

## BUSQUEDA BINARIA DADA POR LA REDUCCION DE LA MITAD DE ELEMENTOS

Jose Andres Carvajal  
Universidad Eafit  
Colombia  
jacarvajab@eafit.edu.co

Brian Morales  
Universidad Eafit  
Colombia  
bfmoralesa@eafit.edu.co

Alejandra Ossa  
Universidad Eafit  
Colombia  
aossay@eafit.edu.co

Mauricio Toro  
Universidad Eafit  
Colombia  
mtorobe@eafit.edu.co

**NOTA DEL DOCENTE:** Para ampliar información sobre los requerimientos aquí descritos, consulten la “Guía para la realización del Proyecto Final de Estructura de Datos 1” que se entrega. Al final: 1. **Borrar este texto escrito en rojo**, 2. **Adecuar los espacios de los textos**, 3. **Cambiar el color de los textos a negro**. Consideren además que:

**Textos en negro** = lo que deben hacer en la entrega 1

**Textos en Azul** = lo que deben hacer en la entrega 2

**Textos en violeta** = lo que deben hacer en la entrega 3

### RESUMEN

Buscar archivos puede ser tedioso y demorado si no se hace por medio de un programa. Mediante un programa que crea una búsqueda de manera binaria, se podrá encontrar archivos de manera eficiente y rápida. Este problema puede relacionarse con el ordenamiento de archivos, programas de optimización y calcular el peso en orden de diferentes archivos. ¿Cuál es la solución?, ¿cuáles los resultados? y, ¿Cuáles las conclusiones? Utilizar máximo 200 palabras.

### Palabras clave

- Árbol binario.
- Búsqueda de archivos.
- Búsqueda binaria.
- Diseño de algoritmos.
- Estructura de datos.

### Palabras clave de la clasificación de la ACM

Theory of computation → Design and analysis of algorithms → Data structures design and analysis → Sorting and searching

## 1. INTRODUCCIÓN

Esta primera entrega tiene como fin la realización de un proyecto del curso estructuras de datos y algoritmos, el cual tiene como fin la implementación de software para el análisis y diseño de problemas que tenga nuestro entorno. Este proyecto tiene la capacidad de que cada estudiante obtenga las cualidades necesarias de poder formular y organizar un problema que constantemente nos afecta, además de obtener la habilidad del trabajo en equipo, el cual es el que nos espera en nuestro futuro laboral; el proyecto cuenta con diferentes partes como la descripción de la problemática y formulación del problema, igualmente en cada entrega se garantizara un desarrollo más amplio de nuestro problema, con el que se evidenciara los conocimientos que obtendremos durante el curso. Cada vez

el mundo necesita más el tiempo, y los algoritmos computacionales hacen que las operaciones que necesitemos rutinariamente sean más rápidas y nos agilicen procedimientos con mayor velocidad, para hacer esta clase de algoritmos se necesita de un pensamiento sistemático que tenga la capacidad de garantizar cada posibilidad de acontecimientos que pueden ocurrir, además de la estructuración de este. Nuestro grupo tiene como fin la solucionar de un problema de búsqueda de datos, en el cual en la actualidad es la acción que más se ven necesaria para la obtención de datos más rápidamente y eficientemente, es por ello que escogimos la búsqueda de datos binaria.

## 2. PROBLEMA

El problema que hay que resolver es mediante diferentes estructuras de datos en nuestro caso búsqueda binaria, saber que tan eficaz es un algoritmo, en el momento de buscar un archivo del computador, esto haciendo uso de las diferentes temáticas que se trabajaran en el semestre, para poder así saber que tan optimo es nuestro programa. La estructura servirá para buscar los archivos en un computador, de tal manera que sea eficiente y rápido, además para comparar que tan veloz es la estructura que se hizo, con la del resto de los compañeros, y que datos influyen el tiempo que transcurre la estructura.

## 3. TRABAJOS RELACIONADOS

### 3.1 Búsqueda Lineal

Consiste en examinar secuencias uno a uno hasta encontrar el buscado o haber revisado todos los elementos sin éxito. Si tenemos un array de enteros y X es el elemento que tratamos de localizar, un algoritmo simple que devuelve la posición del elemento en el array, si este se encuentra, o -1 si no se encuentra.

La complejidad de la búsqueda sucede cuando encontramos el elemento buscado en la última posición array. Como se requiere n ejecuciones del bucle mientras, la cantidad de tiempo es proporcional a la longitud del array n, más un cierto tiempo para realizar las condiciones del bucle mientras y para la llamada al método. Por lo tanto, la función que representa la forma de la curva para dicho algoritmo es de la forma  $an + b$  para ciertas constantes a y b. en notación  $O, O(an + b) = O(an) = O(n)$ .

### 3.2 Búsqueda Indexada

Necesita que los elementos sobre los que se realiza la búsqueda estén ordenados de acuerdo a una determinada clave. El método utiliza un array auxiliar denominado array índice. Cada objeto en el array índice consta de un valor y

la posición que ocupa dicho valor en el correspondiente array ordenado. Mediante cada elemento de array índice se asocian grupos de elementos del array inicial. Los métodos en ambos arráis beben estar ordenados, el método consta de dos pasos:

1. Buscar en el array \_índice el intervalo correspondiente al elemento buscado.
2. Restringir la búsqueda a los elementos a los de intervalo que se localizó previamente.

### 3.3 Título del tercer trabajo relacionado

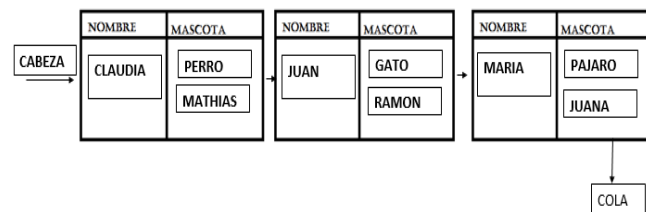
La búsqueda binaria proporciona un medio para reducir el tiempo requerido para buscar en una lista. Este método, sin embargo, exige que los datos estén ordenados y siempre dependientes del número n de elementos del conjunto de datos. Existe otro método que puede aumentar la velocidad de búsqueda donde los datos no necesitan estar ordenados y esencialmente es independiente del número n. Este método se conoce como transformación de claves (claves-dirección) o hashing. El proceso de hashing consiste en convertir el elemento almacenado (numérico o alfanumérico) en una dirección (índice) dentro del array.

### 3.4 Título del cuarto trabajo relacionado

Aquí deberán mencionar el cuarto problema algorítmico relacionado documentado y una solución

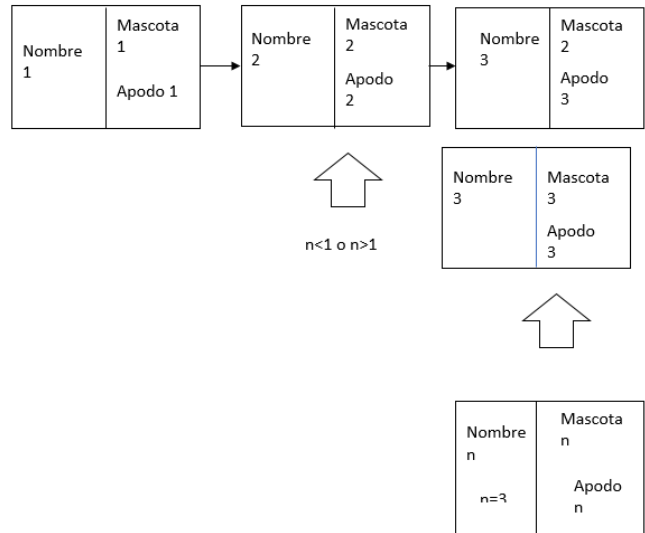
## 4. Búsqueda de Mascotas

El problema está en encontrar el nombre de la mascota y su especie, ingresando el nombre del dueño.



**Gráfica 1:** Lista simplemente enlazadas de nombres y mascotas. Los nombres contienen el nombre del dueño de la mascota y las mascotas contiene su especie y su nombre.

### 4.1 Operaciones de la estructura de datos



**Gráfica 2:** Imagen de una operación de borrado de una lista encadenada

### 4.2 Criterios de diseño de la estructura de datos

Porque de esta manera se recorre de una manera más eficiente la lista en busca del elemento que se piensa buscar. A su vez, su complejidad llega a ser adecuada junto a su consumo de memoria. Y consideramos que es la manera más efectiva de resolver nuestro problema.

### 4.3 Análisis de Complejidad

n	Búsqueda lineal	Búsqueda binaria
10	10	4
10 <sup>2</sup>	100	7
10 <sup>6</sup>	1'000.000	21
2.3x10 <sup>7</sup>	23'000.000	25
7x10 <sup>9</sup>	7'000.000.000	33(!)

**Tabla 1:** Tabla para reportar la complejidad

### 4.4 Tiempos de Ejecución

	Conjunto de datos 1
Operación 1 (10.000)	1
Operación 1 (100.000)	8
Operación 1 (1'000.000)	59

Operación 2(10'000.000)	591
operación 3(100'000.000)	Memory Error

**Tabla 2:** Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

#### 4.5 Memoria

Memoria y utilizada	Cantidad de memoria(KB)
Cantidad de memoria virtual reservada para el sistema operativo para el proceso (KB)	96.796
Cantidad de memoria física utilizada actualmente por el proceso (KB)	55.680
Cantidad de memoria física usada por el proceso que se puede compartir con otros proceso (KB)	4.824
Cantidad de memoria física usada por el proceso que no se puede usar otros procesos (KB)	50.856

**Tabla 3:** Consumo de memoria de la estructura de datos con diferentes conjuntos de datos

#### 4.6 Análisis de los resultados

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:

Tabla de valores durante la ejecución			
Estructuras de autocompletado	LinkedList	Arrays	HashMap
Espacio en el Heap	60MB	175MB	384MB
Tiempo creación	1.16 - 1.34 s	0.82 - 1.1 s	2.23 - 2.6 s
Tiempo búsqueda ("a")	0.31 - 0.39 s	0.37 - 0.7 s	0.22 - 0.28 s
Tiempo búsqueda ("zyzzvas")	0.088 ms	0.038 ms	0.06 ms
Búsqueda ("aerobacteriologically")	0.077 ms	0.041 ms	0.058 ms
Tiempo búsqueda todas las palabras	6.1 - 8.02 s	4.07 - 5.19 s	4.79 - 5.8 s

**Table 4:** Análisis de los resultados obtenidos con la implementación de la estructura de datos

## 5. TÍTULO DE LA SOLUCIÓN FINAL DISEÑADA

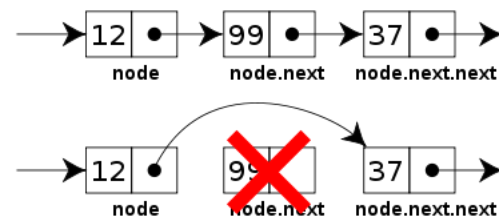
Implementen una estructura de datos para solucionar finalmente el problema y grafíquenla. Además, pruébenla con los datos que están en la carpeta de Conjunto de Datos del .ZIP



**Gráfica 3:** Lista simplemente encadenada de personas. Una persona es una clase que contiene nombre, cédula y foto

### 5.1 Operaciones de la estructura de datos

Diseñen las operaciones de la estructura de datos para solucionar finalmente el problema. Incluyan una imagen explicando cada operación



**Gráfica 4:** Imagen de una operación de borrado de una lista encadenada

### 5.2 Criterios de diseño de la estructura de datos

Expliquen con criterios objetivos, por qué diseñaron así la estructura de datos. Criterios objetivos son, por ejemplo, la eficiencia en tiempo y memoria. Criterios no objetivos y que rebajan la nota son: “me enfermé”, “fue la primera que encontré”, “la hice el último día”, etc. Recuerden: este es el numeral que más vale en la evaluación con 40%

### 5.3 Análisis de la Complejidad

Calculen la complejidad de las operaciones de la nueva estructura de datos para el peor de los casos. Vean un ejemplo para reportarla:

Método	Complejidad
Búsqueda Fonética	O(1)
Imprimir búsqueda fonética	O(m)
Insertar palabra búsqueda fonética	O(1)
Búsqueda autocompletado	O(s + t)
Insertar palabra en TrieHash	O(s)
Añadir búsqueda	O(s)

**Tabla 5:** Tabla para reportar la complejidad

## 5.4 Tiempos de Ejecución

Calculen, (I) el tiempo de ejecución y (II) la memoria usada para las operaciones de la nueva estructura de datos, para el Conjunto de Datos que está en el ZIP. Explicar el tiempo para varios ejemplos

Tomen 100 veces el tiempo de ejecución y memoria de ejecución, para cada conjunto de datos y para cada operación de la estructura de datos

	Conjunto de Datos 1	Conjunto de Datos 2	...Conjunto de Datos n
Creación	10 sg	20 sg	5 sg
Operación 1	12 sg	10 sg	35 sg
Operación 2	15 sg	21 sg	35 sg
Operación n	12 sg	24 sg	35 sg

**Tabla 6:** Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

## 5.5 Memoria

Mencionar la memoria que consume el programa para los conjuntos de datos

	Conjunto de Datos 1	Conjunto de Datos 2	...Conjunto de Datos n
Consumo de memoria	10 MB	20 MB	5 MB

**Tabla 7:** Consumo de memoria de la estructura de datos con diferentes conjuntos de datos

## 5.6 Análisis de los resultados

Expliquen los resultados obtenidos. Hagan una gráfica con los datos obtenidos, como por ejemplo:

Tabla de valores durante la ejecución			
Estructuras de autocompletado	LinkedList	Arrays	HashMap
Espacio en el Heap	60MB	175MB	384MB
Tiempo creación	1.16 - 1.34 s	0.82 - 1.1 s	2.23 - 2.6 s
Tiempo búsqueda ("a")	0.31 - 0.39 s	0.37 - 0.7 s	0.22 - 0.28 s
Tiempo búsqueda ("zyzzzyas")	0.088 ms	0.038 ms	0.06 ms
Búsqueda ("aerobacteriologically")	0.077 ms	0.041 ms	0.058 ms
Tiempo búsqueda todas las palabras	6.1 - 8.02 s	4.07 - 5.19 s	4.79 - 5.8 s

**Tabla 8:** Tabla de valores durante la ejecución

## 6. CONCLUSIONES

Para escribirlas, procedan de la siguiente forma: 1. En un párrafo escriban un resumen de lo más importante que hablaron en el reporte. 2. En otro expliquen los resultados más importantes, por ejemplo, los que se obtuvieron con la solución final. 3. Luego, comparen la primera solución que hicieron con los trabajos relacionados y la solución final. 4. Por último, expliquen los trabajos futuros para una posible continuación de este Proyecto. Aquí también pueden mencionar los problemas que tuvieron durante el desarrollo del proyecto

### 6.1 Trabajos futuros

Respondan ¿Qué les gustaría mejorar en el futuro? ¿Qué les gustaría mejorar estructura de datos o a la implementación?

## AGRADECIMIENTOS

Identifiquen el tipo de agradecimiento que van a escribir: para una persona o para una institución. Tengan en cuenta que: 1. El nombre del docente no va porque él es autor. 2. Tampoco sitios de internet ni autores de artículo leídos con quienes no se han contactado. 3. Los nombres que sí van son quienes ayudaron, compañeros del curso o docentes de otros cursos.

Aquí un ejemplo: Esta investigación fue soportada parcialmente por [Nombre de la fundación que paga su beca].

Nosotros agradecemos por su ayuda con [una técnica particular o metodología] a [Nombre, Apellido, cargo, lugar de trabajo] por sus comentarios que ayudaron a mejorar esta investigación.

BORRAR LOS CORCHETES ([]).

## REFERENCIAS

Referenciar las fuentes usando el formato para referencias de la ACM. Léase en <http://bit.ly/2pZnE5g> Vean un ejemplo: 1. JairoCarrera. (24 de 07 de 2010). algoritmo y estructuras de datos i.rar. pág. [https://www.4shared.com/rar/SC5hGmhd/algoritmo\\_y\\_estructuras\\_de\\_dat.html?cau2=403tNull&ua=WINDO](https://www.4shared.com/rar/SC5hGmhd/algoritmo_y_estructuras_de_dat.html?cau2=403tNull&ua=WINDO) S. 2. <https://es.khanacademy.org/computing/computerscience/algorithms/intro-to-algorithms/a/a-guessing-game>