

**Pontificia Universidad Católica Madre y Maestra  
(PUCMM)**



**Paralelización de Algoritmos y Análisis de Rendimiento**

**Integrantes:**

José Jáquez

Vianny Cruz

**Asignatura:**

Programación Paralela y Concurrente

**Profesor:**

Prof. Freddy Peña

*Santiago de los caballeros*

*16 de junio de 2024*

**Link del repositorio de la práctica:**

<https://github.com/Wolflags/ICC-303/tree/main/practica2>

## **Práctica #2**

### **Actividades**

#### **Parte 1: Ley de Amdahl**

Sea un programa con un tiempo de ejecución secuencial de 120 unidades de tiempo, y se sabe que el 85% de su código es paralelizable.

1. Calcula el tiempo de ejecución paralelo  **$T_p(n)$** , el speedup  **$S(n)$**  y la eficiencia  **$E(n)$**  cuando se ejecuta  $n=1,3,6,9,12$  procesadores.
2. Completa la siguiente tabla:

<b>Procesadores n</b>	<b><math>T_p(n)</math></b>	<b><math>S(n)</math></b>	<b><math>E(n)</math></b>
<b>1</b>	120	1	1
<b>2</b>	69	1.739	0.870
<b>4</b>	43.5	2.759	0.690
<b>8</b>	30.75	3.902	0.488
<b>16</b>	24.38	4.922	0.308

## Parte 2: Programación

Implementación de Algoritmos Paralelos:

Usando Java 21 o superior realiza las siguientes tareas.

Algoritmo a paralelizar: Suma de un arreglo de 1,000,000 números enteros random comprendido entre 1 y 10,000.

1. Genera un archivo con 1,000,000 de registros comprendido entre 1 y 10,000, el cual deberá usar como base para los demás cálculos.
2. Escribe un programa secuencial que sume los elementos de un arreglo de un millón de enteros.
3. Modifica tu programa para que use múltiples hilos o procesos para realizar la suma en paralelo. Divide el arreglo en partes iguales para cada hilo/proceso.
4. Mide y compara el tiempo de ejecución del programa secuencial y del programa paralelo con 2, 4 y 8 hilos/procesos.
5. Completa la siguiente tabla con los tiempos medidos:

Número de Hilos/Procesos	Tiempo de Ejecución Secuencial (s)	Tiempo de Ejecución Paralelo (s)	Speedup	Eficiencia
1 (Secuencial)	0.2200433	0.2003148	1.1	1.1
2	0.2447075	0.1933828	1.27	0.64
4	0.2560918	0.2035741	1.25	0.31
8	0.2294286	0.3272187	0.70	0.09
16	0.245552	0.452763	0.54	0.03
32	0.2392466	0.6790839	0.35	0.01