



Trabajo Práctico Integrador

ANÁLISIS DE ALGORITMOS

Materia: Programación I

Profesor: Sebastián Bruselario

Tutor: Virginia Cimino

Fecha de Entrega: 09 de junio de 2025

Alumnos: Florencia Herrera y Federico Quinteros

Elección del tema



Importancia en la programación



Objetivos del trabajo

¿ Qué es un algoritmo?

Es una serie de pasos lógicos y finitos que, cuando se siguen en un orden específico, permiten resolver un problema o realizar una tarea específica.





¿ Que es el análisis de algoritmos?

El análisis de algoritmos es el proceso de evaluar la eficiencia y rendimiento de un algoritmo, generalmente en términos de su complejidad temporal y espacial.

Principales características:

- El análisis de las características estructurales del algoritmo que respalda el programa.
- La cantidad de memoria que utiliza para resolver un problema.
- La evaluación del diseño de las estructuras de datos del programa, midiendo la eficiencia de los algoritmos para resolver el problema planteado.



Tipos de análisis

Análisis Teórico

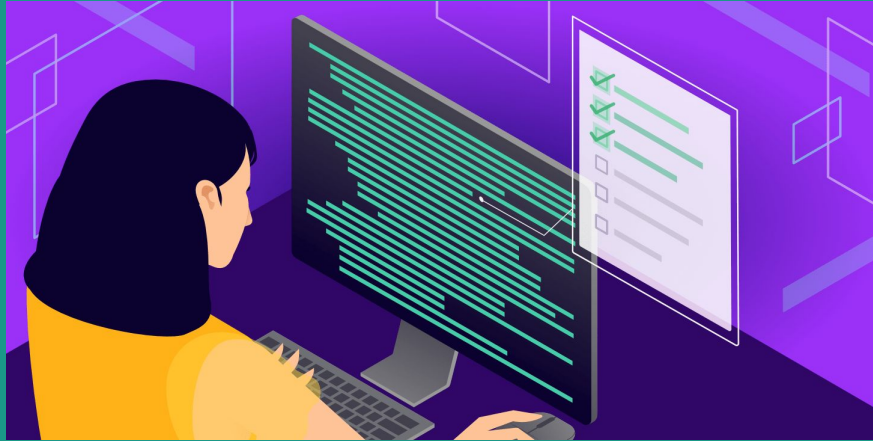
Consiste en evaluar su eficiencia (en tiempo y espacio) utilizando fórmulas matemáticas, expresiones simbólicas (como notación Big O), y sin necesidad de implementarlo o probarlo en una computadora.

Análisis Empírico

El análisis empírico consiste en medir el rendimiento real de un algoritmo ejecutándolo en la práctica con distintos datos de entrada, y observando cuánto tiempo tarda y cuanta memoria usa en la computadora.

Caso Práctico

COMPARACIÓN DE DOS ALGORITMOS PARA ENCONTRAR EL NÚMERO
MÁS GRANDE EN UNA LISTA



—



Resultados obtenidos

Análisis Empírico

```
Resultados de busqueda automatica
1000 0.0000000000
10000 0.0009901524
100000 0.0010099411
1000000 0.0127294064
10000000 0.1263551712
Resultados de busqueda manual
1000 0.0009953976
10000 0.0009970665
100000 0.0210082531
1000000 0.1627681255
10000000 1.5779354572
```

Análisis Teórico

Algoritmo manual: $T(n) = 1+1+1+1+(2N) = 4+(2*n)$.

Algoritmo automático: $T(n) = 1+1+2 = 4$

Notación Big O

Algoritmo manual : $O(n)$

Algoritmo automático : $O(2)$



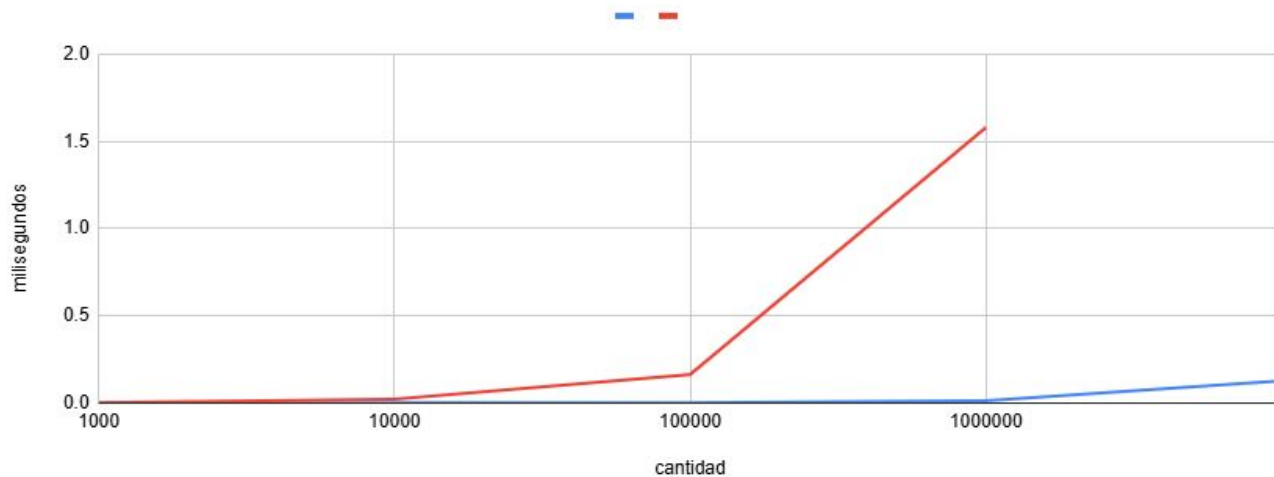
Metodología Utilizada

- Análisis teórico.
- Generación de datos
- Medición del tiempo de ejecución
- Los resultados se registraron y se compararon.
- Comparación de resultados de tiempo real.
- Documentación del proceso en repositorio GitHub.

Resultados Obtenidos a partir de una comparación

- Ambos algoritmos devuelven el mismo resultado correcto.
- La opción automática es mucho más rápida y eficiente que la manual a medida que aumenta el tamaño de la lista.

Búsqueda Manual y Búsqueda Automática





Conclusión

Siempre que sea posible, es recomendable utilizar funciones integradas del lenguaje como `max()`, ya que no sólo simplifican el código, sino que además ofrecen un mejor rendimiento, especialmente en estructuras de datos grandes.

Este tipo de comparaciones permite desarrollar un pensamiento crítico sobre la eficiencia de distintas soluciones y tomar decisiones fundamentadas a la hora de programar.

GRACIAS POR SU ATENCIÓN

