

# Estructura de Datos (2021-1)

## Laboratorio 2

Francisco Ignacio Salinas Alarcón  
frsalinas@udec.cl  
Matrícula: 2015442662

April 5, 2021

### Ejercicios:

1. Completar el fichero "Busqueda.cpp", implementando 2 métodos de búsqueda, el método de búsqueda "lineal" y el método "binariaRecursiva" o "binariaIterativa". Éstos deben retornar la posición del elemento consultado o -1 si no se encuentra el elemento en el arreglo.

**Desarrollo:** El desarrollo de este laboratorio fue realizado en Python 3.9.

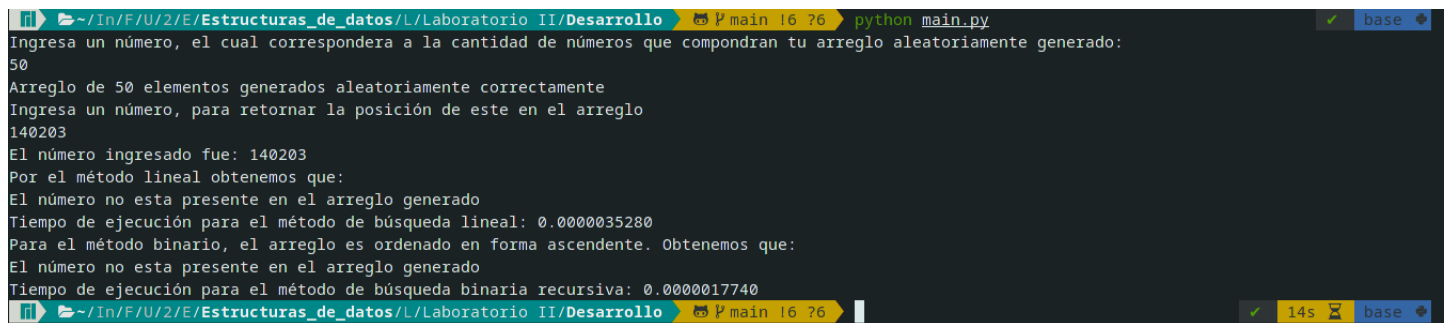
Se adjunta código en la carpeta de entrega.

El programa fue escrito en dos script, main.py como el script principal y functions.py el cual contiene funciones que se llaman y ejecutan con main.py. Deben ser ejecutados en el mismo directorio.

El programa usa el método de búsqueda lineal y binario recursivo para encontrar el índice de un número ingresado por el usuario en un arreglo generado aleatoriamente según la cantidad de número que tenga, número que debe ser ingresado por el usuario.

Si el número no existe en el arreglo, el programa imprime en pantalla que no existe, de forma contraria, si existe, el programa dará el índice correspondiente al número en el arreglo.

Imagen de muestra del funcionamiento del programa:



```
~/In/F/U/2/E/Estructuras_de_datos/L/Laboratorio II/Desarrollo > python main.py
Ingresa un número, el cual correspondera a la cantidad de números que compondran tu arreglo aleatoriamente generado:
50
Arreglo de 50 elementos generados aleatoriamente correctamente
Ingresa un número, para retornar la posición de este en el arreglo
140203
El número ingresado fue: 140203
Por el método lineal obtenemos que:
El número no esta presente en el arreglo generado
Tiempo de ejecución para el método de búsqueda lineal: 0.0000035280
Para el método binario, el arreglo es ordenado en forma ascendente. Obtenemos que:
El número no esta presente en el arreglo generado
Tiempo de ejecución para el método de búsqueda binaria recursiva: 0.0000017740
```

En este caso el número ingresado no se encuentra en el arreglo.

2. Utilizar el main proporcionado para ejecutar las implementaciones de búsqueda y medir sus tiempos de ejecución para distintos tamaños de entrada. Escribir los resultados experimentales en una tabla y crear un gráfico comparativo de los 2 algoritmos, cuyo formato debe ser: número de elementos (en el eje X) vs. tiempo (en el eje Y).

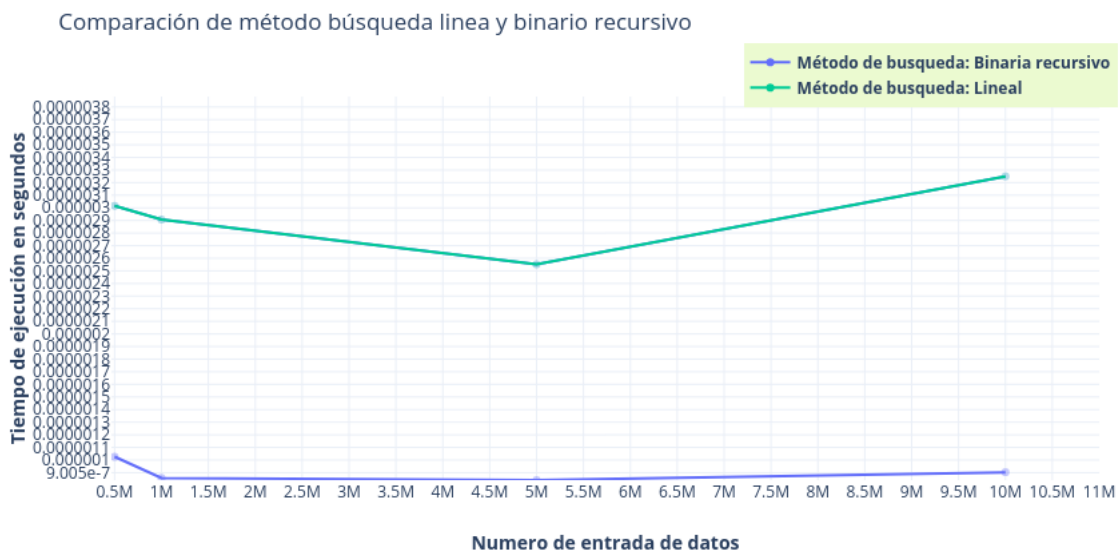
**Desarrollo:** En la captura anterior podemos ver que el programa, para cada método de búsqueda imprime en pantalla cuanto demora la ejecución.

Realice un gráfico del comportamiento para cada búsqueda, ingresando diferentes tamaños de arreglos:

**ERROR:** En la ultima revisión me percate que para el primer gráfico, la recta verde esta asociada al legend de búsqueda lineal, pero es al revés. La linea verde corresponde al método de búsqueda binario y la morada al método de búsqueda lineal.



Observamos el comportamiento para 10, 50, 100, 300, 800 y 1000 datos que componen el arreglo. También hice entradas para una escala mucho mayor:



Podemos observar la escala para datos de entrada de 0.5 Millones, 1M, 5M y 10 millones.

**3. Respecto a los algoritmos implementados, ¿Qué algoritmo es mejor? ¿Cuál es peor? Justifique según los datos obtenidos en la pregunta 2.**

**Desarrollo:** Podemos observar mediante los gráficos que para gran y pequeña escala de datos que compongan el arreglo, el método de búsqueda binario es mucho mas óptimo en tiempo. Si bien los tiempos son poco perceptibles para el usuario, podemos ver la tendencia que a una mayor escala, la búsqueda línea se aleja mucho en los tiempos en comparación con la búsqueda binaria.

Por lo cual el algoritmo de búsqueda binaria recursiva es el mucho mejor método tanto para pequeñas y grandes cantidades de datos que componen al arreglo.