

Trabajo Práctico

Integrador: Algoritmos de búsqueda y ordenamiento

Franco Herrera, Brian Gutierrez Colque

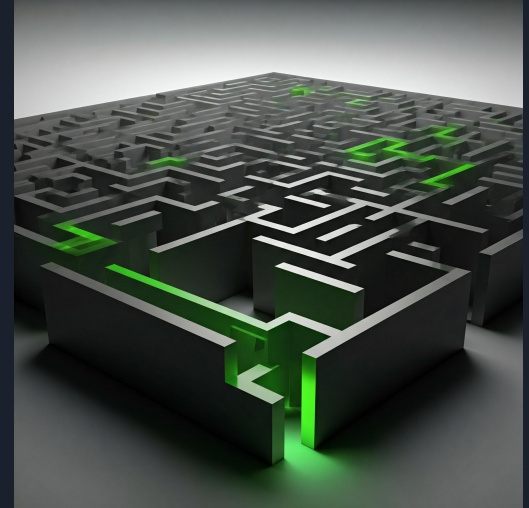
¿Por qué Algoritmos de búsqueda y ordenamiento?

La Importancia de los Algoritmos de Búsqueda y Ordenamiento

Descripción: Herramientas esenciales para manejar datos de manera eficiente y efectiva en programación.

Puntos Clave:

- **Fundamentales:** Permiten manejar datos de manera eficiente y efectiva.
- **Optimización:** Esenciales para optimizar el rendimiento de cualquier aplicación.
- **Dominio Básico:** Conocimiento básico para todo programador.
- **Resolución de Problemas:** Facilitan la resolución efectiva de problemas complejos.
- **Bases para Software:** Sientan las bases para el desarrollo de software robusto y eficiente.



¿Qué son los algoritmos de búsqueda?

¿Qué son los Algoritmos de Búsqueda?

Descripción: Conjunto de instrucciones para encontrar un elemento específico dentro de una colección de datos.

Puntos Clave:

Propósito: Encontrar un elemento específico dentro de una colección de datos.

Tipos de Estructura: Existen diferentes algoritmos según el tipo de estructura de datos.

Usos Comunes: Encontrar el menor o mayor elemento, o el índice de un elemento determinado.

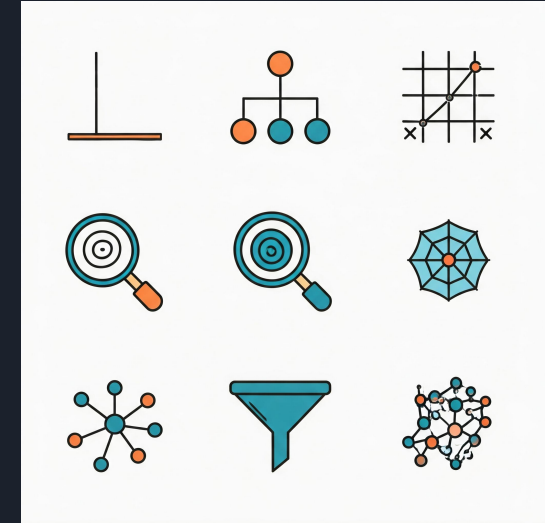
Ubicuidad: Usados por casi todas las aplicaciones.



Tipos comunes de algoritmos de búsqueda

Tipos Comunes de Algoritmos de Búsqueda Puntos Clave:

- **Búsqueda Lineal:**
 - Más sencillo.
 - Recorre secuencialmente hasta encontrar el objetivo o confirmar su ausencia.
 - Funcional pero poco eficiente.
- **Búsqueda Binaria:**
 - Para estructuras ordenadas.
 - Divide repetidamente el espacio de búsqueda a la mitad, reduciendo el tiempo.
- **Búsqueda de Interpolación:**
 - Similar a la búsqueda binaria.
 - Usa una fórmula para estimar la ubicación del valor buscado.
 - Más eficiente en estructuras ordenadas y uniformemente distribuidas.
- **Búsqueda Exponencial:**
 - Combina estrategias lineales y exponenciales.
 - Útil cuando se desconoce el tamaño de la estructura.
- **Búsqueda por Saltos:**
 - Salta algunos elementos y realiza búsqueda lineal dentro de bloques.



Aplicaciones prácticas de algoritmos de búsqueda

Aplicaciones Prácticas de Algoritmos de Búsqueda **Puntos Clave:**

- **Sistemas de Base de Datos:** Consultas eficientes para extraer información específica.
- **Motores de Búsqueda:** Optimizan la recuperación de resultados relevantes.
- **Inteligencia Artificial:** Encontrar soluciones eficientes en problemas complejos (juegos, rutas óptimas).
- **Procesamiento de Lenguaje Natural (PLN):** Facilita búsqueda semántica o recuperación eficiente en textos.



¿Qué son los algoritmos de ordenamiento?

Conjunto de procedimientos para organizar datos en una secuencia específica (ascendente o descendente). **Puntos Clave:**

- **Propósito:** Organizar datos en una secuencia específica.
- **Fundamental:** Operación fundamental para mejorar eficiencia en la búsqueda de datos.
- **Beneficios:** Facilita la visualización y análisis de información.
- **Importancia:** Capacidad para manejar conjuntos de datos grandes y complejos.
- **Desarrollo Eficiente:** Permite crear aplicaciones más eficientes.



Tipos comunes de algoritmos de ordenamiento

Tipos Comunes de Algoritmos de Ordenamiento **Puntos Clave:**

- **Ordenamiento por Selección:**
 - Divide datos en parte ordenada y no ordenada.
 - Busca el elemento más pequeño (o más grande) en la parte no ordenada y lo intercambia.
- **Ordenamiento por Inserción:**
 - Inserta cada elemento en su posición correcta respecto a los anteriores.
 - Eficiente para listas pequeñas o casi ordenadas.
- **Ordenamiento Rápido (QuickSort):**
 - Usa el principio "divide y vencerás".
 - Selecciona un "pivote" y reorganiza elementos menores a la izquierda y mayores a la derecha.
 - Aplica recursivamente el proceso a las sublistas.
- **Ordenamiento por Mezcla (MergeSort):**
 - Divide el conjunto original hasta que cada sublista tiene un solo elemento.
 - Combina las sublistas para crear listas ordenadas.
- **Ordenamiento Burbuja (Bubble Sort):**
 - Compara pares adyacentes e intercambia si están en orden incorrecto.
 - Repite hasta que no hay más intercambios.



Aplicaciones prácticas de algoritmos de ordenamiento

Aplicaciones Prácticas de Algoritmos de Ordenamiento

Puntos Clave:

- **Búsqueda Eficiente:** Mejora el rendimiento de algunos algoritmos de búsqueda (ej. búsqueda binaria).
- **Análisis Estadístico:** Facilita cálculos como promedios o medianas.
- **Visualización:** Permite la presentación clara de datos en gráficos o tablas organizadas.



Caso práctico

Programa que permite al usuario buscar y ordenar una lista.

Características del Programa:

- **Búsqueda:** Búsqueda Lineal y Búsqueda Binaria.
- **Ordenamiento:** Ordenamiento Rápido y Ordenamiento Burbuja.
- **Medición:** Muestra el tiempo de ejecución de cada algoritmo.
- **Tipo de Dato:** Diccionario, para ordenar alumnos (nombre, legajo, promedio).





¿Qué es un diccionario?

Antes de pasar a la prueba del programa, necesitamos definir qué es un diccionario. Un diccionario es un tipo de dato estructurado que permite guardar pares clave-valor. Entonces, en nuestro caso, cada alumno está conformado por un **nombre**, **legajo** y **promedio**:

```
alumno = {  
    "nombre": "Franco",  
    "legajo": 1,  
    "promedio": 75  
}
```

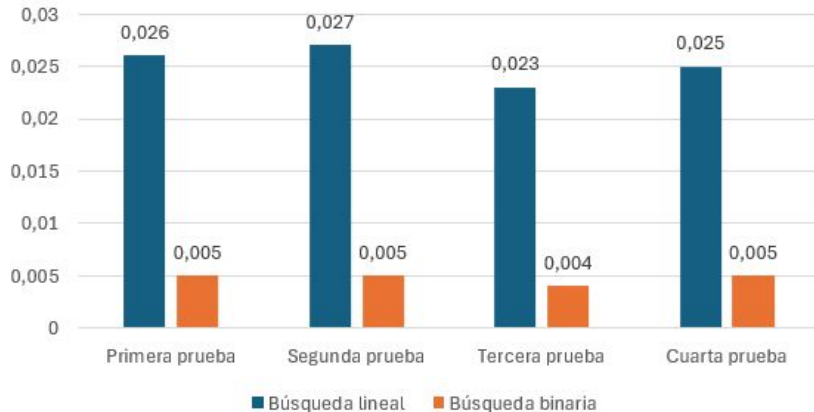
Probamos el programa...

```
src > Components > C3 > Index.js > ...  
1 import {use} from "react"  
2 import React useCallba... function React.useCallba... extends...  
3 import Theme useContext  
4 useDebugValue  
5 function C3 useEffect  
6 const the useImperativeHandle  
7 return ( useLayoutEffect  
8   useMemo  
9   <p>So useReducer  
10  <p>En useRef  
11  <butt useState  
12    Cli useCallback  
13  </but useContext  
14  </>  
15  );  
16 }
```

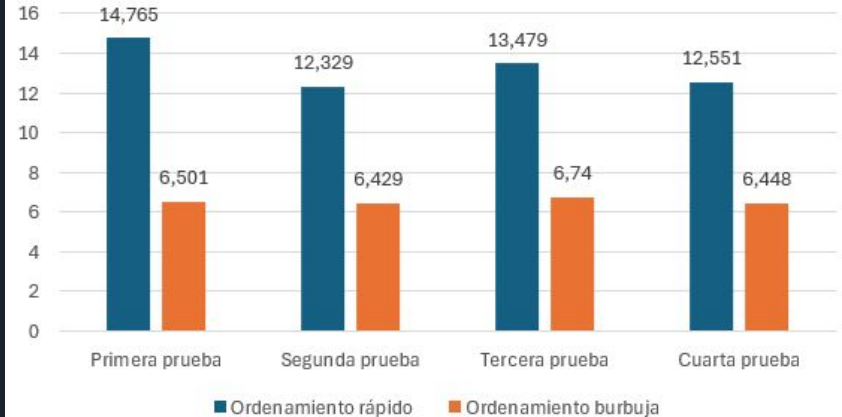
Resultados obtenidos

Para probar nuestro caso práctico, creamos una función que genera una lista de alumnos de manera dinámica. El problema fue que esta lista se genera para cada ejecución, por lo que no usamos exactamente la misma lista para cada caso. Sin embargo, nos aseguramos de que la lista contenga siempre la misma cantidad de elementos.

Tiempos de ejecución (milisegundos)



Tiempos de ejecución (milisegundos)



Conclusiones

- Los algoritmos de búsqueda y ordenamiento son pilares fundamentales en la programación.
- Permiten gestionar y manipular datos de manera eficiente.
- La capacidad de buscar un elemento específico o de ordenar datos para su análisis es esencial en la mayoría de las aplicaciones informáticas.
- En este trabajo, se aplicaron cuatro tipos de algoritmos diferentes: dos de ordenamiento y dos de búsqueda.
- Se observó que la búsqueda binaria es más eficiente que la búsqueda lineal.
- Sin embargo, la búsqueda binaria requiere que la lista esté ordenada.
- El ordenamiento burbuja redujo los tiempos de ejecución a la mitad en comparación con el ordenamiento rápido.
- Cada algoritmo tiene sus propias ventajas y desventajas en términos de complejidad temporal y espacial.

