



A la hora de desarrollar software es importante analizar su rendimiento en el tiempo. Una herramienta muy útil para realizar este análisis es la Notación Big O que el alumno de la escuela debe conocer

## Análisis del rendimiento de las aplicaciones de IA

A la hora de desarrollar software es importante analizar su complejidad, mantenibilidad y escalabilidad. En este análisis el rendimiento de la aplicación se convierte en un factor clave. A medida que las aplicaciones crecen estas necesitan gestionar cada vez más comunicaciones, más datos, más concurrency, más computation, más petitions, ...

Los algoritmos son el pilar de cualquier aplicación y suelen consumir una parte importante de los recursos del servidor pero no todos los algoritmos rinden igual ni consumen lo mismo! Algunos son rápidos con pocos datos y lentos con muchos o viceversa. Algunos tienen un comportamiento lineal en su particular latencia a medida que incorporamos más datos al software y otros sin embargo pueden tener un comportamiento logarítmico o exponencial.

Ahora ya sabemos que el rendimiento de las aplicaciones varia según aumenta el volumen de datos que gestionan, por lo tanto, hacer un exhaustivo análisis del posible comportamiento del software en el futuro es necesario para no llevarnos sorpresas inesperadas.

Saber pronosticar el comportamiento en tiempos de ejecución de los algoritmos utilizados en cualquier software de IA es inevitable por dos motivos:

- 1.- Porque se han sobredimensionado las necesidades computacionales del software y se están contratando servidores en exceso. Esto encarecería nuestro producto y estaríamos pagando por un servicio que realmente no necesitamos. Además en tecnología los precios de la competencia suelen estar muy ajustados por lo que si pretendemos ser competitivos estamos obligados a reducir en costes.
- 2.- Porque se han desdeñado las necesidades reales del aplicativo para funcionar correctamente por lo que el software será lento y la experiencia de usuario pésima. En el Mundo del 5G que un aplicativo sea lento no es una buena noticia.

Lo **costoso** y la **lentitud** no son una opción en el sector tecnológico. El público cada vez es más exigente con el precio y con la experiencia de usuario por lo que cualquier empresa experta en IA está obligada a ajustar el precio todo lo posible para ser competitiva pero sin que afecte al rendimiento de su servicio. Por lo tanto es necesario medir y pronosticar el comportamiento de los algoritmos en diferentes circunstancias y hacer un análisis y una planificación preventiva para evitar problemas en futuros acontecimientos e intentar que los algoritmos sean eficientes.

Según wikipedia la eficiencia algorítmica es: *"En Ciencias de la Computación, el término eficiencia algorítmica es usado para describir aquellas propiedades de los algoritmos que están relacionadas con la cantidad de recursos utilizados por el algoritmo. Un algoritmo debe ser analizado para determinar el uso de los recursos que realiza. La eficiencia algorítmica puede ser vista como análogo a la ingeniería de productividad de un proceso repetitivo o continuo.*

*Con el objetivo de lograr una eficiencia máxima se quiere minimizar el uso de recursos. Sin embargo, varias medidas (e.g. complejidad temporal, complejidad espacial) no pueden ser comparadas directamente. Luego, cuál de dos algoritmos es considerado más eficiente, depende de cuál medida de*

*eficiencia se está considerando como prioridad, e.g. la prioridad podría ser obtener la salida del algoritmo lo más rápido posible, o que minimice el uso de la memoria, o alguna otra medida particular."*

Ya que no todos los algoritmos se comportan igual en el tiempo es necesario medir su rendimiento y eficiencia en el tiempo a medida que el volumen de datos que tienen que gestionar aumenta. Existen diferentes técnicas/metodologías para planificar el comportamiento de los algoritmos. Una de las más utilizadas es la notación O Grande (Big O Notation):

- 1.- [https://medium.com/@charlie\\_fuentes/notacion-big-0-para-principiantes-f9cbb4b6bec8](https://medium.com/@charlie_fuentes/notacion-big-0-para-principiantes-f9cbb4b6bec8)
- 2.- <https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx>
- 3.- <https://www.youtube.com/watch?v=HEISXs0wYIM>

Al tratar de caracterizar la eficiencia de un algoritmo en términos del tiempo de ejecución, independientemente de cualquier programa o computadora, es importante cuantificar el número de operaciones o pasos que necesitará el algoritmo para cumplir con su propósito. Si se considera que cada uno de estos pasos es una unidad básica de cálculo, el tiempo de ejecución de un algoritmo puede expresarse como el número de pasos necesarios para resolver el problema. Este número de pasos dependerá de cómo se implemente el algoritmo para resolver el problema dado y la **notación Big O** lo que trata de medir son el número de pasos.

**Ejercicio:** Debes programar el problema que se plantea en la siguiente secuencia de videos en el lenguaje de programación que desees:

- 1.- <https://www.youtube.com/watch?v=GD254Gotp-4>
- 2.- <https://www.youtube.com/watch?v=MaY6FpP0FEU>

**NOTA:** Debes subir el ejercicio a tu repositorio GitHub como siempre.

**#HASHTAGS** (etiquetas de ayuda para búsqueda de información relevante)

#Notación-O-Grande #Big-O-Notation #algoritmos-de-ordenación #notación-asintótica #eficiencia #computación #algoritmia

## LINKS DE INTERÉS

<https://www.youtube.com/watch?v=HEISXs0wYIM>  
<https://www.youtube.com/watch?v=IZgOEC0NIbw>  
<https://www.youtube.com/watch?v=GD254Gotp-4>  
<https://www.youtube.com/watch?v=MaY6FpP0FEU>  
[https://medium.com/@charlie\\_fuentes/notacion-big-0-para-principiantes-f9cbb4b6bec8](https://medium.com/@charlie_fuentes/notacion-big-0-para-principiantes-f9cbb4b6bec8)  
<https://www.campusmvp.es/recursos/post/Rendimiento-de-algoritmos-y-notacion-Big-O.aspx>  
<https://www.youtube.com/watch?v=D6xkbGLQesk>

## DICCIONARIO

Notación-asintótica | Notación-Big-O | Algoritmos-de-ordenación

## PUNTUACIÓN

Programación: 4  
Redes: 2  
Seguridad: 2  
Algoritmia: 5