Formulario de aprobación de curso de posgrado/educación permanente

Asignatura: Computación de Alta Performance
(Si el nombre contiene siglas deberán ser aclaradas)

Modalidad: (posgrado, educación permanente o ambas)	**Posgrado**	X
	Educación permanente	

Profesor de la asignatura [1]**:** Sergio Nesmachnow, Prof. Agregado, Gr. 5, Facultad de Ingeniería, UdelaR
(título, nombre, grado o cargo, instituto o institución)

Profesor Responsable Local [1]**:**
(título, nombre, grado, instituto)

Otros docentes de la Facultad: Nestor Rocchetti, Ayudante, Gr. 1, Facultad de Ingeniería, UdelaR
(título, nombre, grado, instituto)

Docentes fuera de Facultad:
(título, nombre, cargo, institución, país)

[1] Agregar CV 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:
Maestría y Doctorado en Informática (PEDECIBA)

Maestría y Doctorado en Ingeniería Eléctrica

Maestría y Doctorado en Mecánica de los Fluidos Computacional

Maestría y Doctorado en Ingeniería Física

Instituto o unidad: Centro de Cálculo

Departamento o área: Instituto de Computación, Facultad de Ingeniería

Horas Presenciales: 72
(se deberán discriminar las horas en el ítem Metodología de enseñanza)

Nº de Créditos: 10
[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 las áreas de computación científica y resolución eficiente de problemas complejos.

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: Introducir a los participantes en los conceptos de la computación paralela y distribuida, describir los diferentes tipos de arquitecturas de hardware existentes, pero enfatizar en arquitecturas y técnicas de programación que permitan el uso de un conjunto de computadoras interconectadas en red como si fuera una única fuente de recursos computacionales.

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

Conocimientos previos recomendados: Conocimientos básicos de redes de computadoras.

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

Detalle de horas:

- Horas de clase (teórico): 30
- Horas de clase (práctico): 10
- Horas de clase (laboratorio): 12
- Horas de consulta: 20
- Horas de evaluación: 0
    - Subtotal de horas presenciales: 72
- Horas de estudio: 10
- Horas de resolución de ejercicios/prácticos: 10
- Horas proyecto final/monografía: 60
    - Total de horas de dedicación del estudiante: 152

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:

1. Evolución histórica de la computación paralela y distribuida.
2. Categorización de Flynn, Arquitecturas SMP, MPP y clusters.
3. Modelos de computación paralela y distribuida: maestro-esclavo, cliente-servidor y arquitecturas de 3 niveles.

4. Diferencia entre procesamiento paralelo y distribuido: lenguajes concurrentes = sincronización + comunicación.
5. Método de descomposición de dominio, Método de descomposición funcional, Consideraciones de load-balancing, Medidas de performance (ley de Amdahl).
6. Primitivas de comunicación entre procesos (memoria compartida, semáforos, fork, sockets, pipes). Programación multithreading.
7. Interfases para desarrollar sistemas distribuidos: PVM (Parallel Virtual Machine), MPI (Message Passing Interface).
8. Tendencias actuales: MPP vs SMP, arquitecturas manycore, modelo MapReduce, network computing, computación grid y cloud.
9. Charlas de invitados especiales, presentación de proyectos finales de años anteriores y descripción de proyectos en el área.

Bibliografía:

(título del libro-nombre del autor-editorial-ISBN-fecha de edición)
Foster, Ian (1995). Designing and Building Parallel Programs. Ian Foster. Addison-Wesley.
Baker, Mark (1999). Cluster Computing at a glance. Software Practice and Experience 29 (6), pp. 551-576.
Kerrisk, Michael (2010). The Linux Programming Interface. No Starch Press.
Stevens, Richard (1990). UNIX Network Programming. Prentice Hall.
Geist, Al et. Al (1994). PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing. USA: Massachussetts Institute of Technology.
Gropp, William et. Al (1994). Using MPI: Portable Parallel Programming with the Message-Passing Interface. USA: Massachussetts Institute of Technology.
Gropp, William et. Al (1999). Using MPI-2: Advanced Features of the Message-Passing Interface. USA: Massachussetts Institute of Technology.
Jeffers, Jim (2013).Intel Xeon Phi Coprocessor High-Performance Programming. USA: Morgan Kaufmann.
Srinivasa, K.G. (2015). Guide to High Performance Distributed Computing: Case Studies with Hadoop, Scalding and Spark. Springer.