

Búsqueda con GUI

PRACTICA 1

OSVEN JACIEL OCAMPO AGUILA

Actividad 1

En esta primera practica realice un programa en Python con interfaz de usuario. El programa que hice es un generador de números, el cual permite ingresar la cantidad deseada de números, así como buscar dentro de la lista de números y saber si existe un elemento en específico. Se pueden realizar búsquedas iterativas ya sean lineales o binarias. Al momento de realizar una búsqueda, aparece un texto con los detalles de la misma, como el tamaño de la lista, el tiempo que tardo en ejecutar y si se encontró o no el numero buscado. No se permite realizar búsquedas sin que se haya creado una lista anteriormente y no permite realizar búsquedas de números decimales o caracteres.

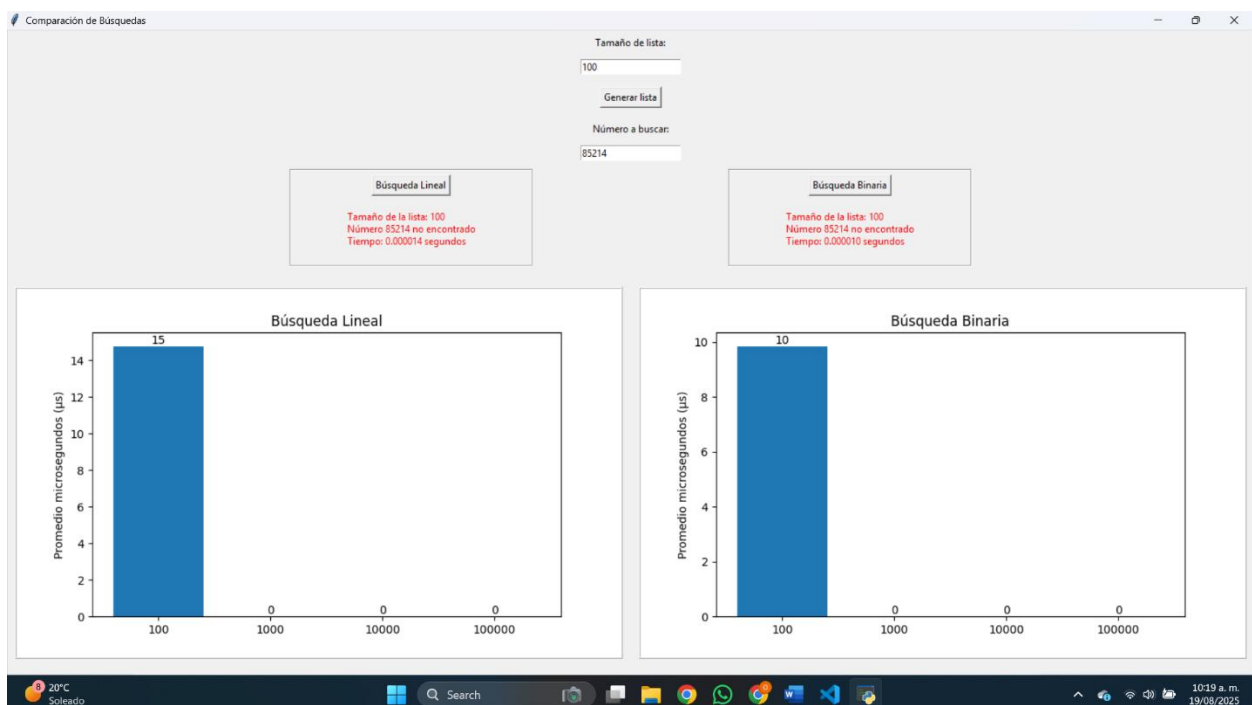
Experimentos

En este programa realice 20 búsquedas: 5 en una lista de 100 elementos, 5 en una de 1,000, 5 en una de 10,000 y 5 en una de 100,000. Para cada 5 búsqueda escogí números distintos usando la misma lista. Una nueva lista se generaba al cambiar el tamaño de esta. Así mismo, medi el tiempo que tardo cada una de las búsquedas y esto lo grafique, sacando un promedio y realizando actualizaciones.

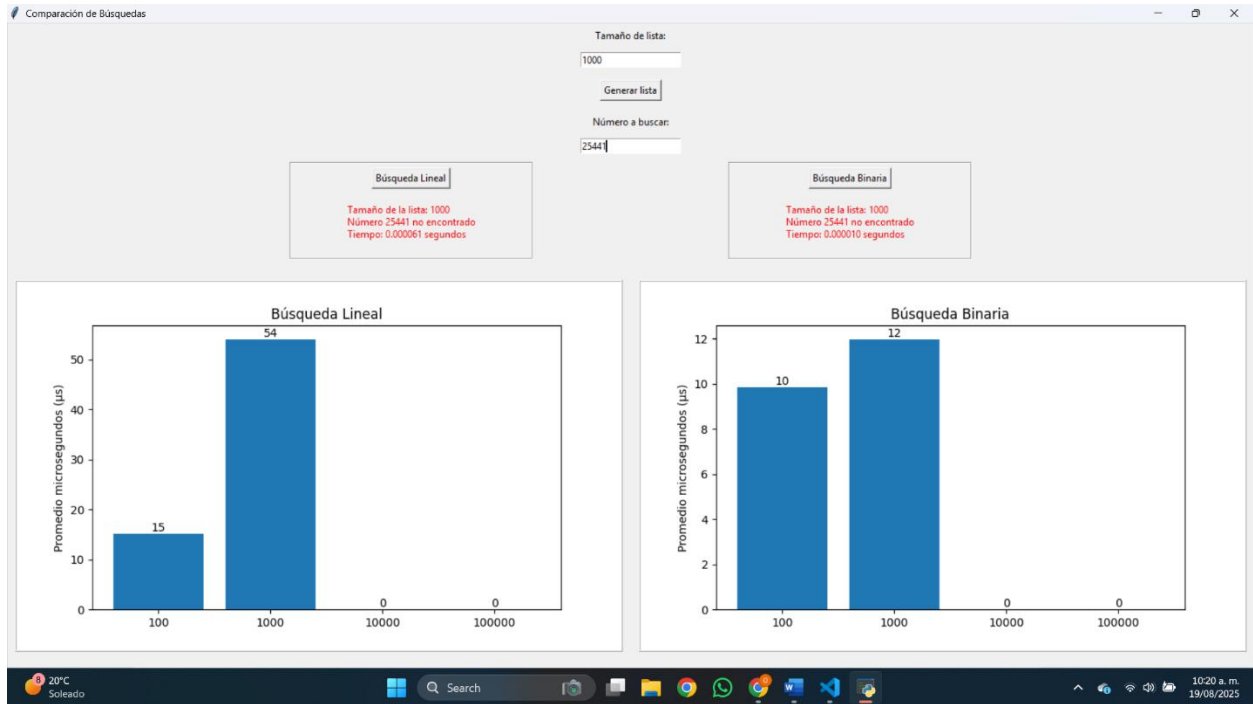
Resultados

Lista 100 elementos

Aquí cree la primera lista con 100 elementos y realice las 5 búsquedas usando los dos métodos. Las cantidades están en microsegundos. Encima de cada columna aparecerá el promedio de las 5 búsquedas. Así lo hice con cada una de las 4 listas.

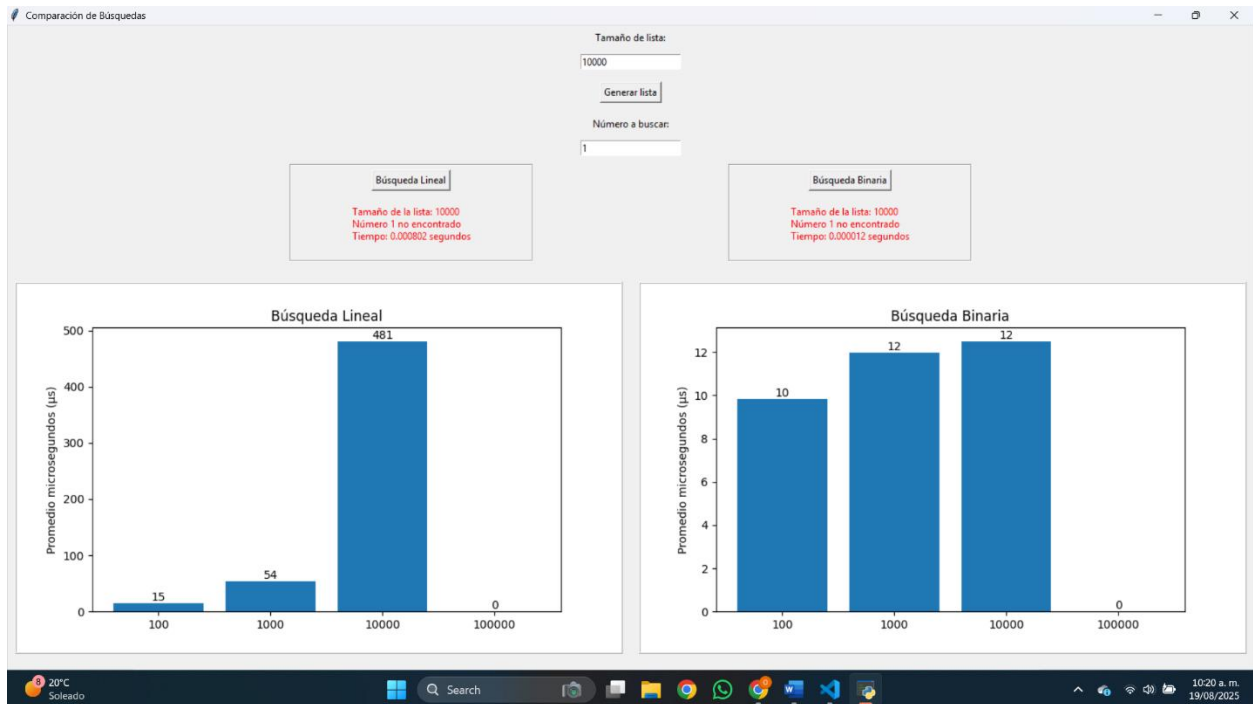


Lista 1,000 elementos



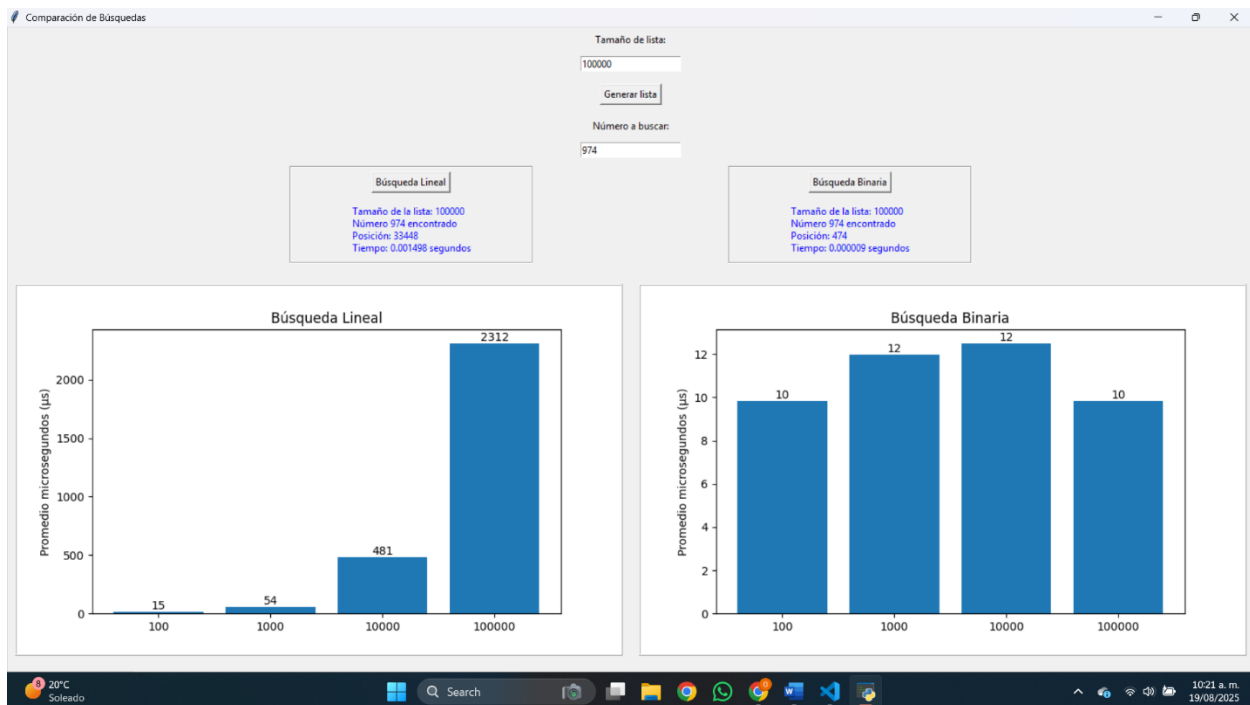
Lista 10,000 elementos

Cuando un número no se encuentra, el texto sale en color rojo.



Lista 100,000 elementos

Cuando un número se encuentra, el texto sale en color azul.



Conclusiones

Después de realizar las 20 búsquedas por medio de los dos algoritmos, es claro que usar una búsqueda binaria será machismo mas eficaz comparado a la lineal, sobre todo cuando usamos cantidades muy grandes. Podemos observar que, en la búsqueda lineal, el tiempo de ejecución iba en aumento al momento que incrementábamos el numero de elementos en la lista; al contrario, en la búsqueda binaria de hecho decremento el tiempo de ejecución en la lista con más datos, así como se mantenía en un rango similar.