



## Nombre de la materia

---

### ***Programa de asignatura (Grado o Posgrado)***

#### **1.- Datos generales**

1.1 Denominación de la asignatura:

*Introducción al Cálculo Numérico en Procesadores Gráficos*

1.2 Carga horaria total:

*64 hs*

1.3 Período del dictado (semestre y año)

*Abril-Mayo 2020*

#### **2.- Composición del Equipo Docente: Responsable, Auxiliares.**

2.1. Responsable a cargo

Apellido y nombres:

*Kolton, Alejandro B.*

2.3. Auxiliares

Apellido y nombres:

*Elías, Federico*

*Cappagli, Pablo*

*Curiale, Ariel*

*Carrasco, Diego*

*Mato, Germán*

#### **3.- Contenidos y Programa**

*La materia tiene como objetivo principal acercar a los estudiantes a nuevas*

*técnicas de cálculo y procesamiento de datos usando aceleradores gráficos, en el marco del paradigma de hardware y software masivamente paralelo.*

*La materia tiene un carácter fundamentalmente práctico, de laboratorio de computación, y está orientado al cálculo numérico. En cada clase, luego de una breve exposición de los conceptos, herramientas y vínculos a la información más relevante, se continúa con la y discusión y aplicación de los mismos en ejemplos simples pero concretos.*

*A lo largo de toda la materia se cada alumno debe realizar un mini-proyecto personal de investigación y desarrollo. Este consiste en la implementación completa de un código de cálculo numérico para GPGPU que resuelva un problema concreto con valor conceptual y/o práctico. Las herramientas a utilizar no se limitan a las presentadas en el curso y se incentiva la investigación del estado del arte. La temática del mini-proyecto es libre y los objetivos a perseguir se consensúan entre los alumnos y los docentes-guías.*

*Los alumnos tendrán cuentas de usuario con acceso las 24hs en el cluster de GPUs de la Gerencia de Física para experimentar con los ejemplos de clase y para realizar sus proyectos de investigación, desarrollo e implementación. Es requisito para la materia un conocimiento básico de lenguaje C y Linux, así como tener acceso a la red interna del CAB (tener una computadora personal para poder acceder remotamente al cluster durante la clase es muy conveniente, si es posible).*

#### *Programa analítico*

- Modelos de programación en paralelo. Paralelismo masivo en GPUs. Paralelismo en la arquitectura CUDA, ecosistema CUDA.*
- Herramientas para programar GPUs (lenguajes, compiladores, profilers, entornos). Uso del cluster de GPU de la GF-CAB.*
- Ejemplos de programación de GPUs usando CUDA C/C++/Fortran, pycuda y openACC.*
- Ejemplos de uso de librerías para facilitar el cálculo numérico paralelo: Patrones paralelos genéricos, algebra lineal densa y rala (incluidos algunos algoritmos para grafos), transformadas rápidas de Fourier, generadores de números aleatorios, deep learning.*
- Ejemplos de Aplicación en cálculo numérico a problemas concretos de Ciencias o Ingeniería: Machine learning, Monte Carlo, ecuaciones diferenciales parciales, dinámica molecular, dinámica estocástica, visualización en tiempo real e interactiva.*
- Optimización básica en CUDA: profiling, buenas prácticas de*

*programación en CUDA, streams, multi-GPU.*

#### 4.- Evaluación:

*La nota de evaluación se basará en ambos, el grado de participación en clase (discusión y resolución conjunta de los ejercicios dados en clase) y en la realización del mini-proyecto personal.*

*Los resultados de los proyectos serán expuestos al final de la materia en una serie de charla abiertas. El código del proyecto deberá compilar, correr y producir algunos resultados, y ser puesto a disposición de todos los alumnos y docentes para su consulta y uso.*

*Para aprobar es necesario haber concretado el proyecto (cumpliendo las condiciones estipuladas) y haber asistido al 80% de las clases.*

#### 5.- Bibliografía:

- *NVIDIA Inc., **CUDA Toolkit Reference Manual**, última versión.  
<https://docs.nvidia.com/cuda/>*
- *Sitios web: **stackoverflow, paralleforall blogs, github.***
- *David B. Kirk, Wen-mei W. Hwu. **Programming Massively Parallel Processors: A Hands-on Approach**, Morgan Kaufmann, 2010.*
- *Jason Sanders, Edward Kandrot, **CUDA by Example**, Addison-Wesley, 2010.*
- *Robert Farber, **CUDA Application Design and Development**, Morgan Kaufman, 2011*
- *Shane Cook, **CUDA Programming**, Morgan Kaufman, 2012.*
- *Duane Storti, Mete Yurtoglu, **CUDA for Engineers: An Introduction to High-Performance Parallel Computing.***
- *Gregory Ruetsch and Massimiliano Fatica, **CUDA Fortran for Scientists and Engineers.***
- *Andrzej Chrzęszczczyk, Jacob Anders, **Matrix computations on the GPU: CUBLAS, CUSOLVER and MAGMA by example**, Version 2017.*
- *Nicholas Wilt, **The CUDA Handbook: A Comprehensive Guide to GPU Programming.***
- *Tolga Soyata, **GPU Parallel Program Development Using CUDA***