



---

## Taller de Programación sobre Arquitecturas Paralelas Avanzadas

Año 2022

**Carreras:** Doctorado en Ciencias Informáticas-  
Maestría / Especialización en Computo de Altas  
Prestaciones

**Duración:** 70 hs. Totales.

**Profesor Responsable:** Dr. Enzo Rucci

Docentes: Dr. Adrián Pousa

Esp. Manuel Costanzo

**Créditos:** 4

---

### OBJETIVOS GENERALES

En la actualidad, la especialización del hardware se ha consolidado como una estrategia efectiva para continuar escalando el rendimiento y la eficiencia de los sistemas de cómputo después del final de la Ley de Moore. Comparado a las CPUs, los aceleradores hardware pueden ofrecer tasas muy superiores de rendimiento/costo y rendimiento/Watt. Esa es la principal razón por la cual los desarrolladores recalcan en una variedad de hardware, como GPUs, FPGAs, y otras clases de aceleradores (p.e. TPUs). Esto trae aparejado la existencia de modelos de programación, librerías y herramientas específicas que incrementan el costo de programación y mantenimiento. Al mismo tiempo, surgen propuestas que buscan unificar la programación de sistemas heterogéneos, con los desafíos que eso implica.

En este taller, se caracterizarán las arquitecturas hardware que integran los sistemas heterogéneos de alto rendimiento de la actualidad. A continuación, se describirán los modelos de programación, estándares y librerías asociados para el desarrollo de software en esta clase de sistemas. También se tratarán métricas de rendimiento y eficiencia energética, junto a técnicas y herramientas para su medición, evaluación y optimización. Finalmente, se presentarán casos concretos de uso de aceleradores hardware para diferentes problemas de alta demanda computacional.

### CONTENIDOS MINIMOS

- Arquitecturas CPU, GPU, MIC, FPGA, ASIC.
- Modelos, lenguajes y librerías de programación para sistemas heterogéneos. CUDA, OpenCL, SYCL, OpenMP, OpenACC, HIP.
- Métricas de performance en cada caso, incluyendo rendimiento y eficiencia energética.
- Casos de estudio experimental con cada tipo de arquitectura.



## PROGRAMA

- Introducción. Fin de la Ley de Moore. Impacto del consumo energético. Especialización del hardware como estrategia para escalar rendimiento y eficiencia de los sistemas de alto rendimiento.
- Arquitecturas CPU, GPU, MIC, FPGA, ASIC. Características, evolución y combinaciones. Fabricantes y modelos. Ventajas y desventajas de cada una. Áreas tradicionales de aplicación y usos actuales.
- Modelos, lenguajes y librerías de programación para sistemas heterogéneos. CUDA, OpenCL, OpenMP, OpenACC, HIP, SYCL. Características y evolución. Implementaciones disponibles. Propósitos, alcances y limitaciones. Técnicas de programación y optimización para c/u.
- Evaluación de prestaciones de sistemas heterogéneos. Métricas de rendimiento, eficiencia energética, portabilidad, esfuerzo de programación. Herramientas y metodologías para medición.
- Casos de estudio para cada tipo de arquitectura.

## METODOLOGIA

El dictado del curso consta de explicaciones teóricas y prácticas, autoevaluaciones para corroborar los conceptos básicos, y actividades individuales para asentar los conceptos y además adquirir experiencia práctica en la programación de sistemas heterogéneos.

En los encuentros sincrónicos, además de una explicación de los temas, se alentará la discusión de estos entre los alumnos, de forma que tengan un rol activo.

Todo el material del curso estará disponible de forma virtual, incluyendo teorías y ejercicios prácticos.

## MODALIDAD DE EVALUACION

Para aprobar el curso se deben cumplir 2 requisitos:

- asistir al 80% de las clases, resolver, entregar y aprobar las actividades correspondientes al marco teórico respetando el cronograma indicado al inicio del curso.
- realizar un trabajo final integrador que involucre el desarrollo y prueba de algoritmos paralelos utilizando alguno de los lenguajes vistos en clase y elaborar un breve informe explicando los resultados obtenidos.



## BIBLIOGRAFÍA

**Computer Architecture. A quantitative approach.**

Hennessy, Patterson. 6ta edición. Elsevier (2017).

**Introduction to High Performance Computing for Scientists and Engineers.**

Georg Hager, Gerard Wellein. CRC Press (2011).

**Parallel Programming: Concepts and practice.**

Schmidt, Bertil; Gonzalez-Domínguez, Jorge; Hundt, Christian ; Scharb, Moritz.. Morgan Kaufmann, 2018.

**Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series) - Third Edition**

David B. Kirk, Wen-mei W. Hwu. Morgan Kaufmann, 2017. ISBN-13: 978-0123814722

**CUDA Programming: A Developer's Guide to Parallel Computing with GPUs**

Shane Cook. Morgan Kaufmann, 2012. ISBN-13: 978-0124159334

**OpenCL in Action**

Matthew Scarpino, 2012. ISBN: 9781617290176

**OpenCL Programming Guide**

Aaftab Munshi, Benedict R. Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg (2021). ISBN: 978-0-321-74964-2

**Intel® Xeon Phi™ Processor High Performance Programming - Knights Landing Edition**

Jim Jeffers, James Reinders, Avinash Sodani. Morgan Kauffman, 2016.

**Data Parallel C++: Mastering DPC++ for Programming of Heterogeneous Systems using C++ and SYCL**

James Reinders, Ben Ashbaugh, James Brodman, Michael Kinsner, John Pennycook, Xinmin Tian. Apress open, 2021. ISBN-13 (electronic): 978-1-4842-5574-2

**A Survey of High Level Synthesis Languages, Tools, and Compilers for Reconfigurable High Performance Computing**

Henry SelvarajHenry SelvarajLuka DaoudLuka DaoudDawid Zydek. DOI: 10.1007/978-3-319-01857-7\_47

**Suitability Analysis of FPGAs for Heterogeneous Platforms in HPC**

Fernando A. Escobar, Xin Chang, and Carlos Valderrama. DOI: 10.1109/TPDS.2015.2407896

**Cloud Tensor Processing Unit (TPU)**

Google Inc. Disponible en <https://cloud.google.com/tpu/docs/tpus?hl=es-419>