



**Instituto Politecnico Nacional**



**ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”**

*DESARROLLO DE SISTEMAS DISTRIBUIDOS*

*TAREA 2: Uso eficiente de la memoria caché*

PROFE: PINEDA GUERRERO CARLOS

ALUMNO: Rojas Alvarado Luis Enrique

GRUPO: 4CM5

## 1.- Objetivo

Compilar y ejecutar los programas de MultiplicaMatriz.java y MultiplicaMatriz\_2.java proporcionados por el profesor para comparar la eficiencia (el tiempo en el que logra hacer la multiplicación de matrices) de estos, para un tamaño de matriz de 100, 200, 300, 500 y 1000.

## 2.- Pruebas

Se compila y se ejecutan ambos programas para comprobar su eficiencia.\*

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 3567ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 922ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

Debemos tomar en cuenta que java almacena las matrices en forma de renglones.

El programa MultiplicaMatriz.java, cada que se accede a la matriz B, se transfiere una línea completa de la RAM a la caché, lo que hace menos eficiente. El acceso a la matriz A es más eficiente debido a que sus elementos se leen secuencialmente (por renglones), como se encuentra almacenada en memoria.

En el programa MultiplicaMatriz\_2.java se modifica de manera que el acceso a la matriz B sea por renglones y no por columnas, simplemente intercambiando los índices con los que se accede a la matriz B (se transpone la matriz). Logrando de esta manera un acceso a los elementos de la matriz B más eficiente, debido a que ahora se leen en forma de renglones y no por columnas (como está guardada en memoria).

### 2.1. Diferentes tamaños de matriz.

Se mide la eficiencia del programa MultiplicaMatriz.java con el tamaño de matriz de 100:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 12ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

200:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 28ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

300:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 93ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

500:

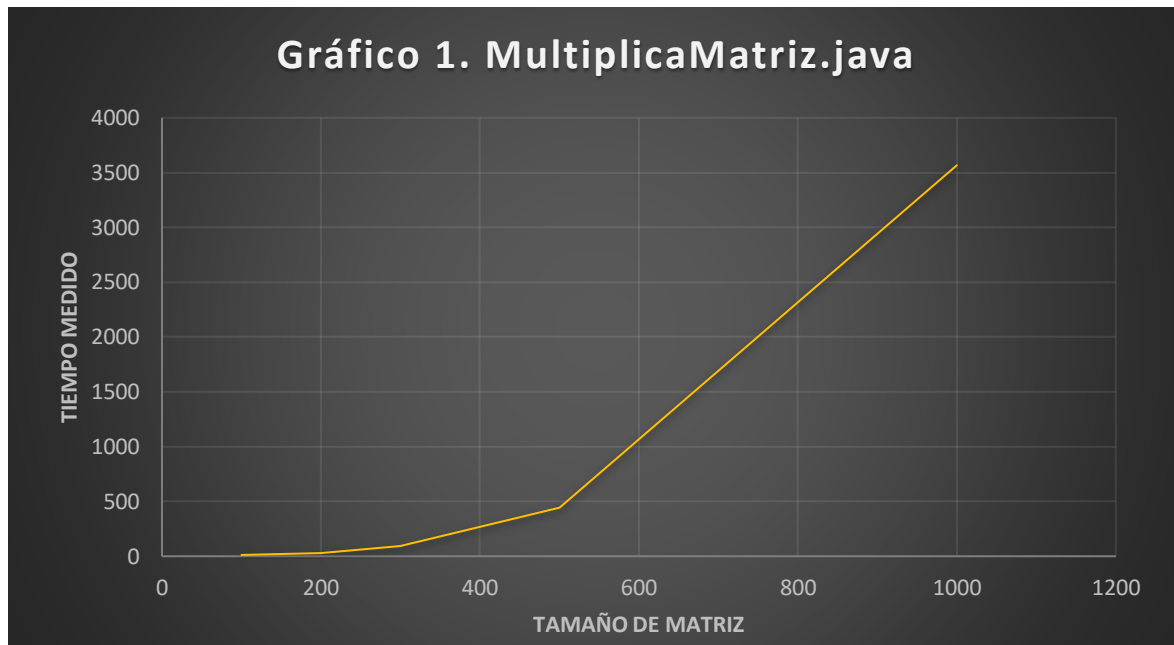
```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 444ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

1000:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz
Tiempo: 3567ms
```

Tabla 1. Eficiencia MultiplicaMatriz.java

TAMAÑO DE MATRIZ	TIEMPO MEDIDO
100	12ms
200	28ms
300	93ms
500	444ms
1000	3567ms



Se mide la eficiencia del programa MultiplicaMatriz\_2.java con el tamaño de matriz de 100:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 13ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

200:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 24ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

300:

```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 42ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

500:

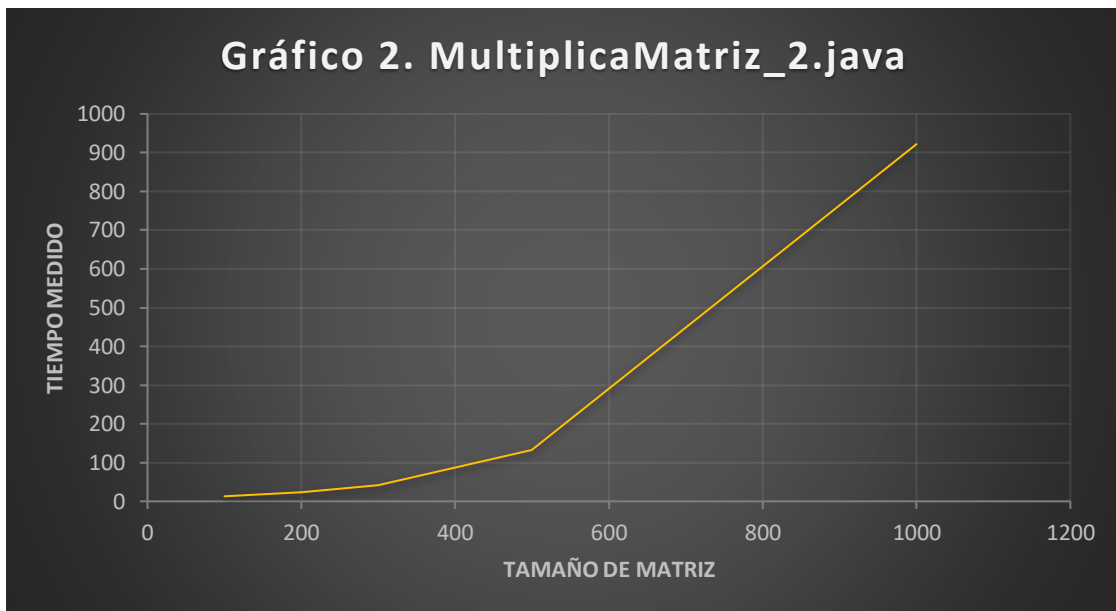
```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 133ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

1000:

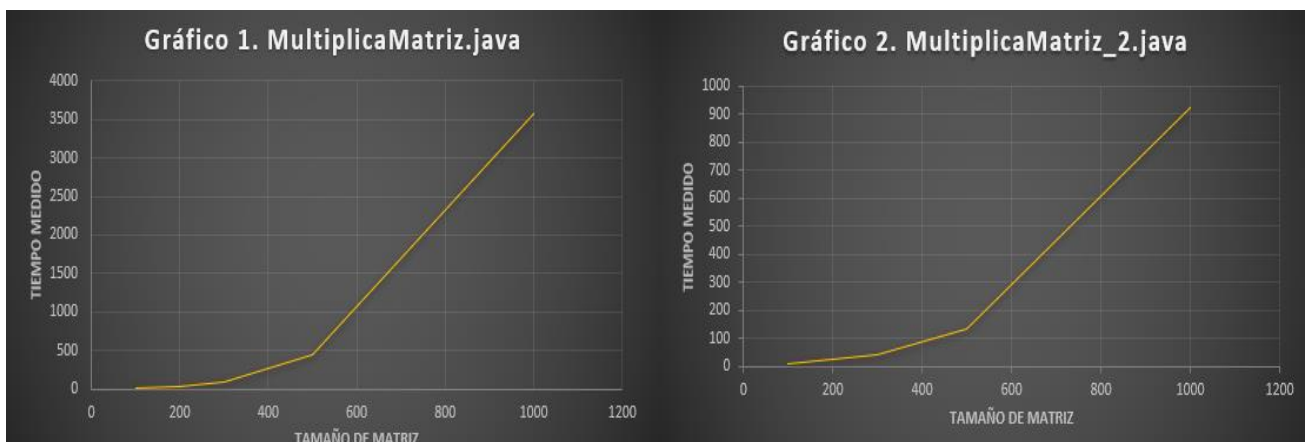
```
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> javac MultiplicaMatriz_2.java
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> java MultiplicaMatriz_2
Tiempo: 922ms
PS C:\Users\ameri\Documentos\ESCOM\7semestre\Distribuidos\clase05-oct> |
```

Tabla 2. Eficiencia MultiplicaMatriz\_2.java

TAMAÑO DE MATRIZ	TIEMPO MEDIDO
100	13ms
200	24ms
300	42ms
500	133
1000	922ms



Forma 1. Comparativo de ambos programas:



### 3.- ESPECIFICACIONES TÉCNICAS DE EQUIPO\*

El equipo que fue utilizado para esta cuenta con las siguientes especificaciones:

- Procesador intel core i7-8550U 1.8GHz hasta 4.0 GHz
- Memoria caché 8 M
- Memoria RAM de 12 GB DDR4

#### Ver información básica acerca del equipo

##### Edición de Windows

Windows 10 Home Single Language

© 2019 Microsoft Corporation. Todos los derechos reservados.



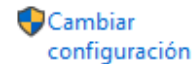
# Windows 10

##### Sistema

Procesador:	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
Memoria instalada (RAM):	12.0 GB (11.9 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

##### Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre del equipo:	DESKTOP-496B8RB
Nombre completo de equipo:	DESKTOP-496B8RB
Descripción del equipo:	
Grupo de trabajo:	WORKGROUP



##### Activación de Windows

Windows está activado [Lee los Términos de licencia del software de Microsoft](#)

## ANEXO1

### MultiplicaMatriz.java

```
1. import java.io.*;
2.
3. class MultiplicaMatriz{
4.     static int N = 300;
5.     static int[][] A = new int[N][N];
6.     static int[][] B = new int[N][N];
7.     static int[][] C = new int[N][N];
8.
9.     public static void main(String[] args)
10.    {
11.        long t1 = System.currentTimeMillis();
```

```

12.
13.     // inicializa las matrices A y B
14.
15.     for (int i = 0; i < N; i++)
16.         for (int j = 0; j < N; j++)
17.             {
18.                 A[i][j] = 2 * i - j;
19.                 B[i][j] = i + 2 * j;
20.                 C[i][j] = 0;
21.             }
22.
23.     // multiplica la matriz A y la matriz B, el resultado queda en la matriz C
24.
25.     for (int i = 0; i < N; i++)
26.         for (int j = 0; j < N; j++)
27.             for (int k = 0; k < N; k++)
28.                 C[i][j] += A[i][k] * B[k][j];
29.
30.     long t2 = System.currentTimeMillis();
31.     System.out.println("Tiempo: " + (t2 - t1) + "ms");
32. }
33. }

```

## ANEXO 2

### MultiplicaMatriz\_2.java

```

1. import java.io.*;
2.
3. class MultiplicaMatriz_2
4. {
5.     static int N = 500;
6.     static int[][] A = new int[N][N];
7.     static int[][] B = new int[N][N];
8.     static int[][] C = new int[N][N];
9.
10.    public static void main(String[] args)
11.    {
12.        long t1 = System.currentTimeMillis();
13.
14.        // inicializa las matrices A y B
15.
16.        for (int i = 0; i < N; i++)
17.            for (int j = 0; j < N; j++)
18.                {
19.                    A[i][j] = 2 * i - j;
20.                    B[i][j] = i + 2 * j;
21.                    C[i][j] = 0;
22.                }
23.
24.        // transpone la matriz B, la matriz traspuesta queda en B
25.
26.        for (int i = 0; i < N; i++)
27.            for (int j = 0; j < i; j++)
28.                {
29.                    int x = B[i][j];
30.                    B[i][j] = B[j][i];

```

```
31.         B[j][i] = x;
32.     }
33.
34.     // multiplica la matriz A y la matriz B, el resultado queda en la matriz C
35.     // notar que los indices de la matriz B se han intercambiado
36.
37.     for (int i = 0; i < N; i++)
38.         for (int j = 0; j < N; j++)
39.             for (int k = 0; k < N; k++)
40.                 C[i][j] += A[i][k] * B[j][k];
41.
42.     long t2 = System.currentTimeMillis();
43.     System.out.println("Tiempo: " + (t2 - t1) + "ms");
44. }
45. }
```