

PROGRAMA DE ASIGNATURA
“INFO 335 - Computación de alto rendimiento”

1. INFORMACIÓN GENERAL

1.1 Nombre de la asignatura	: Computación de alto rendimiento
1.2 Código	: INFO335
1.3 Créditos	: 4
1.4 Período académico	: I semestre
1.5 Tipo de asignatura	: obligatoria
1.6 Horas Teóricas	: 2
1.7 Horas Prácticas	: 2
1.8 Horas no presenciales	: 6
1.9 Cupo	: 10
1.10: Prerrequisitos	: -
1.11 Prof. Responsable	: Dr. Navarro
1.12 Prof. (es) Colaborador(es)	: Dr. Ferrada, Dr. Poblete

2. DESCRIPCIÓN DE LA ASIGNATURA

Este curso se estudia el conocimiento teórico y el manejo práctico de la computación de alto rendimiento para procesar información utilizando arquitecturas paralelas de computadores. La parte teórica se enfoca en los modelos de computación paralela y la forma de analizar la complejidad de ciertos algoritmos paralelos que son los bloques fundamentales para la construcción de soluciones más elaboradas. La parte práctica se enfoca en conocer y usar herramientas de programación paralela para implementar las técnicas en CPUs multicore, GPUs y Clusters.

3. OBJETIVOS

Objetivo general: Sintetizar bases teóricas y metodológicas de la computación de alto rendimiento para resolver problemas computacionales de forma eficiente.

Objetivos específicos:

- Analizar y diseñar algoritmos utilizando el marco teórico de computación paralela.
- Aprender y utilizar las principales herramientas de programación en paralelo.
- Resolver problemas de concurrencia en memoria compartida y distribuida.
- Extraer las principales medidas de rendimiento de una implementación en paralelo.

4. CONTENIDOS

Unidad 1: Hardware, programación paralela y medidas de rendimiento

Historia, evolución de la arquitectura Von Neumann e impacto.

[grupos] Discusión de programa paralelo entregado como ejemplo.

[seminarios] Arquitecturas de CPU, GPU, FPGA, TPU, Cluster, y emergentes.

Concurrencia y Paralelismo, tipos de paralelismo, taxonomía de Flynn.

Herramientas para paralelizar algoritmos en CPU y GPU: OpenMP y CUDA.

Introducción al Supercomputador Patagón de la UACH.

Medidas de rendimiento: *work*, *span*, *speedup*, eficiencia paralela, *Strong/Weak Scaling*.

Unidad 2: Modelos de cómputo/programación paralela y estrategias para lograr HPC.

Modelos de cómputo y de programación en memoria compartida y distribuida.

Técnicas de paralelización para CPU, GPU y errores frecuentes.

[H. Ferrada] Técnicas no-paralelas de optimización de código fuente.

Programación en memoria distribuida.

Unidad 3: Patrones y casos de estudio relevantes en HPC.

Map, reduce, filter.

Radix-sort, prefix sum, *Fast Fourier Transform* (FFT).

[S. Poblete] Aplicaciones HPC (muestreo mejorado en simulaciones moleculares)

5. METODOLOGÍA DEL TRABAJO

El curso se compone de:

1. Clases expositivas-activas por parte de los profesores.
2. Presentaciones (por parte de los estudiantes) de síntesis de tópicos de investigación basados en uno o más artículos.
3. Durante el curso, cada estudiante estudiará un problema de investigación dentro del estado del arte de la computación paralela y distribuida (idealmente vinculado a su futuro proyecto de tesis). Como resultado, elaborará un reporte técnico donde analiza el problema seleccionado, propone una solución HPC y la implementa. Dicha actividad constituirá el 50% de la nota final.

6. EVALUACIÓN

La nota final se determinará de la siguiente manera:

Lecturas, tareas y presentaciones.	: 50 %
Reporte técnico y defensa.	: 50 %

7. BIBLIOGRAFIA

1. Jaja J., *An Introduction to Parallel Algorithms*, 1992.
2. Kumar et al., *Introduction to Parallel Computing*, 1994.
3. Cormen et al., *Introduction to Algorithms*, 2010.
4. Stallings W., *Computer Organization and Architecture*, 2015.
5. Nvidia, *CUDA C Programming Guide*, 2018.
6. OpenCL Specification, Khronos Group, 2018.
7. C.A. Navarro et al., *A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures*, 2014.
8. Eijkhout V., *Introduction to High Performance Scientific Computing*, 2016.