

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN
ALGORITMOS Y ESTRUCTURAS DE DATOS



Laboratorio Nro. 01

Presentado por:

Fiorela Villarroel Ramos

Docente :

YESSENIA DEYSI YARI RAMOS



Práctica de Hilos

1. Ejercicio 1

Hacer una calculadora de matrices (adición , sustracción y multiplicación) de matrices cuadradas.

- **Implementación 1:** Manera tradicional
- **Implementación 2:** Usando hilos
- Medir los tiempos de cada operación y realizar una comparación.

1.1. Implemetación 1:

1.1.1. Código:

La implementación de la calculadora de matrices de forma tradicional se encuentra en el siguiente enlace **Click Aquí**

1.1.2. Entradas :

```
villa7523@LAPTOP-E0T5UEV5:/mnt/d/unsas/so/hilos$ g++ -o a.exe traditionalWay.cpp
villa7523@LAPTOP-E0T5UEV5:/mnt/d/unsas/so/hilos$ ./a.exe
8
84 87 78 16 94 36 87 93
50 22 63 28 91 60 64 27
41 27 73 37 12 69 68 30
83 31 63 24 68 36 30 3
23 59 70 68 94 57 12 43
30 74 22 20 85 38 99 25
16 71 14 27 92 81 57 74
63 71 97 82 6 26 85 28

37 6 47 30 14 58 25 96
83 46 15 68 35 65 44 51
88 9 77 79 89 85 4 52
55 100 33 61 77 69 40 13
27 87 95 40 96 71 35 79
68 2 98 3 18 93 53 57
2 81 87 42 66 90 45 20
41 30 32 18 98 72 82 76
```

Figura 1: Matrices a operar

1.1.3. Resultados:

- Resultado de la Adición:

```
Resultado de la Suma(FORMA TRADICIONAL)

121 93 125 46 108 94 112 189
133 68 78 96 126 125 108 78
129 36 150 116 101 154 72 82
138 131 96 85 145 105 70 16
50 146 165 108 190 128 47 122
98 76 120 23 103 131 152 82
18 152 101 69 158 171 102 94
104 101 129 100 104 98 167 104
```

Figura 2: Adición

- Resultado de la Sustracción:

```
Resultado de la Resta(FORMA TRADICIONAL)

47 81 31 -14 80 -22 62 -3
-33 -24 48 -40 56 -5 20 -24
-47 18 -4 -42 -77 -16 64 -22
28 -69 30 -37 -9 -33 -10 -10
-4 -28 -25 28 -2 -14 -23 -36
-38 72 -76 17 67 -55 46 -32
14 -10 -73 -15 26 -9 12 54
22 41 65 64 -92 -46 3 -48
```

Figura 3: Sustracción

- Resultado del Producto:

```
Resultado del Producto(FORMA TRADICIONAL)

23280 22186 31845 23082 27889 36091 16055 27980
17392 17876 28596 16149 23329 29413 12855 21427
17322 12553 22988 14591 17670 26264 11186 15809
16880 13240 22637 15090 18849 24683 10279 21151
22082 19518 25135 18843 24693 28852 12999 20620
15327 21344 25210 16579 21953 28224 14384 18887
17322 18961 25350 14363 20112 27782 14750 19315
23392 20217 24779 23335 24876 32620 13783 19387
```

Figura 4: Producto

1.2. Implementación 2:

La implementación de la calculadora de matrices usando hilos se encuentra en el siguiente enlace [Click Aquí](#)

1.2.1. Entradas :

```
villa7523@LAPTOP-E0T5UEV5:/mnt/d/unsas/so/hilos$ gcc -o a.exe PthreadWay.c -lpthread
villa7523@LAPTOP-E0T5UEV5:/mnt/d/unsas/so/hilos$ ./a.exe
8
84 87 78 16 94 36 87 93
50 22 63 28 91 60 64 27
41 27 73 37 12 69 68 30
83 31 63 24 68 36 30 3
23 59 70 68 94 57 12 43
30 74 22 20 85 38 99 25
16 71 14 27 92 81 57 74
63 71 97 82 6 26 85 28

37 6 47 30 14 58 25 96
83 46 15 68 35 65 44 51
88 9 77 79 89 85 4 52
55 100 33 61 77 69 40 13
27 87 95 40 96 71 35 79
68 2 98 3 18 93 53 57
2 81 87 42 66 90 45 20
41 30 32 18 98 72 82 76
```

Figura 5: Matrices a operar

1.2.2. Resultados:

- Resultado de la Adición:

```
Resultado de la Adición(USANDO HILOS)
121 93 125 46 108 94 112 189
133 68 78 96 126 125 108 78
129 36 150 116 101 154 72 82
138 131 96 85 145 105 70 16
50 146 165 108 190 128 47 122
98 76 120 23 103 131 152 82
18 152 101 69 158 171 102 94
104 101 129 100 104 98 167 104
```

Figura 6: Adición

- Resultado de la Sustracción:

```

Resultado de la Sustracción(USANDO HILOS)

47 81 31 -14 80 -22 62 -3
-33 -24 48 -40 56 -5 20 -24
-47 18 -4 -42 -77 -16 64 -22
28 -69 30 -37 -9 -33 -10 -10
-4 -28 -25 28 -2 -14 -23 -36
-38 72 -76 17 67 -55 46 -32
14 -10 -73 -15 26 -9 12 54
22 41 65 64 -92 -46 3 -48
    
```

Figura 7: Sustracción

- Resultado del Producto:

```

Resultado de la Producto(USANDO HILOS)

27046 24895 34790 24770 36923 42809 23619 35051
18532 18710 29412 16675 25919 31362 15049 23503
18599 13435 23952 15173 20687 28440 13582 18111
16975 13399 22703 15181 19152 24932 10535 21389
23849 20836 26536 19589 28909 31962 16548 23924
16390 22022 26086 17012 24336 30079 16388 20819
20342 21191 27791 15710 27338 33119 20806 24885
24518 21016 25610 23775 27712 34682 16076 21563
    
```

Figura 8: Producto

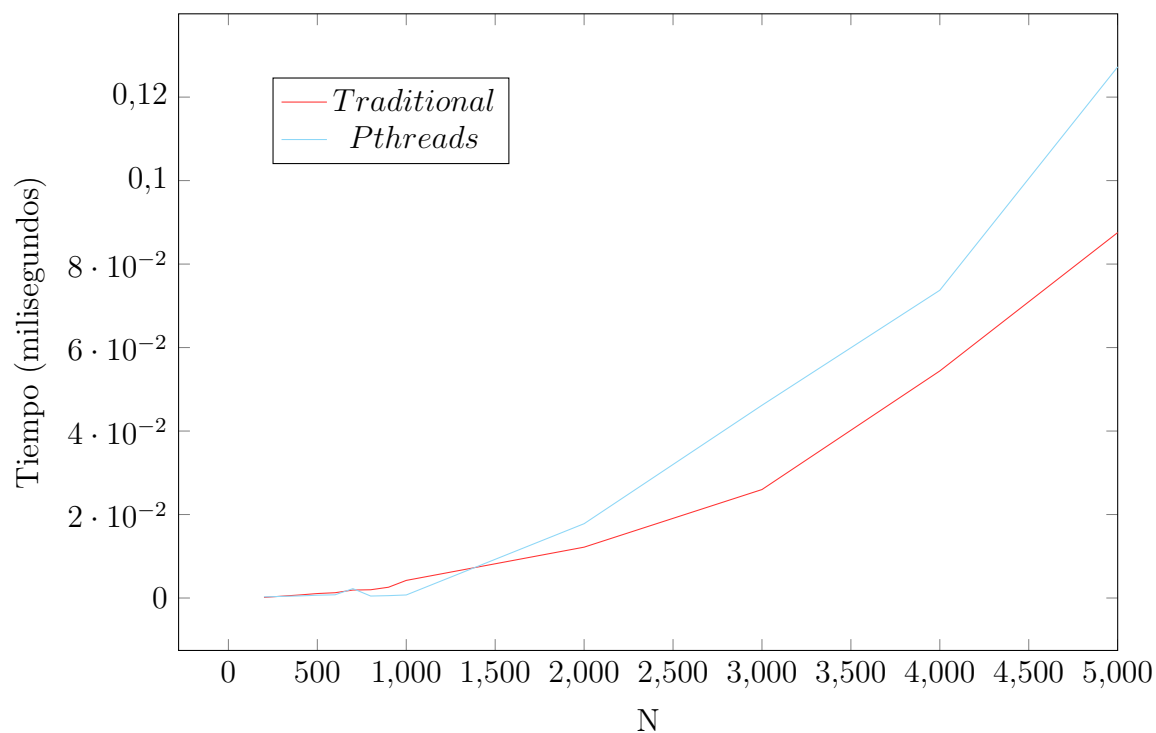
1.3. Medir los tiempos de cada operación y realizar una comparación.

Información de la CPU	
Velocidad de base	1.80 GHz
Sockets	1
Núcleos	4
Procesadores Lógicos	8

1.3.1. Explicación de la implementación

Para la implementación de la calculadora de matrices usando la librería **Pthread**, se ha considerado la cantidad de núcleos del equipo el cual es de 4, con la finalidad de que los resultados del tiempo de ejecución sean notables, por lo que la matriz se ha dividido entre 4, donde cada hilo trabaja con la cuarta parte de las matrices, así las tareas están divididas entre los núcleos.

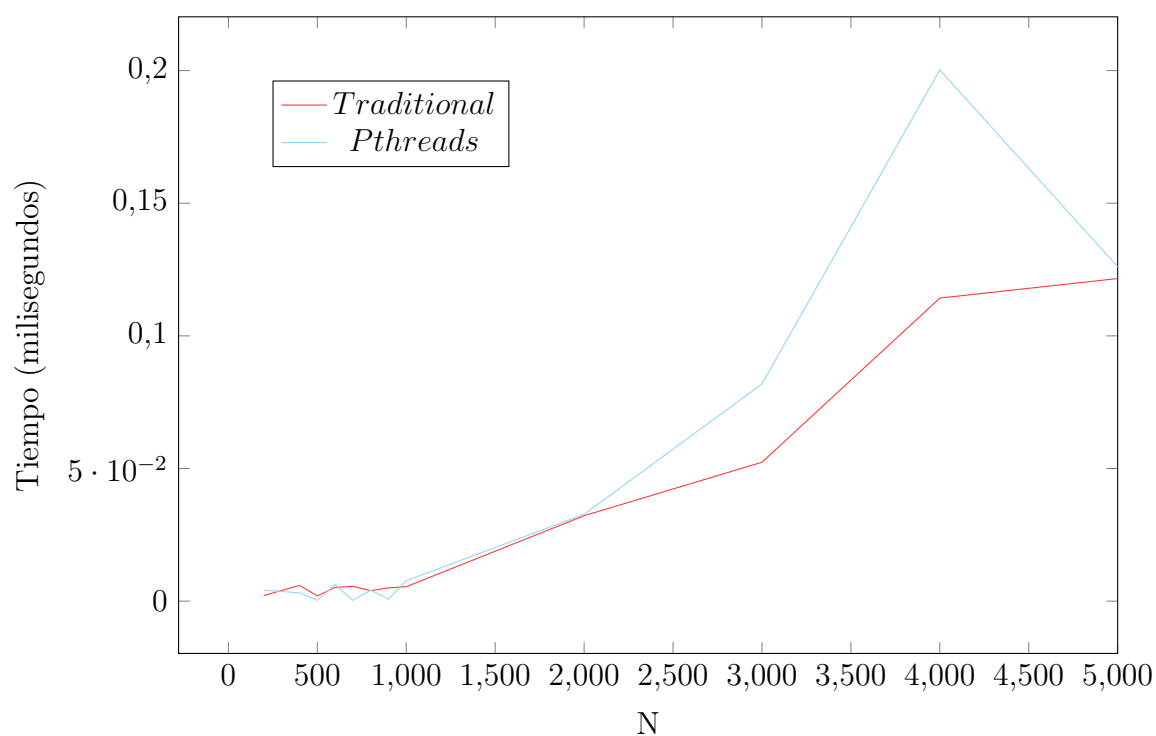
1.3.2. Sustracción de matrices:



1.3.2.1. Análisis y comparación de los tiempos de ejecución

Tiempos de ejecución de sustracción entre matrices		
N	Pthreads	Traditional
200	0.000272	0.000148
400	0.000497	0.00073
500	0.000668	0.001069
600	0.000772	0.001267
700	0.002245	0.001906
800	0.000472	0.001978
900	0.000566	0.002581
1000	0.000731	0.004217
2000	0.017805	0.012184
3000	0.046175	0.025984
4000	0.07373	0.0544
5000	0.127243	0.087546

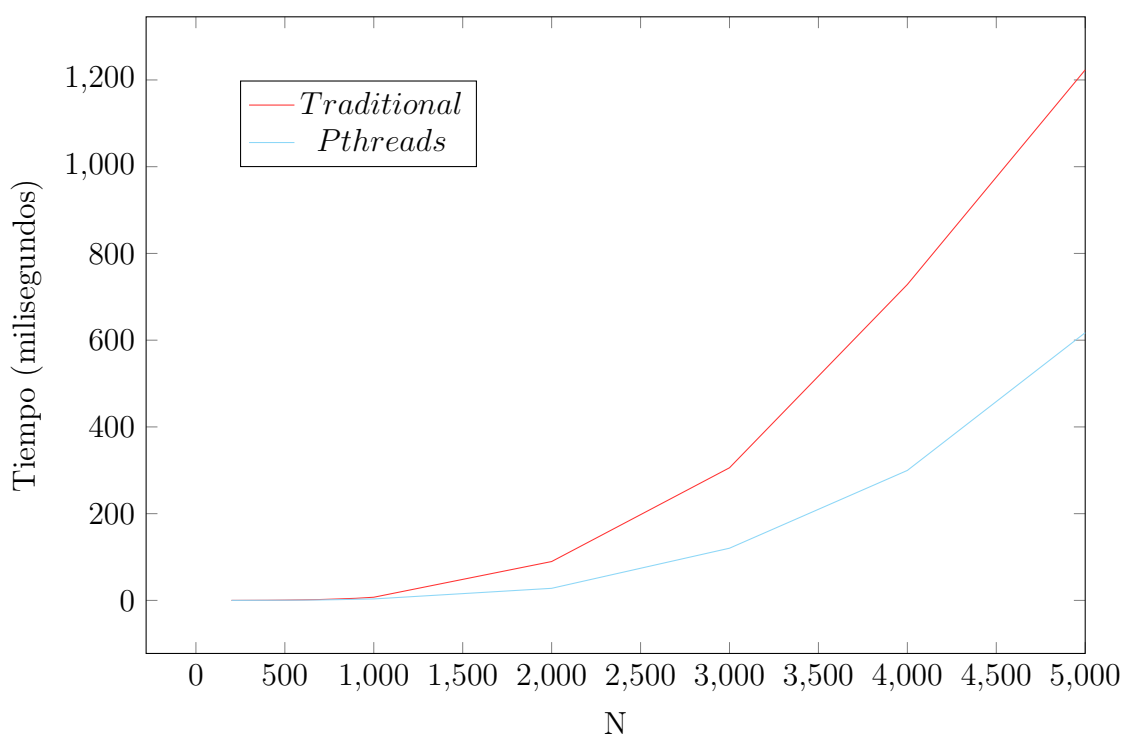
1.3.3. Adición de matrices:



1.3.3.1. Análisis y comparación de los tiempos de ejecución

Tiempos de ejecución de Adición entre matrices		
N	Traditional	Pthreads
200	0.004143	0.002138
400	0.003143	0.005932
500	0.000335	0.001981
600	0.006242	0.005197
700	0.000284	0.005579
800	0.004201	0.003953
900	0.000777	0.004992
1000	0.007706	0.005447
2000	0.032747	0.032206
3000	0.081867	0.052364
4000	0.200264	0.114243
5000	0.126012	0.121612

1.3.4. Producto de matrices:



1.3.4.1. Análisis y comparación de los tiempos de ejecución

Tiempos de ejecución de producto entre matrices		
N	Traditional	Pthreads
200	0.014979	0.033142
400	0.235642	0.301496
500	0.397973	0.646843
600	0.665479	1.224661
700	1.110301	2.042047
800	1.73077	3.355926
900	2.632725	4.846891
1000	3.387945	7.301147
2000	27.85369	89.616728
3000	120.337791	305.786948
4000	299.672949	728.381

1.3.5. Análisis de resultados

Como se puede observar tanto en la gráfica 1 , 2 como en las tablas 1 , 2 , la implementación en la forma tradicional es más veloz que la implementación que hace uso de hilos tanto en las operaciones de adición como sustracción , esto se debe a que la cantidad de datos no es demasiado grande por lo que las operaciones son rápidas , solo

hace un recorrido de $n \times n$ y en el tamaño máximo de data es de 5000 dado un resultado de 25000000 elementos que han de ser visitados , además sabemos que solo se puede hacer una tarea por núcleo por lo que al contar con solo 4 núcleos sabiendo que uno usa el sistema entonces no tenemos la certeza en que momento un núcleo esta vacío para ejecutar nuestra tarea , sin embargo en **Figura 3** se observa una notable diferencia , notando que la implementación haciendo el uso de hilos es mucho más veloz que la tradicional , esto se debe a que el producto entre matrices no solo es recorrer filas de ambas matrices para realizar las operaciones , si no que se debe recorrer una fila de una matriz por las demás columnas de la otra matriz lo que genera una complejidad de $O(n^3)$ en nuestra data , con el tamaño más grande que hay que es de 5000 elementos seria recorrer unos 125000000000 elementos lo que es demasiado por lo que dividir la tarea en hilos es mucho más optima que hacerlo tradicionalmente. .