

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

INGENIERO EN COMPUTACIÓN

ANÁLISIS DE ALGORITMOS

MACIEL VARGAS OSWALDO DANIEL

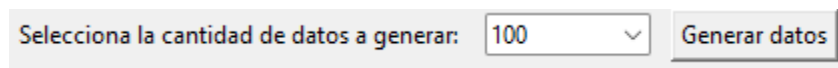
Actividad #01

Búsqueda con GUI

I.Introducción

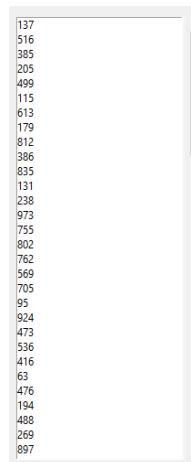
En esta actividad tuvimos la tarea de realizar una GUI en Python, la cual nos ayudara a visualizar mediante una gráfica, el promedio de tiempo que tarda nuestro equipo en realizar dos algoritmos: Búsqueda lineal y búsqueda binaria.

Empezaremos explicando las distintas partes de nuestra GUI. Empezamos con lo principal, la cual es la sección donde se generan ciertos datos de manera aleatoria y se almacenan en una lista la cual es la que se utiliza en los algoritmos. En esta parte encontraremos un menú desplegable donde podemos seleccionar entre cuatro distintas cantidades de datos a generar: 100, 1000, 10000 y 100000. Ya que seleccionemos el tamaño, presionamos el botón “Generar datos”:



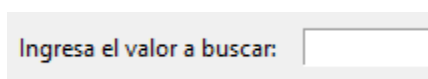
Selecciona la cantidad de datos a generar: 100 ▼ Generar datos

Otra parte interesante de nuestro programa es que en el lado izquierdo se muestran los datos generados por nuestro programa al presionar el botón correspondiente:



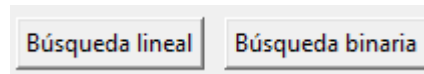
137
516
385
205
499
115
613
179
812
386
835
131
238
973
755
802
762
569
705
95
924
473
536
416
63
476
194
488
269
897

El programa no puede funcionar hasta que no se generen estos datos. Después de seleccionar los datos y generarlos (aunque no es necesario generarlos), encontramos una entrada de datos donde podemos ingresar el valor entero a buscar mediante los algoritmos en nuestras listas de datos aleatorios:

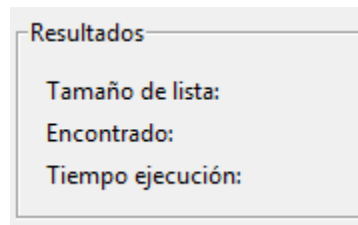


Ingresa el valor a buscar:

Ya que generemos los datos (y no antes), y haber ingresado el valor a buscar, se habilitarán los botones para usar el algoritmo de búsqueda binaria o búsqueda lineal, una vez que presionemos estos se ejecutará el algoritmo correspondiente 5 veces y se hará un promedio de ese tiempo:



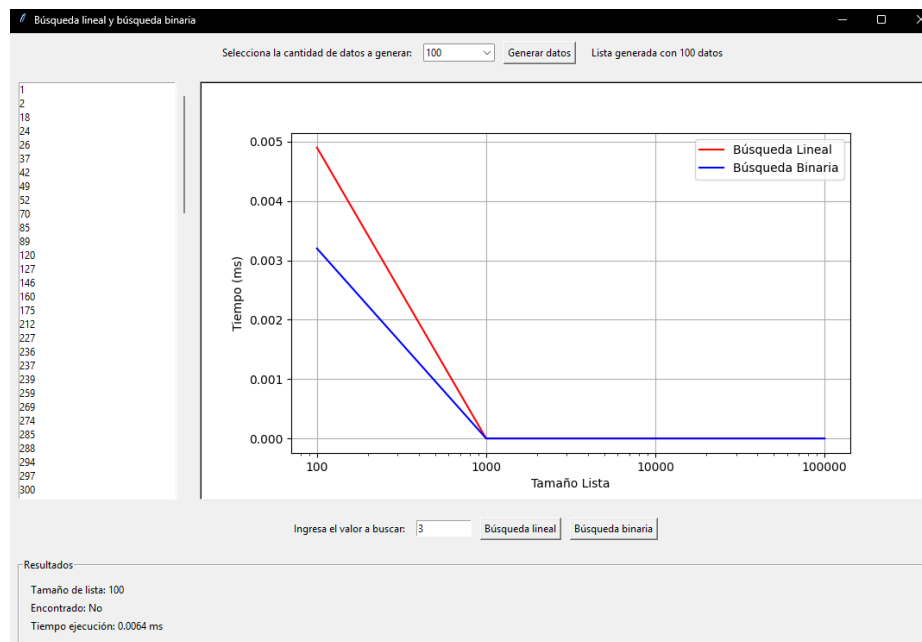
Podemos encontrar un área donde se presentan la cantidad de datos generados, si se encontró el elemento o no y el tiempo promedio:



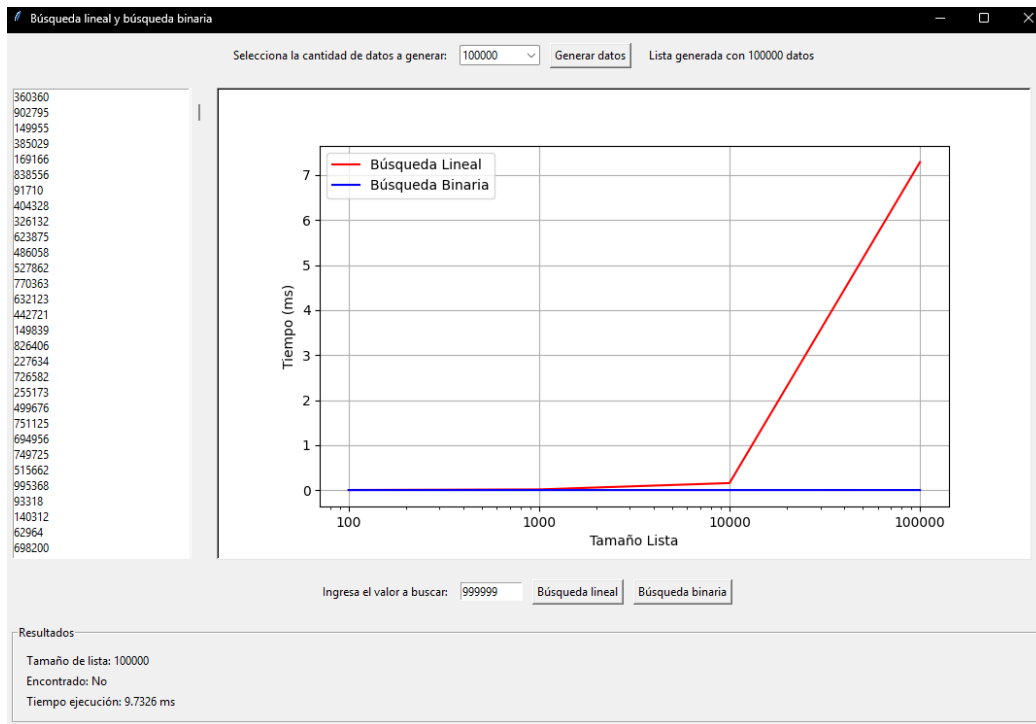
Al final encontramos la gráfica donde se muestran la relación de la cantidad de datos (eje x) y el tiempo promedio de las 5 iteraciones del algoritmo seleccionado (eje y).

II. Gráfica y experimentos

Ahora mostraremos el funcionamiento de nuestro programa. Primero haremos una prueba de búsqueda lineal y binaria con 100 datos, buscando un valor pequeño:



Realicé experimentos para las cantidades de datos 1000 y 10000, estos se visualizarán en la siguiente gráfica. El experimento realmente importante es en el peor de los casos, haremos un experimento con 100000 datos, usando los dos algoritmos y buscando un valor muy grande:



III.Conclusiones

En lo personal fue muy difícil esta actividad, el principal motivo fue que nunca había realizado un trabajo en Python, a eso se le suma que nunca había realizado un programa con una interfaz gráfica tan completa. Esta actividad implicó aprender un nuevo lenguaje de programación y a parte realiza un trabajo complejo, pero definitivamente fue una experiencia muy gratificante, al terminar el programa sentí que había aprendido mucho.

Muchas veces no dimensionamos lo que implica que nuestro equipo realice un algoritmo, sobre todo por las distintas variaciones de tiempo que puede tener la misma operación, con los mismos datos, en el mismo equipo. Espero encontrarme con más desafíos como este programa.