

Formulario de aprobación de curso de posgrado/educación permanente

Asignatura: Paralelización de aplicaciones de alto rendimiento utilizando arquitecturas híbridas
(Si el nombre contiene siglas deberán ser aclaradas)

Modalidad:

(posgrado, educación permanente o ambas)

Posgrado

Educación permanente

Profesor de la asignatura ¹: Xavier Teruel y Marc Jordà, Barcelona Supercomputing Center, Barcelona, España
(título, nombre, grado o cargo, instituto o institución)

Profesor Responsable Local ¹: Sergio Nesmachnow, Prof. Titular, Gr. 5, Instituto de Computación
(título, nombre, grado, instituto)

Otros docentes de la Facultad: Santiago Iturriaga, Prof. Adjunto, Gr. 3, Instituto de Computación
(título, nombre, grado, instituto)

Docentes fuera de Facultad: --

(título, nombre, cargo, institución, país)

¹CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

[Si es curso de posgrado]

Programa(s) de posgrado: --

Instituto o unidad: Instituto de Computación

Departamento o área: Centro de Cálculo

Horas Presenciales: 40

(se deberán discriminar las horas en el ítem Metodología de enseñanza)

Nº de Créditos: 7

[Exclusivamente para curso de posgrado]

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem Metodología de enseñanza)

Público objetivo: El curso está orientado a estudiantes de posgrado y profesionales interesados en el área de computación de alto desempeño.

Cupos: El curso no tiene cupo.

(si corresponde, se indicará el número de plazas, mínimo y máximo y los criterios de selección. Asimismo, se adjuntará en nota aparte los fundamentos de los cupos propuestos. Si no existe indicación particular para el cupo máximo, el criterio general será el orden de inscripción, hasta completar el cupo asignado)

Objetivos: El objetivo del curso es introducir a los participantes en los conceptos de la computación paralela y distribuida, describiendo los principales tipos de arquitecturas de hardware existentes. Se presentarán conceptos, técnicas y herramientas de desarrollo de aplicación inmediata en la práctica, que

se ilustrarán con ejemplos y proyectos concretos. Se realizarán prácticas utilizando infraestructuras de alto desempeño de gran porte.

Conocimientos previos exigidos: Conocimientos básicos de arquitectura de sistemas, sistemas operativos y programación.

Conocimientos previos recomendados: Conocimientos básicos de redes de computadoras.

Metodología de enseñanza:

(comprende una descripción de la metodología de enseñanza y de las horas dedicadas por el estudiante a la asignatura, distribuidas en horas presenciales -de clase práctica, teórico, laboratorio, consulta, etc.- y no presenciales de trabajo personal del estudiante)

Descripción de la metodología:

Exposiciones teórico prácticas y trabajos prácticos sobre casos de estudio. Estudio y aplicación de los conceptos presentados en el curso, por parte del estudiante.

Detalle de horas:

- Horas de clase (teórico): 15
- Horas de clase (práctico): 10
- Horas de clase (laboratorio): 10
- Horas de consulta: 5
- Horas de evaluación: 0
 - Subtotal de horas presenciales: 40
- Horas de estudio: 20
- Horas de resolución de ejercicios/prácticos: 15
- Horas proyecto final/monografía: 30
 - Total de horas de dedicación del estudiante: 105

Forma de evaluación: Trabajos de aplicación durante el curso (evaluaciones orales y escritas basadas en casos de estudio). Redacción de un manuscrito aplicando los conceptos y metodologías estudiadas en el curso.

[Indique la forma de evaluación para estudiantes de posgrado, si corresponde]

[Indique la forma de evaluación para estudiantes de educación permanente, si corresponde]

Temario:

- Introducción a la interfaz de programación OpenMP.
- El modelo de paralelismo fork-join.
- El entorno de datos de OpenMP.
- Distribución de trabajos entre threads.
- Paralelismo de ciclos.
- El modelo de tareas de OpenMP
- Introducción a la programación en GPU y el modelo de programación en CUDA.
- Jerarquía de memorias en la GPU y localidad de datos.

- Estudio de eficiencia y desempeño de aplicaciones en GPU.
 - Introducción al estándar OpenACC para aceleradores abiertos y el modelo de computación heterogénea CPU/GPU.
 - Análisis (profiling) y paralelización con el toolkit OpenACC
 - Localidad de datos y optimización con OpenACC
-

Bibliografía:

(título del libro-nombre del autor-editorial-ISBN-fecha de edición)

- Using OpenMP - Portable Shared Memory Parallel Programming. Chapman B., Jost G. and Van der Pas R. MIT Press, 2008.
 - Parallel Programming with OpenACC. Farber, R. Morgan Kaufmann, 2017.
 - CUDA by Example: An Introduction to General-Purpose GPU Programming. Sanders, J and Kandrot E. Addison-Wesley, 2010.
-

Datos del curso

Fecha de inicio y finalización: 21 al 26 de octubre

Horario y Salón: a confirmar

Arancel:

[Si la modalidad no corresponde indique "no corresponde". Si el curso contempla otorgar becas, indíquelo]

Arancel para estudiantes inscriptos en la modalidad posgrado: no corresponde

Arancel para estudiantes inscriptos en la modalidad educación permanente: \$12.000
