

# INFORME TIEMPOS DE EJECUCIÓN

Laura Silva Loureiro  
Programación II  
Práctica 3  
laura.silva.loureiro@rai.usc.es

El objetivo de este informe es el estudio de los tiempos de ejecución cuando realizamos la suma y el producto de dos matrices de dimensiones suficientemente grandes como para poder observar y exponer los correspondientes algoritmos en situaciones de estrés.

El programa está realizado en C y el código ha sido reutilizado de la Práctica 0, realizada en esta asignatura, basada en la manipulación de matrices con una estructura dinámica. Se ha modificado este código de tal forma que se ha creado un nuevo TAD matriz y se han implementado las correspondientes funciones ajustadas a esta nueva estructura. Otras modificaciones que se han realizado han sido el cambio de ints a longs que nos ha permitido trabajar con números más grandes y consecuentemente con más bytes.

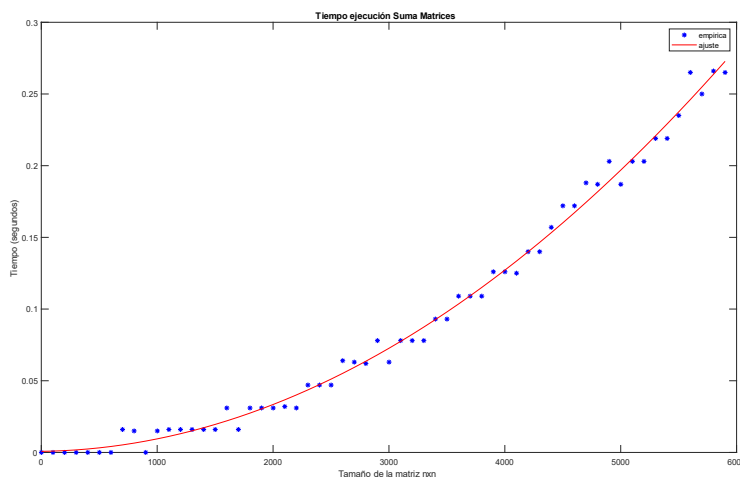
En ambas operaciones se ha experimentado con matrices cuadradas, inicializadas con valores aleatorios, en las cuales se ha ido modificando su tamaño hasta llegar a dimensiones considerables para poder observar el algoritmo en su estado de ‘estrés’.

La medición de los tiempos se ha implementado la librería `< time.h >` que contiene la función `clock_t` que permite medir el tiempo transcurrido al principio del proceso y al final, su resta será nuestro tiempo de ejecución.

## 1. Suma de matrices:

Para el cálculo del tiempo de ejecución de la suma se ha inicializado el tamaño de la matriz en  $1 \times 1$ , se ha aumentado la dimensión con un paso de 100 hasta alcanzar un tamaño máximo de  $6000 \times 6000$ . Los resultados obtenidos de esta medición se muestran en la Figura 1, en la cual se relaciona el tiempo (eje y) con la dimensión de la matriz (eje x).

La suma tiene una complejidad computacional de orden cuadrático (tal y como observamos en la Figura 1), ya que su código depende de dos interacciones que dependen del tamaño de la matriz.

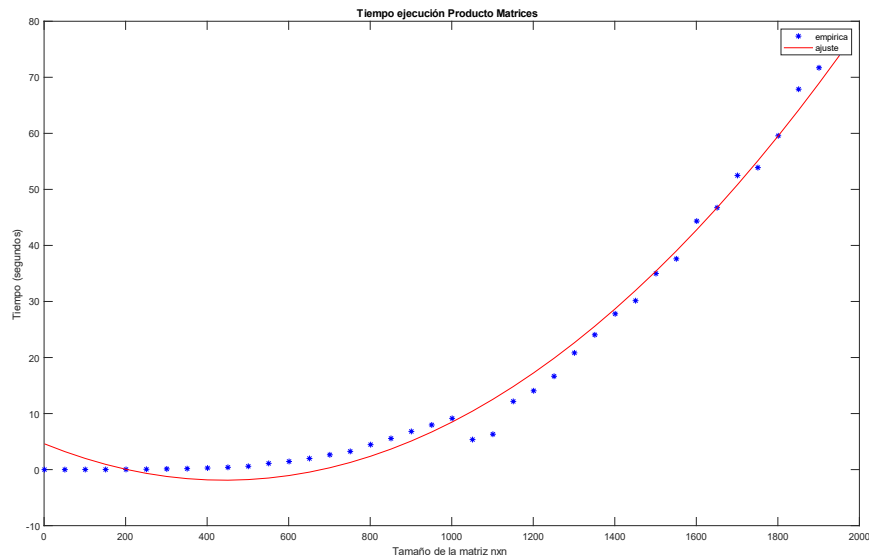


**Figura 1:** Gráfica de los tiempos de ejecución de la suma de matrices.

## 2. Producto de Matrices:

Para el cálculo del tiempo de ejecución del producto se ha inicializado el tamaño de la matriz en 1x1, se ha aumentado la dimensión con un paso de 50 hasta alcanzar un tamaño máximo de 2000x2000. Los resultados obtenidos de esta medición se muestran en la Figura 2, en la cual se relaciona el tiempo (eje y) con la dimensión de la matriz (eje x).

El producto tiene una complejidad computacional de orden cúbico (en la figura 3 se observa una gráfica correspondiente a una  $O(n^3)$ ), ya que su código depende de tres interacciones que dependen del tamaño de la matriz (tres bucles/lazos for).



**Figura 2:** Gráfica de los tiempos de ejecución del producto de matrices.

## 3. Conclusiones obtenidas:

El análisis de los algoritmos es muy importante a la hora de realizar programas, especialmente si nuestros recursos en tiempo o en memoria son limitados. En este caso, se están comparando dos algoritmos distintos: la suma y el producto de dos matrices. En un primer momento, de manera teórica podríamos predecir que la suma fuese más rápida en el tiempo ya que solo necesita dos interacciones (complejidad cuadrática), sin embargo, el producto necesita tres (orden cúbico).

Este hecho se ha cumplido en nuestra fase experimental, ya que al sumar dos matrices de nuestro tamaño máximo (6000x6000) ha llevado 0.3s y multiplicar dos matrices de 2000x2000 ha llevado 74.3s.

La eficiencia de un algoritmo está basada en su capacidad de resolver problemas empleando un bajo consumo de recurso computacionales, relacionados con el coste espacial (memoria requerida) y el coste temporal. Un algoritmo será mejor que otro cuando es capaz de solucionar un problema en menos tiempo y/o empleando menos cantidad de memoria que otro.

Por ello, hemos estudiado la relación del tiempo con respecto a las dimensiones de nuestras matrices, y hemos observado que para llevar a cabo el producto de matrices se necesita mucho más tiempo que para la suma.

Por último, cabe mencionar que los resultados dependen del procesador utilizado, en otro podríamos obtener resultados totalmente distintos. Los datos mostrados en las gráficas se corresponden a las pruebas realizadas en un único dispositivo.

#### 4. Preguntas propuestas:

- *¿qué tamaños de matrices son gestionables en tiempo “razonable” por cada algoritmo?*

Un tiempo razonable para estas operaciones podría ser considerado de unos 10 segundos. Teniendo en cuenta este dato, en la suma el tamaño máximo que se le podría dar para que esté dentro de nuestros ‘límites de tiempo’ sería de 15000x15000. En cuanto al producto las dimensiones de las matrices serían de 1100x1100.

- *¿Es más eficiente en general alguna de las dos operaciones? En caso afirmativo, indica si es así para todos los valores de  $n$  o únicamente para algunos. En este último caso, indica el rango de valores de  $n$  para los que es más ágil cada operación.*

Para poder comparar la eficiencia de los dos algoritmos deberían resolver el mismo problema, es decir, en este caso realizar la misma operación matemática. En general, podríamos decir que la suma sería más eficiente que el producto ya que conlleva menos tiempo y necesita de menos memoria para operar. Además, el hecho de que la suma tenga complejidad cuadrática y el producto complejidad cúbica, también nos permiten decantarnos por la suma.

El rango de valores de  $n$ /tamaño para los que es más ágil cada operación sería para valores muy pequeños en el caso del producto en un rango [1,1000] y en el caso de la suma [1,15000].