

# Programación Científica y HPCI

Máster Universitario en Ingeniería Matemática y Computación

## Tema 9 / Programación Heterogénea- HPC

# Introducción a la Programación Heterogénea

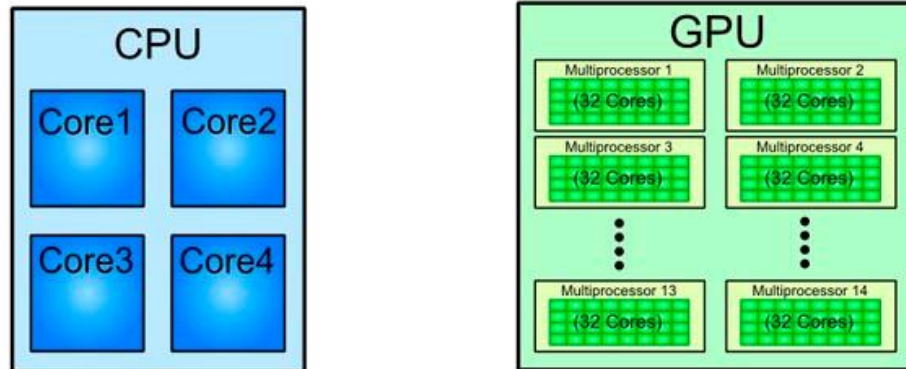
- Computación heterogénea: Uso de hardware especializado para acelerar programas.
- Aceleradores comunes: Aceleradores criptográficos, DSP y GPUs.
- GPUs: Alta relevancia, velocidad y procesamiento intensivo.



# Motivación para el Uso de GPUs

- Velocidad en el procesamiento de gráficos.
- Uso de dispositivos especializados para reducir tiempo de ejecución.
- GPU de propósito general (GPGPU) para otros cálculos intensivos.

## ARQUITECTURA CPU Y GPU



# Herramientas de Programación Heterogénea

- Adaptación del código para ejecución en GPU.
- Interfaces como OpenCL y CUDA.
- Características de Python y herramientas como Numba, PyCUDA y PyOPENCL.



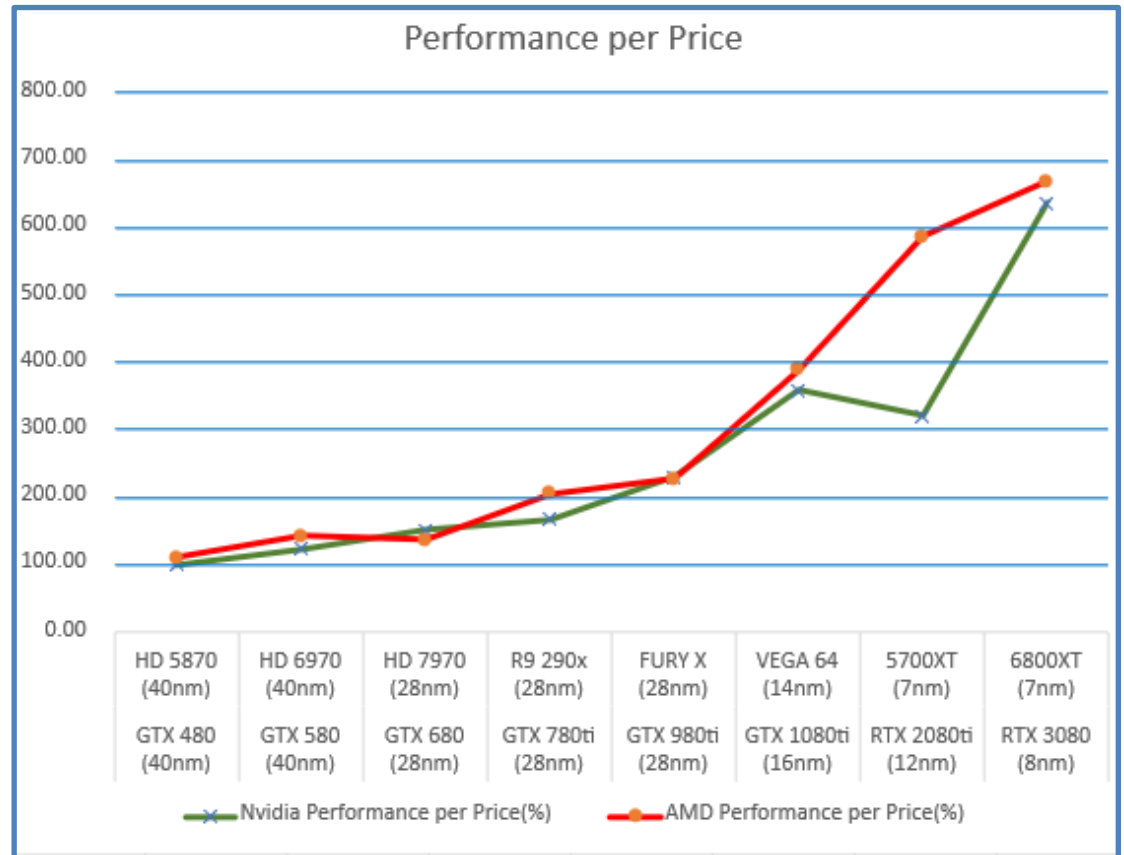
# Ventajas del Uso de GPUs

- Transparencia para el usuario y mayor velocidad de ejecución.
- Procesamiento de conjuntos de datos grandes.
- Paralelismo de grano fino (SIMD).
- Baja latencia en operaciones de coma flotante.

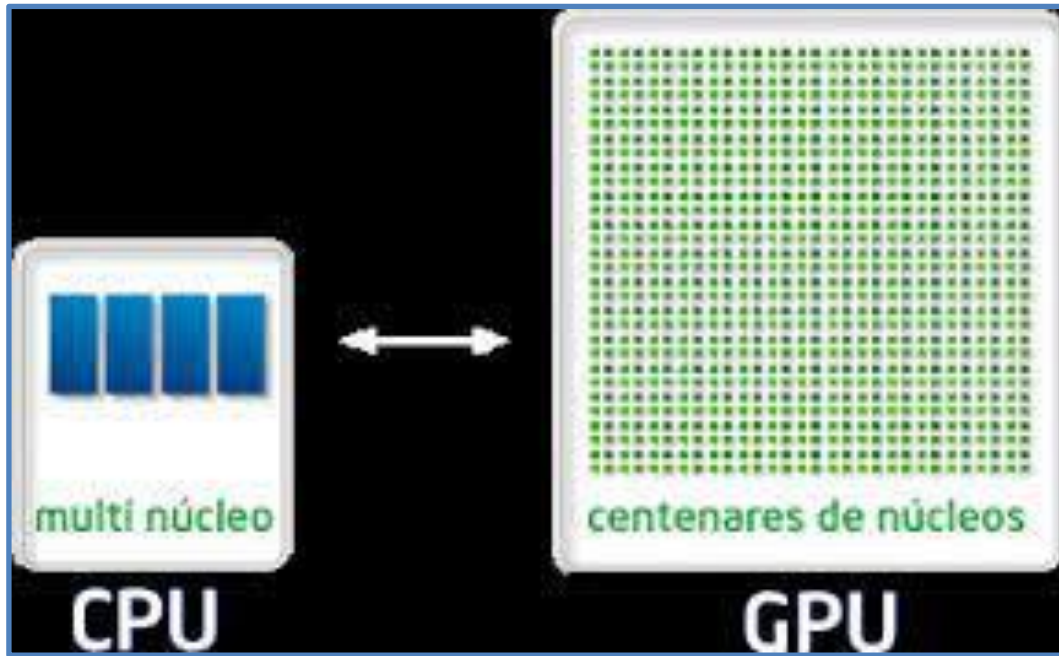


# Rendimiento de las GPUs

- GPUs con teraflops de rendimiento ( $10^{12}$  flops).
- Un flop como medida de operaciones en coma flotante.
- Aumento del rendimiento y capacidad de cálculo en GPUs.



# Factores que Afectan la Capacidad de Cómputo de una GPU



- Capacidad depende de recursos hardware.
- Número de núcleos e hilos para computación.
- Memoria local o compartida del dispositivo.



# Herramientas de Programación Heterogénea

- OpenCL y CUDA como bibliotecas más utilizadas.
- Adaptación del código y aprovechamiento de GPUs.





## Ideas clave

- Programación heterogénea y uso de GPUs aceleran aplicaciones.
- Cómputo paralelo y alto rendimiento.
- Beneficios en velocidad y eficiencia en cálculos intensivos.



# Ejemplos de Aplicaciones de Programación Heterogénea

- Ejemplo 1: Simulaciones científicas.
- Ejemplo 2: Procesamiento de imágenes y vídeos.
- Ejemplo 3: Aprendizaje automático y redes neuronales.
- Ejemplo 4: Criptografía y seguridad informática.



# Recomendaciones para la Programación Heterogénea

- Entender el problema y las necesidades de computación.
- Optimizar el código para aprovechar las características de las GPUs.
- Realizar pruebas y ajustes para obtener el mejor rendimiento.
- Mantenerse actualizado sobre las últimas tecnologías y herramientas.



# Próximos Desarrollos en Programación Heterogénea

- Avances en hardware de GPU y aceleradores especializados.
- Nuevas bibliotecas y lenguajes de programación heterogénea.
- Integración de técnicas de inteligencia artificial y aprendizaje automático.





[www.unir.net](http://www.unir.net)