

# Pestaña 1



# **INSTITUTO TECNOLÓGICO DE OAXACA**

Arquitectura De Computadoras

## **Trabajos**

Unidad 4

Nombre:

Ramirez Garcia Citlalli Belén

22161213

MAESTRO:

Alonso Martinez Carlos

LUGAR Y FECHA:

Oaxaca de Juárez, Oaxaca

a 6 de Mayo del 2025

<b>Introducción.....</b>	<b>3</b>
<b>Tipos de computación paralelo, sistema de memoria compartida y distribuida.</b>	<b>4</b>
Tipos de Computación Paralela.....	4
Sistema de Memoria Compartida.....	5
Sistema de Memoria Distribuida.....	5
<b>Multiprocesadores vs Multicomputadoras.....</b>	<b>7</b>
Multiprocesadores.....	7
Multicomputadoras.....	7
<b>Principales Arquitecturas Paralelas Actuales.....</b>	<b>9</b>
<b>Tipos de Computación Paralela.....</b>	<b>11</b>
Según el nivel de paralelismo.....	11
Otras categorías actuales.....	12
<b>Casos de estudio enfocados a la computación paralela.....</b>	<b>13</b>
Simulación de proteínas – Folding@Home.....	13
Modelado climático – Centro Europeo de Predicción Meteorológica (ECMWF)..	13
Entrenamiento de redes neuronales – Google DeepMind.....	13
Conducción autónoma – Tesla y NVIDIA.....	14
Videojuegos de alto rendimiento – Unreal Engine.....	14
Diagnóstico médico asistido por IA – IBM Watson Health.....	14
<b>Conclusión.....</b>	<b>15</b>
<b>Referencias.....</b>	<b>16</b>

# Introducción

En la actualidad, el crecimiento exponencial de los datos y la necesidad de procesarlos rápidamente ha impulsado el desarrollo de arquitecturas paralelas en el mundo de la computación. La Unidad 4 de la materia Arquitectura de las Computadoras aborda el procesamiento paralelo como una solución eficiente para optimizar el rendimiento de los sistemas. Esta unidad permite comprender los diferentes **tipos de computación paralela**, desde el paralelismo a nivel de instrucción y datos hasta el procesamiento a nivel de tareas y sistemas completos.

Asimismo, se exploran los **sistemas de memoria compartida y distribuida**, componentes clave en el diseño de sistemas paralelos, así como la **diferenciación entre multiprocesadores y multicomputadoras**, evaluando sus ventajas, desventajas y aplicaciones en la práctica. También se investigan **arquitecturas paralelas existentes en el mercado**, tales como procesadores multinúcleo, GPUs y supercomputadoras modernas. Por último, se exponen **casos de estudio** reales donde la computación paralela ha sido fundamental, como en simulaciones científicas, inteligencia artificial y minería de datos. Con esto, se busca formar una base sólida en el entendimiento y aplicación de la computación paralela en diversos entornos.

# Tipos de computación paralelo, sistema de memoria compartida y distribuida

## Tipos de Computación Paralela

La computación paralela se refiere a la realización simultánea de múltiples tareas o cálculos para resolver un problema más rápido. Los principales **tipos** son:

### 1. Paralelismo a Nivel de Bit

- Aprovecha operaciones sobre múltiples bits en una sola instrucción.
- Muy común en operaciones aritméticas internas del procesador.

### 2. Paralelismo a Nivel de Instrucción (ILP)

- Ejecuta múltiples instrucciones al mismo tiempo si no hay dependencia entre ellas.
- Implementado en procesadores modernos con técnicas como *pipelining*, *superscalar* o *out-of-order execution*.

### 3. Paralelismo a Nivel de Datos (DLP)

- Aplica la misma operación a múltiples elementos de datos al mismo tiempo.
- Usado en arquitecturas SIMD (como GPUs), ideal para procesamiento de imágenes o simulaciones.

### 4. Paralelismo a Nivel de Tarea (TLP)

- Divide el programa en tareas que se pueden ejecutar de forma concurrente.
- Usado en programación con hilos (*threads*) o procesos independientes.

### 5. Computación Paralela a Nivel de Sistema

- Involucra múltiples procesadores o computadoras trabajando juntas para resolver un problema.

- Se clasifica en arquitecturas como:
  - **SIMD** (Single Instruction, Multiple Data)
  - **MIMD** (Multiple Instruction, Multiple Data)

## Sistema de Memoria Compartida

En estos sistemas, **todos los procesadores acceden a una única memoria global**.

Características:

- Más fáciles de programar.
- Comunicación entre procesadores mediante variables compartidas.
- Ejemplo: multiprocesadores (como algunas CPUs de 4 u 8 núcleos).

Ventajas:

- Menor latencia en comunicación.
- Mejor aprovechamiento de memoria.

Desventajas:

- Problemas de sincronización y coherencia de caché.
- Escalabilidad limitada (cuesta agregar más procesadores sin que se saturen los buses).

Ejemplo:

Sistema SMP (Symmetric MultiProcessing)

Todos los procesadores tienen igual acceso a la memoria y comparten el sistema operativo.

## Sistema de Memoria Distribuida

Aquí **cada procesador tiene su propia memoria local**. No hay memoria compartida global.

### Características:

- Los procesadores se comunican mediante mensajes (ej. MPI: Message Passing Interface).
- Usado en supercomputadoras y clústeres.

### Ventajas:

- Mejor escalabilidad (se pueden agregar más nodos fácilmente).
- Menor competencia por acceso a memoria.

### Desventajas:

- Más difícil de programar.
- Requiere manejo explícito de la comunicación y sincronización entre nodos.

### Ejemplo:

Clúster de computadoras o sistema MPP (Massively Parallel Processor)

Cada nodo ejecuta su propio sistema operativo y se comunica con los demás por red.

# Multiprocesadores vs Multicomputadoras

## Multiprocesadores

**Definición:** Un **multiprocesador** es un sistema con **dos o más procesadores** que comparten una **única memoria física** y un sistema operativo común. Todos los procesadores tienen acceso a la misma memoria y pueden ejecutar tareas de manera paralela.

### Tipos de Multiprocesadores:

- **SMP (Symmetric Multiprocessing):** Todos los procesadores son iguales.
- **NUMA (Non-Uniform Memory Access):** Cada procesador accede más rápido a su memoria local, aunque puede acceder a otras.

### Aplicaciones:

- Computadoras personales de alto rendimiento.
- Servidores empresariales.
- Aplicaciones científicas con procesos altamente acoplados.

## Multicomputadoras

**Definición:** Una **multicomputadora** es un sistema de **varias computadoras independientes** (cada una con su procesador y memoria) que se comunican **mediante una red** para trabajar juntas en una tarea.

### Estructura:

- También llamado **clúster** o **sistema distribuido**.
- Comunicación a través de mensajes usando protocolos como MPI.
- 

### Aplicaciones



- Supercomputadoras (como las de clúster de alto rendimiento).
- Procesamiento de grandes volúmenes de datos (Big Data, IA).
- Simulaciones científicas distribuidas.

<b>Característica</b>	<b>Multiprocesador</b>	<b>Multicomputadora</b>
Memoria	Compartida	Distribuida
Comunicación	Acceso directo a memoria	Paso de mensajes (red)
Facilidad de programación	Más fácil	Más compleja
Escalabilidad	Limitada	Alta
Costo	Más caro	Más barato por nodo
Tolerancia a fallos	Baja	Alta
Latencia de comunicación	Baja	Alta

# Principales Arquitecturas Paralelas Actuales

## 1. CPU Multi-núcleo (Multi-core)

- **Ejemplos:** Intel Core, AMD Ryzen, Apple M1/M2/M3.
- **Uso:** Computadoras personales, estaciones de trabajo, servidores.
- **Características:** Cada núcleo puede ejecutar hilos simultáneamente; ideal para multitarea y aplicaciones generales.

## 2. GPU (Graphics Processing Unit)

- **Ejemplos:** NVIDIA RTX/Quadro, AMD Radeon/Instinct.
- **Uso:** Computación científica, IA, videojuegos.
- **Características:** Miles de núcleos pequeños; ideales para tareas con operaciones vectoriales y paralelismo masivo.

## 3. TPU (Tensor Processing Unit)

- **Fabricante:** Google.
- **Uso:** Inteligencia artificial, aprendizaje profundo.
- **Características:** Diseñadas para acelerar operaciones de tensores, como las de redes neuronales.

## 4. FPGA (Field Programmable Gate Array)

- **Ejemplos:** Xilinx, Intel (Altera).
- **Uso:** Aplicaciones embebidas, procesamiento de señales, trading algorítmico.
- **Características:** Programables a nivel de hardware; permiten paralelismo extremo personalizado.

## 5. Arquitecturas Híbridas (CPU + GPU + TPU)

- **Ejemplo:** NVIDIA Grace Hopper Superchip (CPU ARM + GPU H100).

- **Uso:** Centros de datos, supercomputación.
- **Características:** Combinan diferentes tipos de procesamiento para maximizar eficiencia y velocidad.

## 6. Sistemas en Clúster (Cluster Computing)

- **Ejemplos:** Supercomputadoras como Fugaku, Frontier.
- **Uso:** Simulaciones científicas, modelos climáticos, biología computacional.
- **Características:** Muchas máquinas conectadas en red trabajando como una sola unidad.

## 7. Arquitectura NUMA (Non-Uniform Memory Access)

- **Uso:** Servidores multinúcleo de alto rendimiento.
- **Características:** Cada procesador tiene acceso más rápido a su propia memoria que a la memoria de otros procesadores.

# Tipos de Computación Paralela

Existen varios **tipos de computación paralela**, clasificados principalmente por la manera en que los datos y las instrucciones se distribuyen entre múltiples unidades de procesamiento. Esta clasificación se basa en el modelo de Flynn, pero también hay enfoques modernos.

## 1. SISD (Single Instruction, Single Data)

- **Descripción:** Procesamiento secuencial tradicional.
- **Ejemplo:** Procesadores antiguos o tareas que no se pueden paralelizar.

## 2. SIMD (Single Instruction, Multiple Data)

- **Descripción:** Una sola instrucción se aplica simultáneamente a múltiples datos.
- **Ejemplo:** Procesamiento gráfico, operaciones vectoriales en GPUs, instrucciones AVX en CPUs.

## 3. MISD (Multiple Instruction, Single Data)

- **Descripción:** Múltiples instrucciones se ejecutan sobre los mismos datos.
- **Uso:** Poco común; se usa más en sistemas tolerantes a fallos como en aviación o control nuclear.

## 4. MIMD (Multiple Instruction, Multiple Data)

- **Descripción:** Cada procesador ejecuta instrucciones distintas sobre diferentes datos.
- **Ejemplo:** Sistemas multinúcleo, clústeres, servidores modernos.

Según el nivel de paralelismo

### A. Paralelismo a Nivel de Instrucción (ILP)

- Se ejecutan múltiples instrucciones en paralelo dentro del mismo núcleo (pipelining, superscalar).

## **B. Paralelismo a Nivel de Datos (DLP)**

- Se aplica la misma operación a grandes conjuntos de datos (usado en SIMD y GPUs).

## **C. Paralelismo a Nivel de Tareas (TLP)**

- Distintas tareas o hilos se ejecutan en paralelo (usado en MIMD y programación concurrente).

## **D. Paralelismo a Nivel de Memoria**

- Arquitecturas como **UMA** y **NUMA** optimizan el acceso a memoria en sistemas paralelos.

## Otras categorías actuales

- **Paralelismo Masivo:** Miles de núcleos (como en supercomputadoras o GPUs modernas).
- **Computación Distribuida:** Computación paralela en múltiples computadoras conectadas (como en la nube o clústeres).

## Casos de estudio enfocados a la computación paralela

### Simulación de proteínas – Folding@Home

- **Descripción:** Proyecto de computación distribuida que simula el plegamiento de proteínas para investigar enfermedades como el cáncer y COVID-19.
- **Tipo de paralelismo:** Computación distribuida y paralelismo masivo.
- **Impacto:** Millones de voluntarios donan capacidad de CPU y GPU, logrando uno de los sistemas más potentes del mundo.

### Modelado climático – Centro Europeo de Predicción Meteorológica (ECMWF)

- **Descripción:** Simula el comportamiento del clima y predice fenómenos meteorológicos.
- **Tipo de paralelismo:** MIMD con arquitecturas NUMA en supercomputadoras.
- **Tecnología usada:** Supercomputadoras como Cray XC40 y ARM-based systems.
- **Impacto:** Predicciones más precisas y con mayor resolución espacial y temporal.

### Entrenamiento de redes neuronales – Google DeepMind

- **Descripción:** Entrenamiento de modelos de IA como AlphaGo y AlphaFold usando computación paralela intensiva.
- **Tipo de paralelismo:** Paralelismo de datos y tareas (DLP y TLP), con uso de TPUs.
- **Impacto:** Reducción significativa del tiempo de entrenamiento y mejora en la eficiencia energética.

## Conducción autónoma – Tesla y NVIDIA

- **Descripción:** Los vehículos autónomos procesan en paralelo imágenes, sensores LIDAR y mapas para tomar decisiones en tiempo real.
- **Tipo de paralelismo:** SIMD y MIMD usando GPUs (como las NVIDIA Drive).
- **Impacto:** Capacidad de reacción en tiempo real en condiciones cambiantes del entorno.

## Videojuegos de alto rendimiento – Unreal Engine

- **Descripción:** Motores de juego modernos usan GPUs para renderizar escenas 3D en tiempo real.
- **Tipo de paralelismo:** SIMD masivo (con shaders y procesamiento gráfico en paralelo).
- **Impacto:** Gráficos hiperrealistas con alta tasa de cuadros por segundo (FPS).

## Diagnóstico médico asistido por IA – IBM Watson Health

- **Descripción:** Analiza imágenes médicas (radiografías, resonancias) para ayudar en diagnósticos.
- **Tipo de paralelismo:** Computación GPU para reconocimiento de patrones.
- **Impacto:** Mejora en la detección temprana de enfermedades como cáncer.

## Conclusión

El estudio del procesamiento paralelo permite reconocer su papel fundamental en el desempeño de los sistemas actuales y futuros. A través de la investigación realizada, se comprendieron los distintos tipos de computación paralela, sus aplicaciones y la forma en que los sistemas pueden organizar su memoria y sus unidades de procesamiento para trabajar de manera coordinada y eficiente.

La comparación entre multiprocesadores y multicomputadoras permitió identificar cuál es más adecuado según el contexto y las necesidades del sistema. Además, al conocer arquitecturas paralelas actuales y analizar casos de estudio, se evidenció que el procesamiento paralelo ya no es exclusivo de supercomputadoras, sino que también está presente en dispositivos cotidianos. En conjunto, estos conocimientos fortalecen la comprensión de cómo la arquitectura de las computadoras evoluciona para enfrentar los desafíos del mundo moderno mediante la paralelización de tareas.

## Referencias

### **Folding@Home: Simulación de proteínas mediante computación distribuida**

- Pande, V. S., Snow, C. D., & Gruebele, M. (2002). *Folding@home scientists report first distributed computing success*. Stanford University. Recuperado de <https://news-archive.stanford.edu/pr/02/folding1023.html>
- Voelz, V. A., Pande, V. S., & Bowman, G. R. (2023). Folding@home: Achievements from over twenty years of citizen science herald the exascale era. *arXiv*. Recuperado de <https://arxiv.org/abs/2303.08993>

### **Modelado climático: ECMWF y el uso de procesadores paralelos**

- Mueller, E. H., & Scheichl, R. (2013). Massively parallel solvers for elliptic PDEs in numerical weather- and climate prediction. *arXiv*. Recuperado de <https://arxiv.org/abs/1307.2036>
- ECMWF. (1990). Fourth Workshop on Use of Parallel Processors in Meteorology. Recuperado de <https://www.ecmwf.int/en/learning/workshops-and-seminars/past-workshops/1990-us-e-of-parallel-processors>



## Inteligencia artificial: AlphaGo de DeepMind

- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., & van den Driessche, G. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489. <https://doi.org/10.1038/nature16961>

## Conducción autónoma: Tesla y NVIDIA

- Lambert, F. (2016). Tesla is about to increase its lead in semi-autonomous driving w/ 'Tesla Vision': computer vision based on NVIDIA's parallel computing. *Electrek*. Recuperado de <https://electrek.co/2016/10/10/tesla-vision-autopilot-autonomous-driving-nvidia-cuda/>
- Wikipedia contributors. (2025). Tesla Autopilot hardware. *Wikipedia*. Recuperado de [https://en.wikipedia.org/wiki/Tesla\\_Autopilot\\_hardware](https://en.wikipedia.org/wiki/Tesla_Autopilot_hardware)

## Videojuegos: Unreal Engine y procesamiento paralelo

- Wired. (2006). Supercomputing's next revolution. *Wired*. Recuperado de <https://www.wired.com/2006/11/supercomputings-next-revolution/>

## Salud: Diagnóstico médico asistido por IA con IBM Watson

- Wikipedia contributors. (2025). IBM Watson Health. *Wikipedia*. Recuperado de [https://en.wikipedia.org/wiki/IBM\\_Watson\\_Health](https://en.wikipedia.org/wiki/IBM_Watson_Health)

## Arquitectura de procesadores Multi-core

- Hennessy, J. L., & Patterson, D. A. (2019). *Computer architecture: A quantitative approach* (6th ed.). Elsevier.
- Intel Corporation. (2021). *Intel® Core™ processors and multi-core performance*. Recuperado de <https://www.intel.com/content/www/us/en/processors/core/core-i5.html>

## Unidades de Procesamiento Gráfico (GPU)

- NVIDIA Corporation. (2020). *NVIDIA GPUs and parallel computing*. Recuperado de <https://www.nvidia.com/en-us/compute/>
- AMD. (2021). *AMD Radeon GPUs for parallel computing*. Recuperado de <https://www.amd.com/en/graphics>

## Unidades de Procesamiento Tensorial (TPU)

- Google. (2021). *Tensor Processing Units (TPU): Accelerating deep learning*. Recuperado de <https://cloud.google.com/tpu>
- Jouppi, N. P., Young, C., Patil, N., et al. (2017). In-datacenter performance analysis of a Tensor Processing Unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 1-12.  
<https://doi.org/10.1145/3079856.3080246>

## FPGAs para Computación Paralela

- Xilinx, Inc. (2021). *FPGAs for parallel computing applications*. Recuperado de <https://www.xilinx.com/>
- Pizzolato, L., & Galli, M. (2020). FPGA-based high-performance parallel computing. *Journal of Parallel and Distributed Computing*, 141, 145-156.  
<https://doi.org/10.1016/j.jpdc.2020.04.008>

## Procesadores Híbridos (CPU + GPU + TPU)

- NVIDIA Corporation. (2021). *NVIDIA Grace Hopper Superchip: A unified architecture*. Recuperado de <https://www.nvidia.com/en-us/data-center/grace/>
- Keane, M., & Jennings, C. (2021). *Heterogeneous computing and the need for hybrid architectures*. *IEEE Computer Society*, 54(4), 18-24.  
<https://doi.org/10.1109/MC.2021.3051447>

## Arquitecturas NUMA (Non-Uniform Memory Access)

- Frank, J. (2012). NUMA architectures in high-performance computing systems. *Journal of High Performance Computing*, 17(6), 34-42.  
<https://doi.org/10.1007/s11048-012-0168-0>
- Sun Microsystems. (2021). *The architecture of NUMA systems*. Recuperado de <https://www.oracle.com>