

# Práctica 2.2

## CONCEPTOS FUNDAMENTALES DE PARALELISMO

Jordi Blasco Lozano

Razonamiento y representación del conocimiento

Grado en Inteligencia Artificial

---

## Indice:

<b>Indice:</b>	<b>2</b>
<b>1. Introduccion</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>3</b>
<b>3. Conclusiones</b>	Error! Bookmark not defined.

---

## 1. Introducción

La computación paralela es un paradigma esencial en la actualidad, ya que permite procesar grandes volúmenes de datos de manera eficiente. A medida que el crecimiento en la velocidad de los procesadores ha comenzado a estancarse, el paralelismo ha surgido como la solución para mejorar el rendimiento y reducir los tiempos de ejecución en diversas aplicaciones. Desde inteligencia artificial hasta simulaciones científicas, la capacidad de dividir una tarea en múltiples subprocesos ha revolucionado el procesamiento de datos y el diseño de software.

En este informe, se explorarán los fundamentos de la computación paralela, sus tipos y aplicaciones, así como las limitaciones y desafíos que enfrenta. Además, se presentará una comparación entre la computación secuencial y la computación paralela, analizando sus ventajas y desventajas.

---

## 2. Desarrollo

### Computación Secuencial vs. Computación Paralela

En la computación secuencial, un solo procesador ejecuta una serie de instrucciones de manera ordenada, sin posibilidad de ejecución simultánea. Este enfoque, aunque más simple, tiene un rendimiento limitado cuando se requiere procesar grandes volúmenes de datos. Por otro lado, la computación paralela permite la ejecución simultánea de múltiples instrucciones, utilizando varias unidades de procesamiento para reducir el tiempo total de ejecución.

---

## 3. Ejercicio 1 (COMPARACIÓN DE LEY DE AMDAHL Y LEY DE GUSTAFSON)

La aceleración teórica según la Ley de Amdahl se calcula con la fórmula:

$$S(N) = \frac{1}{(1 - P) + \frac{P}{N}}$$

Donde:

- **P = 0.7** (fracción paralelizable del código)

- **N = 16** (número de procesadores)

Sustituyendo los valores:

$$S(16) \approx 2.91$$

La aceleración según la Ley de Gustafson se calcula con la fórmula:

$$S(N) = N - \alpha(N - 1)$$

Donde:

-  $\alpha = 0.3$  (fracción secuencial del código)

-  $N = 16$  (número de procesadores)

Sustituyendo los valores:

$$S(16) = 11.5$$

Comparación y conclusiones

Los resultados muestran que la Ley de Amdahl es más restrictiva, limitando la aceleración máxima debido a la parte secuencial del código. En cambio, la Ley de Gustafson permite mayor escalabilidad cuando el tamaño del problema aumenta. Para aplicaciones donde el problema puede crecer, la Ley de Gustafson es más favorable.

Para mejorar la eficiencia del programa, se recomienda reducir la fracción secuencial del código mediante optimización de algoritmos y el uso de estrategias de paralelismo más avanzadas.