).

La búsqueda binaria, en cambio, mantiene un tiempo de ejecución casi constante incluso al incrementar el tamaño de la lista. Esto se debe a que divide el espacio de búsqueda en mitades sucesivas, reduciendo drásticamente el número de comparaciones. Su complejidad es O(log n).

En listas pequeñas (100 elementos), la diferencia de tiempos no es muy significativa; sin embargo, a medida que el tamaño crece (100,000 elementos), la búsqueda binaria resulta miles de veces más rápida que la lineal, demostrando la importancia de elegir el algoritmo adecuado. Aunque la búsqueda lineal es simple de implementar, la binaria resulta mucho más eficiente cuando se trabaja con grandes volúmenes de datos ordenados.