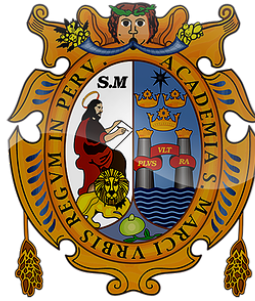


” Año del Fortalecimiento de la Soberanía Nacional”



**Universidad Nacional Mayor de San Marcos**  
(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA Escuela  
Profesional de Ingeniería de Software



**Primera EC: Laboratorio Open MP**

**Curso:** Programación Paralela y Concurrente

**Profesor:** Edson Ticona Zegarra

**Integrantes:**

- |  |          |
|--|----------|
| ■ Pichilingue Pimentel, Nathaly Nicole | 19200247 |
| ■ Torre Arteaga, Alexander             | 19200246 |
| ■ Ricse Perez, Anthony Elias           | 19200276 |

Ciudad Universitaria 2022

### Entorno de pruebas:

- Procesador: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
- Memoria RAM: 16 GB
- Número de procesadores: 8
- Sistema Operativo: Windows 10 Home
- IDE para pruebas: Dev C++
- IDE de desarrollo: Visual Studio Code

### Versión Secuencial

Ejecutamos la versión secuencial 10 veces para, en base a ello, poder analizar el tiempo de ejecución en gráficas.

Vez	Cantidad
1	0.0362661
2	0.0362704
3	0.0362803
4	0.0367448
5	0.0357731
6	0.0360422
7	0.0176593
8	0.0388498
9	0.0357934
10	0.0353939

Tabla 1: Tiempos de ejecución versión secuencial

<pre>D:\natha\Documents\Docs Natha 3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0362661s</pre>	<pre>3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0362704s</pre>
<pre>3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0362803s</pre>	<pre>3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0367448s</pre>
<pre>3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0357731s</pre>	<pre>3 5 0.001 (x+2)^2 72.6735 0.00087661 14 elapsed time: 0.0360422s</pre>

Figura 1: Ejecucion 1 a 6 en orden de izquierda a derecha

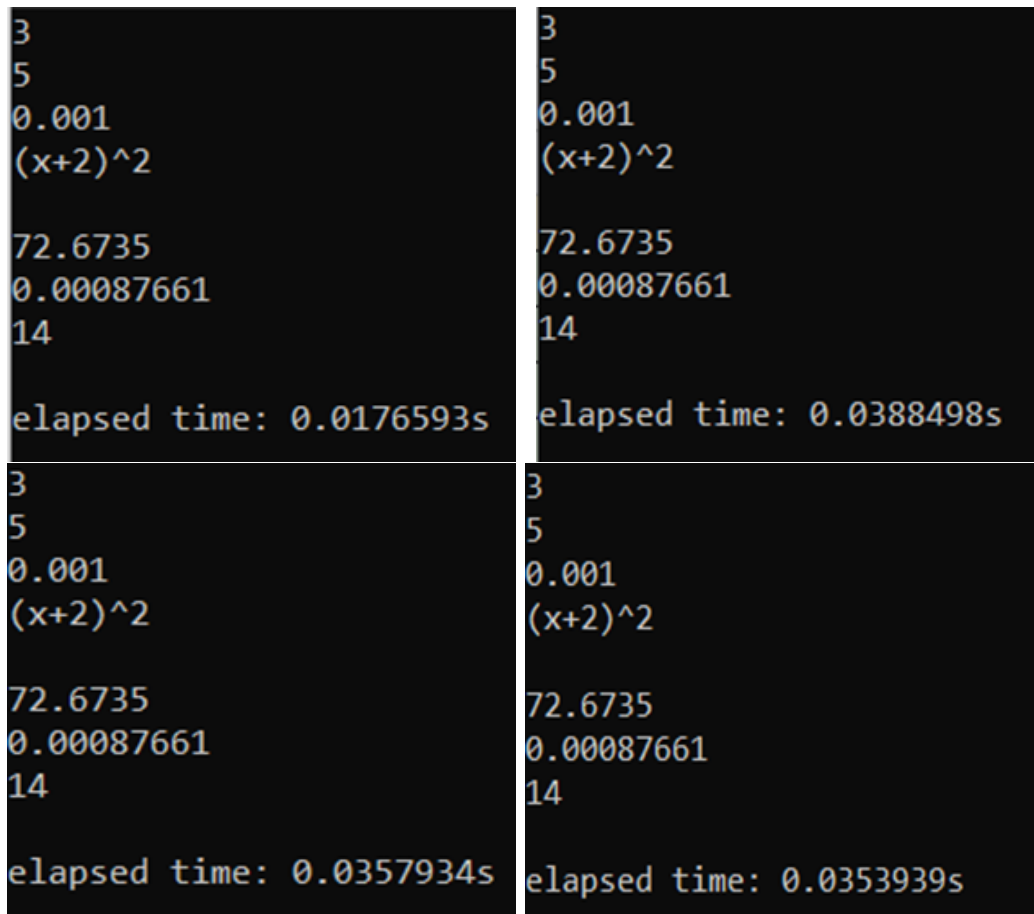


Figura 2: Ejecucion 7 a 10 en orden de izquierda a derecha

Y en gráfico de dispersión obtenemos el siguiente (Realizado con Python en Google Colab):

Y en gráfico de dispersión obtenemos el siguiente (Realizado con Python en Google Colab):

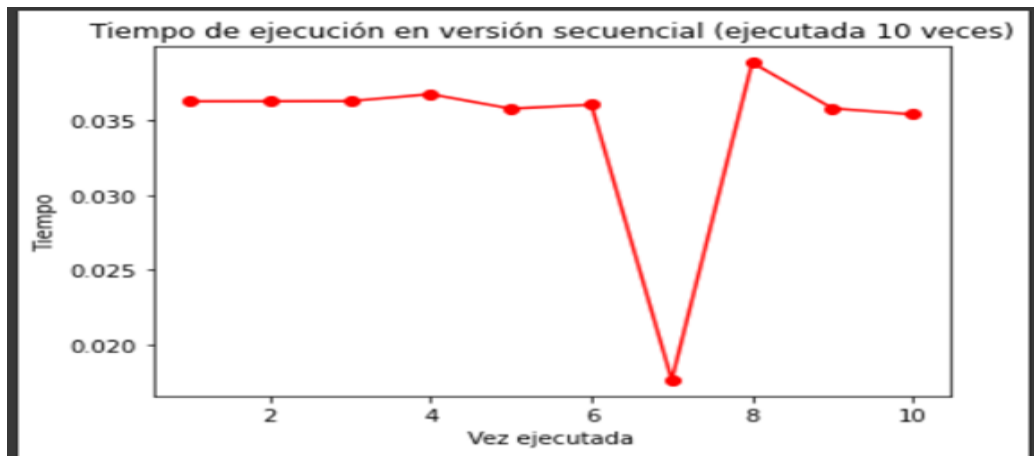


Figura 3: Tiempo de ejecución en version secuencial (ejecutada 10 veces)

Podemos observar que el tiempo de ejecución es en promedio 0.036 segundos, teniendo un valor divergente en la 7ma ejecución del programa donde se obtuvo un tiempo de 0.017 segundos.

### Versión Paralela

Ejecutamos la versión paralela 10 veces para, en base a ello, poder analizar el tiempo de ejecución en gráficas.

Vez	Cantidad
1	0.0144939
2	0.0142298
3	0.0143356
4	0.0149603
5	0.0154302
6	0.0146521
7	0.0140904
8	0.0161136
9	0.0144786
10	0.0142954

Tabla 2: Tiempos de ejecución versión paralela

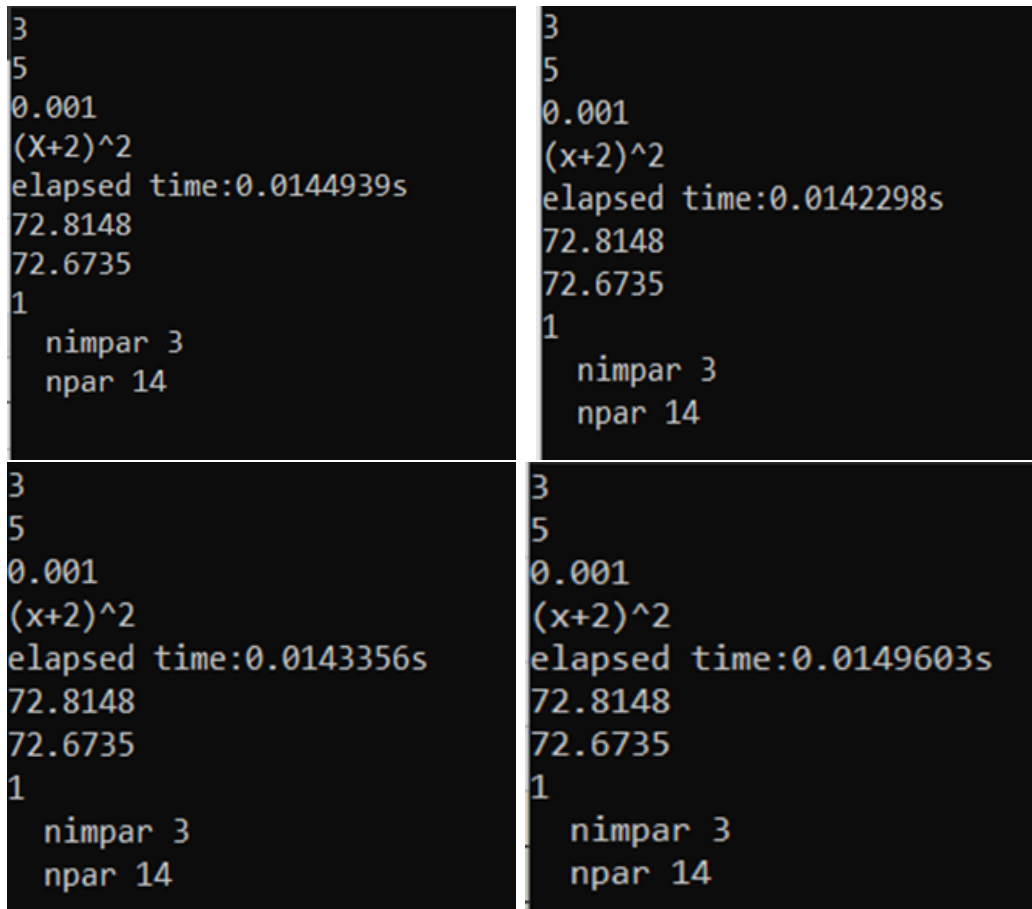


Figura 4: Ejecucion 12 a 15 en orden de izquierda a derecha

<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0154302s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>	<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0146521s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>
<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0140904s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>	<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0161136s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>
<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0144786s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>	<pre> 3 5 0.001 (x+2)^2 elapsed time:0.0142954s 72.8148 72.6735 1   nimpar 3   npar 14 </pre>

Figura 5: Ejecucion 16 a 21 en orden de izquierda a derecha

Y en gráfico de dispersión obtenemos el siguiente (Realizado con Python en Google Colab):

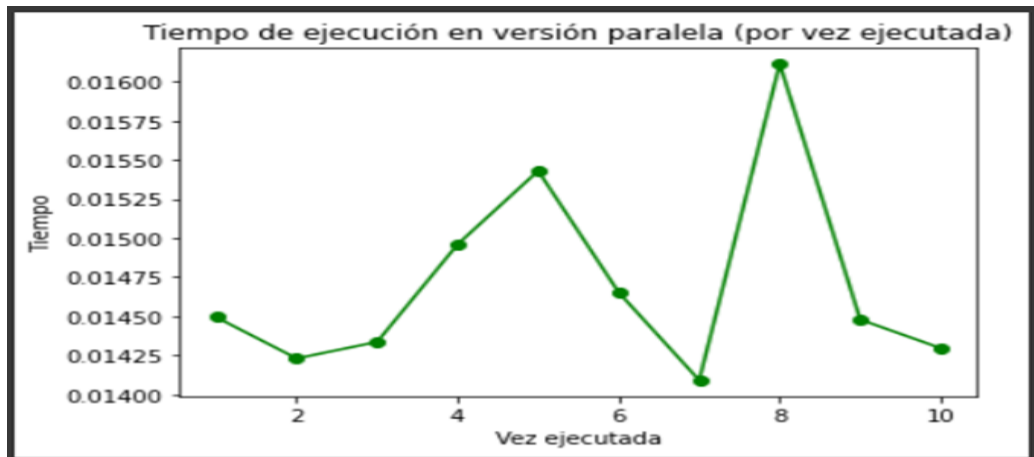


Figura 6: Tiempo de ejecución en version paralela

Podemos observar que el tiempo de ejecución es en promedio 0.014 segundos, teniendo dos valores divergentes en la 5ta y 8va vez de ejecución, de 0.015 y 0.018 segundos respectivamente.

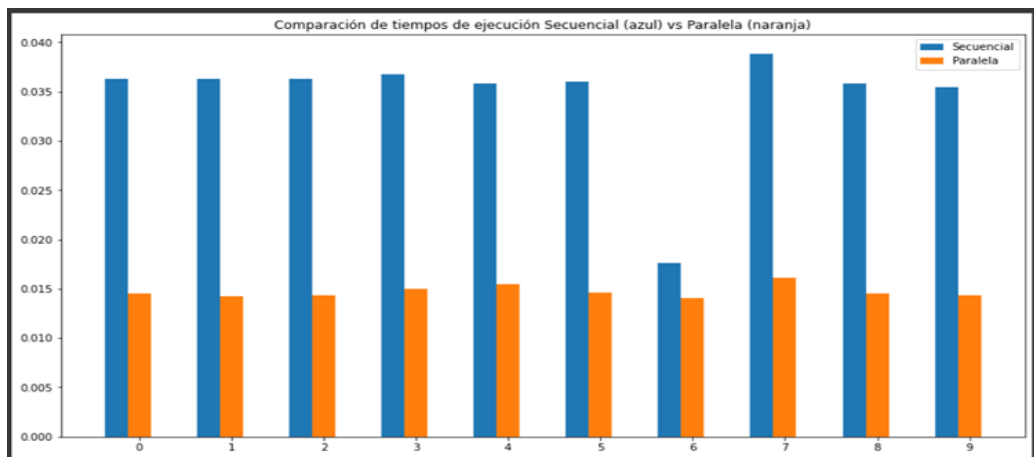


Figura 7: Comparacion de tiempos algoritmos secuencial vs paralelo



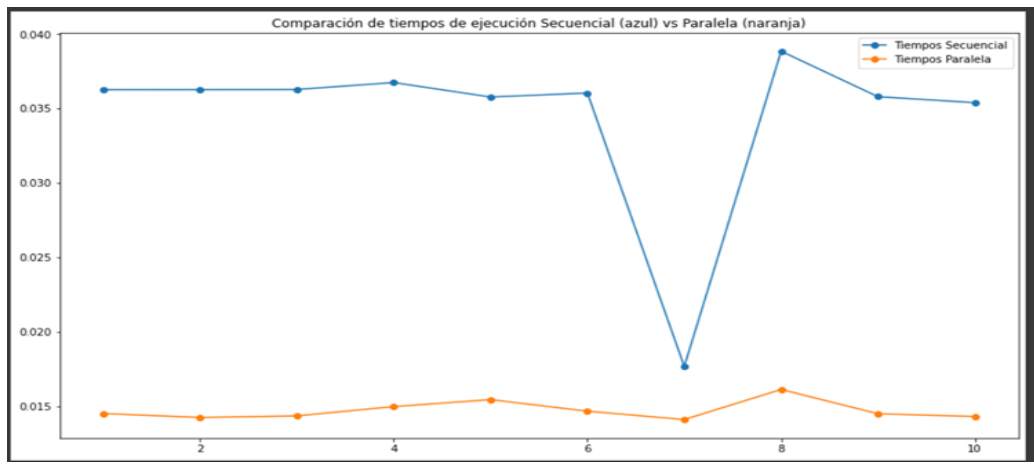


Figura 8: Comparacion de tiempos algoritmo secuencial vs paralelo

Se realizó un gráfico de barras comparando los tiempos de ejecución entre la versión Secuencial y la versión Paralela, donde se encontró por amplio margen que la versión paralela tiene un tiempo de ejecución mucho menor que la versión secuencial, es decir, es más rápido al realizar la ejecución del programa, esto debido a que, en el método del trapecio, la versión paralela divide en distintas particiones (como pares e impares) y ejecuta esas varias particiones en simultáneo, no debe esperar a la anterior, por lo tanto, obtiene la solución mucho más rápido.