

# **BÚSQUEDA SECUENCIAL Y BINARIA**



## **INTEGRANTES**

Gabriela Martínez Eslava - 20202020117

Luis Alejandro Cely Díez - 20211020093

## **PROFESOR**

Santiago Salazar Fajardo

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS  
FACULTAD DE INGENIERÍA  
INGENIERÍA DE SISTEMAS  
BOGOTÁ, D.C.  
12 DE FEBRERO 2024**

## **CONTENIDO**

### **INTRODUCCIÓN**

#### **BÚSQUEDA SECUENCIAL**

- Qué es la búsqueda secuencial
- Ventajas y desventajas de la búsqueda secuencial

#### **BÚSQUEDA BINARIA**

- Qué es la búsqueda binaria
- Ventajas de la búsqueda secuencial

#### **MODELO CONCEPTUAL Y DE OBJETOS**

- Diagrama de clases de la implementación
- Diagrama de objetos de la implementación

#### **TOMA DE TIEMPOS DE EJECUCIÓN**

- Toma de tiempos para 100 elementos
- Toma de tiempo para 1.000 elementos
- Toma de tiempo para 100.000 elementos
- Toma de tiempo para 700.000 elementos

### **CONCLUSIONES**

## **INTRODUCCIÓN**

Los métodos de ordenamiento son algoritmos diseñados para organizar una colección de elementos en un orden específico, como ascendente o descendente, según un criterio definido. Estos algoritmos son esenciales en la informática debido a su amplia gama de aplicaciones y su importancia en la optimización de la eficiencia y el rendimiento de los programas.

En el presente documento se abordan los métodos de ordenamiento búsqueda secuencial y búsqueda binaria, se compara su procedimiento y eficacia a la hora de ordenar una colección de datos, partiendo de una pequeña hasta considerar cantidades más amplias de datos, todo esto, basado en la implementación realizada.

## **BÚSQUEDA SECUENCIAL**

La búsqueda secuencial es un algoritmo de búsqueda que recorre secuencialmente cada elemento en una colección de datos, comparando cada uno con el valor buscado. Se detiene cuando encuentra el elemento deseado o cuando llega al final de la colección. Es simple pero puede ser menos eficiente en conjuntos de datos grandes en comparación con otros algoritmos de búsqueda.

### **VENTAJAS:**

- Implementación simple: La búsqueda secuencial es fácil de entender e implementar, es adecuado su uso para situaciones en donde la complejidad del algoritmo no es algo primordial
- No requiere datos ordenados: no es necesario que los datos estén ordenados de ninguna manera para su implementación
- Uso en conjuntos no uniformes de datos: funciona adecuadamente en contextos en donde la probabilidad de encontrar un elemento está igualmente distribuida en todos ellos, ya que no asume ninguna estructura de los datos.
- eficiencia en conjuntos pequeños: en conjuntos pequeños puede ser tan eficiente como otros algoritmos más complejos.

### **DESVENTAJAS:**

- Poca eficiencia en grandes conjuntos de datos: la búsqueda secuencial puede volverse ineficiente cuando se trabaja con grandes cantidades de datos, ya que requiere examinar cada elemento de la colección en orden.
- Tiempo de ejecución: el tiempo de ejecución de la búsqueda secuencial aumenta linealmente con el tamaño de la colección de datos. Lo cual no es práctico en conjuntos grandes.
- No aprovecha datos ordenados: a diferencia de otros algoritmos de ordenamiento, la búsqueda secuencial no obtiene resultados más eficientes si se proporciona una colección ordenada de datos.

## BÚSQUEDA BINARIA

A diferencia de la búsqueda secuencial donde se asumen los elementos en cualquier orden, se puede hacer una búsqueda mucho más eficiente en un arreglo ordenado. La búsqueda binaria compara si el valor buscado se encuentra en la mitad superior o la mitad inferior de un arreglo, subdividiendo hasta encontrar un resultado.

Este algoritmo posee una complejidad de  $O(\log n)$ , siendo “n” el tamaño de la lista, lo cual comparado con la búsqueda secuencial es muchísimo más eficiente considerando su peor caso, especialmente con listas grandes.

### VENTAJAS

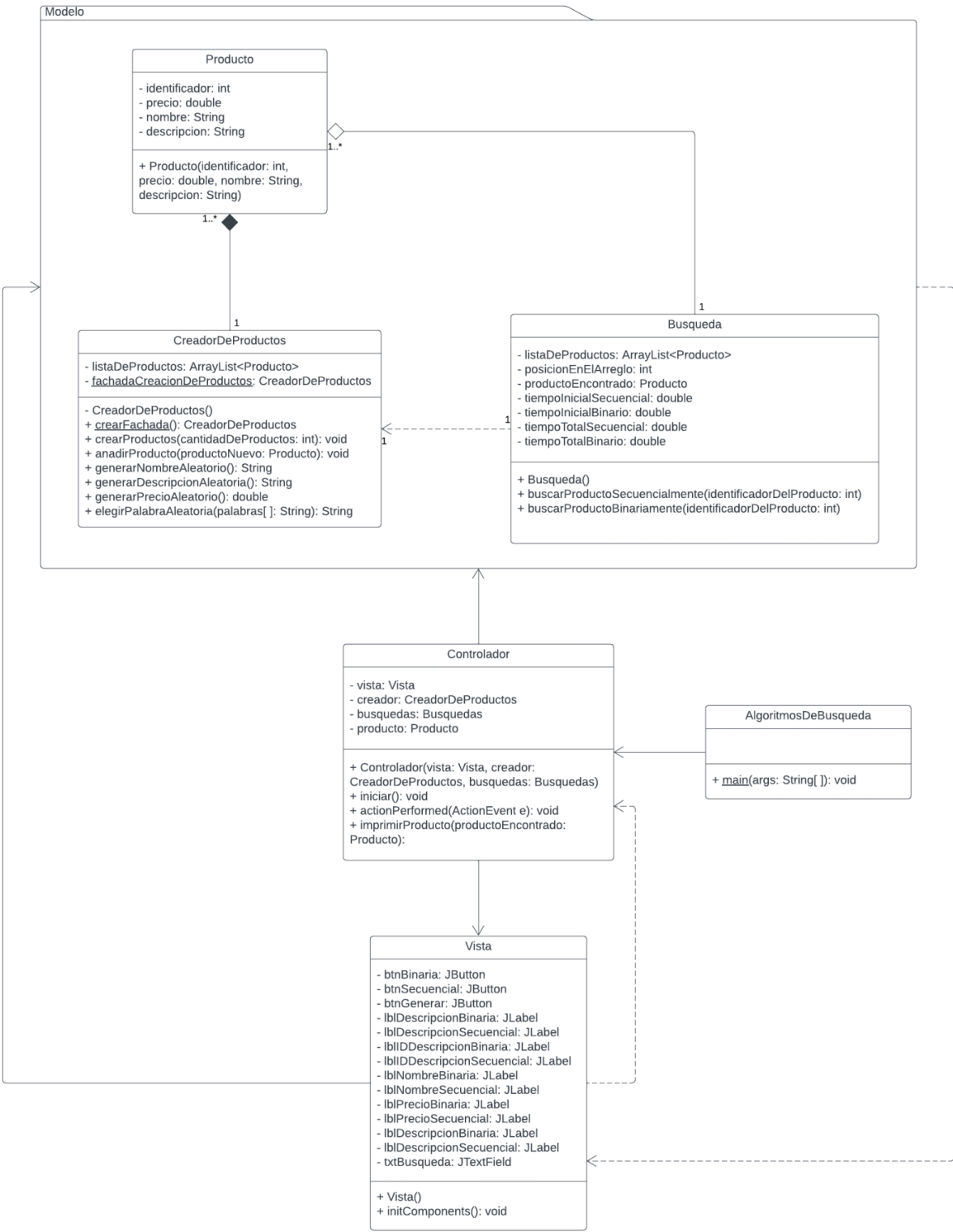
- Eficiencia en el tiempo: Tal y como se mencionó anteriormente, debido a su complejidad, este algoritmo es mucho más rápido comparado con otros de un orden mucho más complejo. Su complejidad de  $O(\log n)$  lo hace una muy buena opción, especialmente teniendo arreglos previamente ordenados.
- Reducción del espacio de búsqueda: Una de las razones por las que este algoritmo es mucho más rápido que el anterior es debido a que al cercar el espacio de búsqueda con cada iteración, nos permite reducir significativamente el espacio en el cual tenemos que buscar.
- Implementación sencilla: La facilidad de la comprensión de su lógica lo convierte en un algoritmo de una aplicación e implementación muy sencilla, convirtiéndolo en un algoritmo ampliamente organizado y elegante.
- Predictibilidad del rendimiento: El hecho de tener una complejidad logarítmica, asegura que el tiempo de ejecución es extremadamente manejable independientemente de qué tanto se amplíe el conjunto de búsqueda.

### DESVENTAJAS

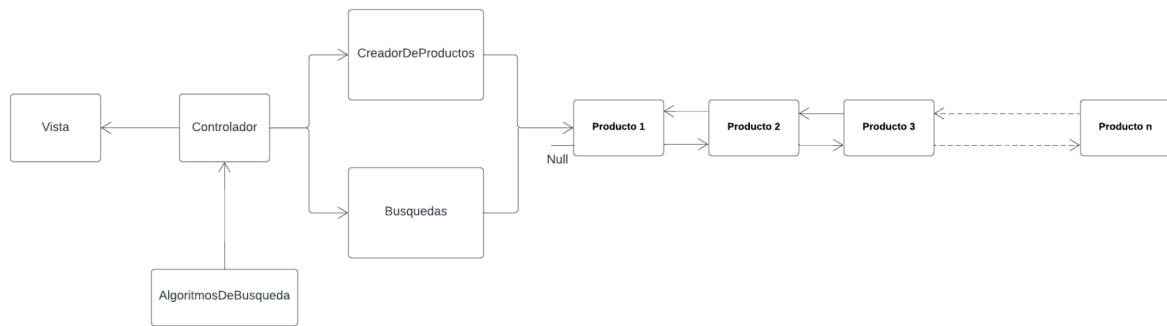
- Debe ser ordenado: La principal desventaja es que el arreglo sobre el cual se trabaja debe estar ordenado, por lo que si la lista no lo está, se debe aplicar antes un algoritmo de ordenamiento.
- No es versátil en todas las situaciones: Si la lista es muy pequeña es mejor optar por otros algoritmos que brinden mayor velocidad.
- Datos duplicados: El algoritmo no está diseñado para manejar datos duplicados adecuadamente, por lo que devolverá la primera ocurrencia que encuentre.

MODELO CONCEPTUAL Y DE OBEJTS

DIAGRAMA DE CLASES



## DIAGRAMA DE OBJETOS



## TOMA DE TIEMPOS DE EJECUCIÓN

Para cada una de las cantidades de datos propuestas, se tomarán 10 muestras de tiempos de ejecución y se promediarán, es este valor el que se consignará en la tabla para ambas búsquedas,

Algoritmo	100 Elementos	1.000 elementos	100.000 elementos	700.000 elementos
Búsqueda Secuencial	29.000 nanosegundos	142.800 nanosegundos	4'438.200 nanosegundos	894'466.100 nanosegundos
Búsqueda Binaria	20.400 nanosegundos	23.400 nanosegundos	32.800 nanosegundos	5'857.900 nanosegundos

Como podemos observar. La búsqueda binaria, en todos los casos fue más efectiva que a búsqueda secuencial, sin embargo, podemos resaltar que, entre más grande es el conjunto de datos, mayor se hace la diferencia entre tiempos.



## CONCLUSIONES

En términos de velocidad de los algoritmos de búsqueda, la búsqueda binaria es una gran opción debido a su complejidad logarítmica del orden  $O(\log n)$ , demostrando en cada ejecución ser extremadamente mejor que la búsqueda secuencial, siendo el indicado para encontrar información en listas de un gran tamaño; sin embargo, este algoritmo tiene sus desventajas marcadas por el orden y el tamaño de la lista.

Si la lista se encuentra desordenada el indicado para esta tarea de búsqueda es la búsqueda secuencial, pues aunque tiene un tiempo de ejecución mucho mayor, la búsqueda binaria no aplicaría sobre este tipo de arreglos, por lo cual previamente habría que aplicar un algoritmo de ordenamiento sobre el arreglo para poder utilizarla.

Mientras más grande sea el tamaño de la lista, más notable será la brecha de tiempos entre la búsqueda secuencial y la búsqueda binaria, dando como indiscutible ganador a la búsqueda binaria. Y, aunque sea más simple aplicar un algoritmo de búsqueda secuencial, la búsqueda binaria no está muy lejos de la simplicidad y es mucho más eficiente en diversos aspectos.

Para saber qué algoritmo aplicar es importante conocer el contexto sobre el cual se está trabajando, por lo que si poseemos un conjunto de datos estáticos o que no poseen muchos cambios y se encuentran ordenados o poseen un algoritmo de ordenamiento previo, es claramente mucho más factible utilizar el algoritmo de búsqueda binaria, y más aún si necesitamos rapidez a la hora de encontrar rápidamente aquel elemento que estamos buscando, teniendo mucho cuidado con la repetición de elementos.