

algoritmo: Secuencia finita de instrucciones donde cada una de las cuales tiene un significado preciso y puede ejecutarse con una cantidad finita de recursos computacionales en un tiempo finito.

### **Análisis de algoritmo**

El análisis de algoritmos es una herramienta para hacer la evaluación del diseño de un algoritmo, permite establecer la calidad de un programa y compararlo con otros que puedan resolver el mismo problema, sin necesidad de desarrollarlos. El análisis de algoritmos estudia, desde un punto de vista teórico, los recursos computacionales que requiere la ejecución de un programa, es decir su eficiencia (tiempo de CPU, uso de memoria, ancho de banda, ...). Además de la eficiencia en el desarrollo de software existen otros factores igualmente relevantes: funcionalidad, corrección, robustez, usabilidad, modularidad, mantenibilidad, fiabilidad, simplicidad y aún el propio costo de programación.

El análisis de algoritmos se basa en:

- 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.

### **Eficiencia de algoritmos**

Un algoritmo será más eficiente cuanto menos recursos computacionales consuma: tiempo y espacio de memoria requerido para su ejecución.

La eficiencia de un algoritmo se cuantifica en términos de su complejidad temporal (tiempo de cómputo del programa) y su complejidad espacial (memoria que utiliza el programa durante su ejecución).

### **Complejidad en espacio de memoria**

Decir que un programa es  $O(n)$  en espacio significa que sus requerimientos de memoria aumentan proporcionalmente con el tamaño del problema. Esto es, si el problema se duplica, se necesita el doble de memoria. Del mismo modo, para un programa de complejidad  $O(n^2)$  en espacio, la cantidad de memoria que se necesita para almacenar los datos crece con el cuadrado del tamaño del problema: si el problema se duplica, se requiere cuatro veces más memoria. En general, el cálculo de la complejidad en espacio de un algoritmo es un proceso sencillo que se realiza mediante el estudio de las estructuras de datos y su relación con el tamaño del problema.

Los requerimientos estáticos de memoria se refieren al tamaño de los objetos que resuelven el problema. El espacio requerido por los objetos de datos de los tipos primitivos y de los tipos estructurados de los lenguajes de programación, son fácilmente calculables, dependiendo de la implementación del mismo. Se les conoce como objetos de datos estáticos porque la cantidad de espacio que requieren es deducible en su declaración, por

lo que ya en compilación (si el lenguaje es compilado), el compilador conoce cuánta memoria requerirán, que será invariante a lo largo del alcance del objeto.

El análisis de los requerimientos dinámicos de memoria es relativo a los lenguajes que proveen mecanismos de asignación dinámica de la misma. En este caso, la complejidad en espacio viene dada por la cantidad de objetos existentes en un punto del programa, según las reglas de alcance. Así, la cantidad de espacio requerido no será la sumatoria de todas las declaraciones de datos, sino la máxima cantidad de memoria en uso en un momento dado

de la ejecución del programa.

El problema de eficiencia de un programa se puede plantear como un compromiso entre el tiempo y el espacio utilizados. Por eso, la etapa de diseño es tan importante dentro del proceso de construcción de software, ya que va a determinar en muchos aspectos la calidad del producto obtenido.

Ing. Nestor Diaz. (2004). Introducción a la complejidad computacional. 3/09/2018, de Universidad del Cauca Sitio web: <http://artemisa.unicauca.edu.co/~nediaz/EDDI/cap01.htm>