

Análisis de Algoritmos

Sección: D01

Maestro: JORGE ERNESTO LOPEZ ARCE DELGADO

Gerardo Dagnino Zatarain

223992337

17 de agosto de 2025

## Experimentos

Para poder obtener el tiempo promedio de los métodos de búsqueda lineal y binaria, se generará una lista de números enteros de entre 0 y 100000, la lista variará de tamaños, estas siendo 100, 1000, 10000 y 100000, se repetirá la búsqueda 5 veces por cada combinación. Se observará la lista generada y se elegirán cinco números para posteriormente obtener el promedio. La lista está previamente ordenada en todos los casos. Esto debido a que el método de búsqueda binaria no funcionará al estar no ordenada.

Se usaron los siguientes números para hacer cada prueba

100: 871, 32454, 51545, 74022, 99202 (5 números)

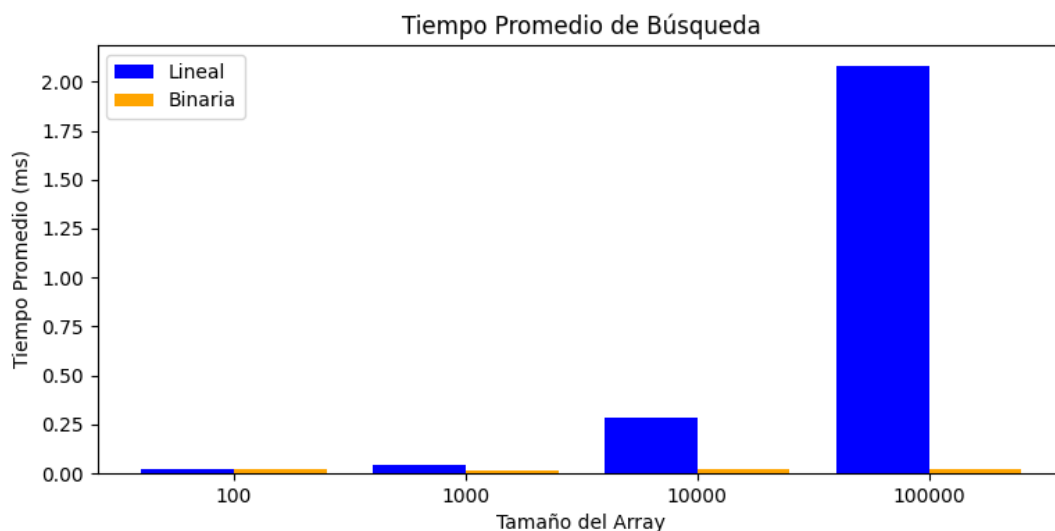
1000: 150, 23823, 54082, 88293, 99998 (5 números)

10000: 1, 929, 13648, 65437, 99548 (5 números)

100000: 3, 1903, 34529, 50123, 99957 (5 números)

## Resultados

-Grafica de Barras



## -Resultados texto

Resultados Lineal (Tiempo promedio)	Resultados Binaria (Tiempo promedio)
100: 0.022 ms, 5 pruebas	100: 0.019 ms, 5 pruebas
1,000: 0.046 ms, 5 pruebas	1,000: 0.015 ms, 5 pruebas
10,000: 0.281 ms, 5 pruebas	10,000: 0.023 ms, 5 pruebas
100,000: 2.083 ms, 5 pruebas	100,000: 0.020 ms, 5 pruebas

## Conclusiones

Al momento de terminar las pruebas y viendo los resultados, es bastante obvio que la búsqueda binaria muchísimo más efectiva al momento de buscar en datasets grandes, por ejemplo, en la lista de 100,000 números, hay una diferencia de más de 2 milisegundo en las 5 pruebas entre la búsqueda binaria y lineal.

Pero lo que no se ve en los resultados finales es que al momento de hacer las pruebas, si ingresaba un número no muy grande, estando cerca del inicio de la lista, la búsqueda lineal lo encontraba más rápido que la búsqueda binaria, esto era más claro en el dataset de 10,000 y 100,000, esto se puede explicar con el hecho de que la búsqueda lineal empieza desde el inicio y lo encuentra en el inicio, que es en donde está.