SILABO
PERIODO ACADÉMICO 2018-01

1. INFORMACIÓN GENERAL

CURSO	: Programación Paralela y Distribuida
SEMESTRE	: I
CRÉDITOS	: 4
HORAS TEÓRICAS	: 4
HORAS PRÁCTICAS	: 0

2. PROFESOR (es):

- Marc-Antoine Le Guen (Maestría - Universidad Aix-Marseille - France)

3. JUSTIFICATIVA:

Hoy en día, la computación paralela se encuentra omnipresente a lo largo del espectro de las plataformas computacionales. Desde un nivel microscópico, los procesadores fueron usados como unidades con múltiples funcionalidades tanto de forma concurrente como pipeline por años. Ahora, los *chips multi-core* son comunes con una tendencia de incremento en el número de *chips* por *core*. Por otro lado, en un nivel microscópico, es posible construir *clusters* de cientos o miles de computadores (*multi-core*). Tales sistemas de memoria distribuida se han convertido en la principal y asequible forma de poseer un *cluster*. Por otro lado los avances en la tecnología en redes e infraestructura hicieron posible la agregación de plataformas de computación paralela a través de redes de área amplia, las cuales son conocidas como *grids*. Técnicas de virtualización han permitido consolidar la forma de servicios y su respectiva explotación en la nube.

4. OBJETIVOS:

Desarrollar un profundo conocimiento en plataformas paralelas, tanto en arquitectura, software, mecanismos de infraestructura y principios algorítmicos avanzados.
Perfeccionar la capacidad de los alumnos en el análisis de sus soluciones, aplicando modelamiento, diseño de computación paralela para la solución de sus problemas algorítmicos, teniendo en cuenta detalles como arquitecturas homogéneas y heterogéneas, complejidad y análisis en el desempeño .

5. CONTENIDOS:

- Primera Parte
  - Introducción a la Computación Paralela.
  - Comprender la necesidad de la computación paralela.
  - Comprender la arquitectura del hardware paralelo y software paralelo (SIMD, MMID, Cache, etc).
  - Programación a memoria-distribuida MPI.
  - Programación a memoria compartida con Thread C++11 así como OpenMP.
  - Algoritmos paralelos.
  - Evaluación y estudio de algoritmos.
- Segunda Parte
  - Introducción al concepto de GPGPU(General-Purpose Computing on Graphics Processing Units)..
  - Cómo realizar programación general sobre un GPU diseñado para la síntesis de imágenes entendiendo el concepto de Kernel
  - Estudiar y entender la arquitectura de un GPU, sus diferentes tipos de memorias especializadas (Texture memory, constant memory, zero-copy host memory, etc.) y como comunica con su entorno describiendo una de sus limitaciones.
  - Evaluar la posible paralelización en GPU de un algoritmo.

6. FORMA DE EVALUACIÓN:

- Trabajo 1: 30%
- Examen 1: 20%
- Trabajo 2: 35%
- Examen 2: 15%

7. BIBLIOGRAFÍA:

- Peter Pacheco. **An Introduction to Parallel Programming**, 2011.
- David Padua, Encyclopedia of Parallel Computing, 2011
- Sanders Jason, Kandrot Edward. CUDA by Example:An Introduction to General-Purpose GPU Programming, 2010
- Jason Sanders,edward Kandrot***, Cuda by example (2010)***

- John Cheng,Professional CUDA C Programming, 2014