



PROGRAMA DE ASIGNATURA¹

Nombre Asignatura: Computación de alto rendimiento

Código: INFO242-22

Identificación general

Docente responsable	Cristóbal A. Navarro	Docentes colaboradores	Dr. Héctor Ferrada
Correo electrónico	cristobal.navarro@uach.cl	Correo electrónico	Dr. Simón Poblete hferrada@inf.uach.cl
Horario y sala de clases	Martes V período: 15:50 hrs - 17:20 hrs Martes VI período: 17:30 hrs - 19:00 hrs		
Año y semestre	2023 I Semestre		

[illegible]

Programación por Unidades de Aprendizaje					
Unidades de Aprendizaje	Resultados de aprendizaje El estudiante es capaz de...	Estrategias de enseñanza y aprendizaje Como lo hago	Estrategias de evaluación de los aprendizajes y ponderación	Horas lectivas	Horas trabajo autónomo
Unidad 1 (5 semanas): Responsable: Cristóbal Navarro Hardware paralelo y su programación 1.1 Historia, evolución de la arquitectura Von Neumann e impacto. 1.2 Discusión de programa paralelo entregado como ejemplo. 1.3 Arquitecturas de CPU, GPU, FPGA, TPU, Cluster, y emergentes. 1.4 Concurrencia y Paralelismo, tipos de paralelismo, taxonomía de Flynn. 1.5 Herramientas para paralelizar algoritmos en CPU y GPU: OpenMP y CUDA. 1.6 Introducción al Supercomputador Patagón de la UACH. 1.7 Medidas de rendimiento: <i>work</i> , <i>span</i> , <i>speedup</i> , eficiencia paralela, <i>Strong/Weak Scaling</i> .	<ul style="list-style-type: none"> Reconocer los hitos de la historia de la Computación de alto rendimiento (sigla HPC en inglés). Identificar rendimiento acelerado y cómo afectan sus parámetros. Reconocer arquitecturas paralelas diversas. Reconocer las diferencias entre Concurrencia y Paralelismo Asociar procesadores a taxonomía de Flynn. Paralelizar un algoritmo con OpenMP y CUDA. Hacer un uso eficiente del hardware y software del supercomputador Patagón Calcular las medidas de rendimiento de forma teórica y experimental. 	Clases expositivas-activas: a través de presentación de la unidad por parte del docente responsable Clases prácticas-guiadas: actividades de programación, trabajos grupales y reuniones de discusión.	<ul style="list-style-type: none"> [15%] Tarea de programación paralela con reporte técnico. [15%] Presentación grupal de exploración. 	15	1.5 * 15

<p>Unidad 2 (6 semanas): Responsable: Cristóbal Navarro</p> <p>Modelos de cómputo y de programación</p> <p>2.1 Modelos de cómputo y de programación en memoria compartida y distribuida.</p> <p>2.2 Técnicas de paralelización para CPU, GPU y errores frecuentes.</p> <p>2.3 Técnicas no-paralelas de optimización de código fuente.</p> <p>2.4 Programación en memoria distribuida.</p>	<ul style="list-style-type: none"> • Reconocer los modelos de cómputo paralelo y programación paralela. • Manejar y aplicar las técnicas de paralelización en CPU y GPU. • Identificar los errores comunes en HPC y evitarlos al diseñar soluciones. • Aplicar optimizaciones de código no-paralelas. • Programar en un ambiente de memoria distribuida. 	<p>Clases expositivas-activas: a través de presentación de la unidad por parte del docente responsable</p> <p>Clases prácticas-guiadas: actividades de programación, trabajos grupales y reuniones de discusión.</p>	<ul style="list-style-type: none"> • [20%] Tarea de programación paralela con reporte técnico. • [20%] Tarea de optimización de código con reporte técnico. 	18	1.5 * 18
<p>Unidad 3 (6 semanas): Responsable: Cristóbal Navarro</p> <p>Casos de estudio relevantes en HPC.</p> <p>3.0 Map, reduce, filter</p> <p>3.1 Radix-sort, prefix sum, FFT.</p> <p>3.4 Aplicaciones HPC (muestreo mejorado en simulaciones moleculares)</p>	<ul style="list-style-type: none"> • Programar patrones esenciales de HPC en CPU y GPU. • Solucionar problemas clásicos de HPC usando programación paralela CPU y GPU. • Medir y comparar el rendimiento de su implementación paralela con respecto a otra de la literatura. 	<p>Clases expositivas-activas: a través de presentación de la unidad por parte del docente responsable</p> <p>Clases prácticas-guiadas: actividades de programación, trabajos grupales y reuniones de discusión.</p>	<ul style="list-style-type: none"> • [30%] Trabajo grupal con presentación. 	18	1.5 * 18

Requisitos de Aprobación

Es **requisito de aprobación** del curso obtener

- proyecto personal ≥ 4.0
- nota final ≥ 4.0

EVALUACIÓN: La nota final se determinará de la siguiente manera:

Laboratorios	40%
Proyecto Personal	30%
Trabajo con Prof. Hector	15%
Trabajo con Prof. Simon	15%

Recursos de Aprendizaje

Bibliografía

Obligatorio (profesor facilitará el material en PDF):

- Nvidia, *CUDA C Programming Guide*, 2023.
- Wen-mei Hwu et al, *Programming Massively Parallel Processors: A Hands-on Approach*, 2022.
- C.A. Navarro et al., *A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures*, 2014.
- Jaja J., *An Introduction to Parallel Algorithms*, 1992.
- Kumar et al., *Introduction to Parallel Computing*, 1994.

Complementaria:

Cormen et al., *Introduction to Algorithms*, 2010.
Stallings W., *Computer Organization and Architecture*, 2015.
OpenCL Specification, Khronos Group, 2018.
Eijkhout V., *Introduction to High Performance Scientific Computing*, 2016.