

**Formulario de Aprobación Curso de Actualización 2012**

**Asignatura: Computación de Alta Performance**

(Si el nombre contiene siglas deberán ser aclaradas)

**Profesor de la asignatura <sup>1</sup>:** Sergio Nesmachnow, Prof. Agregado Gr. 4  
(título, nombre, grado o cargo, Instituto o Institución)

**Profesor Responsable Local <sup>1</sup>:**  
(título, nombre, grado, Instituto)

**Otros docentes de la Facultad:** Gerardo Ares, Asistente Gr. 2, InCo  
(título, nombre, grado, Instituto)

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

**Instituto ó Unidad:** Instituto de Computación  
**Departamento ó Area:** Centro de Cálculo

**Fecha de inicio y finalización:** agosto a diciembre de 2012  
**Horario y Salón:** martes y jueves, 17 a 19 hs.

**Horas Presenciales: 70**  
(sumar horas directas de clase – teóricas, prácticas y laboratorio – horas de estudio asistido y de evaluación)  
Se deberán discriminar las mismas en el ítem Metodología de enseñanza.

**Arancel: No corresponde (\$0)**

**Público objetivo y Cupos:**

El curso está dirigido a profesionales e investigadores de cualquier área técnica, con grandes necesidades de cálculo y proceso, y que no necesariamente hayan tenido contacto anterior con el tema.

Cupo total (el curso también se dicta como curso de posgrado): 30 estudiantes.

Cupo para estudiantes de actualización: 15 estudiantes.

Justificación del cupo total: La asignatura presenta un paradigma de desarrollo de programas novedoso para los estudiantes y tiene como objetivo presentar su aplicabilidad a problemas variados utilizando la infraestructura para programación distribuida existente en Facultad (multiprocesadores de memoria compartida, redes de computadores no dedicados, clusters de computadores). La incorporación de los conceptos de la programación paralela y distribuida, el uso de herramientas de desarrollo novedosas y el trabajo sobre arquitecturas específicas, en muchos casos no utilizadas por los estudiantes en la carrera, exigen un seguimiento personal de los alumnos, que se realiza a través de trabajos prácticos a lo largo del curso. La evaluación de la asignatura involucra la realización de un proyecto práctico de carácter obligatorio, que permita aplicar los conceptos estudiados en el curso a problemas de interés académico y/o profesional de los estudiantes. La dedicación de docentes para tareas de consulta y seguimiento de los trabajos prácticos, apoyo y seguimiento del proyecto final y su posterior corrección es tal que con la disponibilidad actual de dos docentes y un ayudante, sería inadecuado tratar de atender a un número de estudiantes que supere al propuesto de 30 alumnos. De superarse este límite, la calidad de atención disminuiría notoriamente y el seguimiento personal de los estudiantes sería virtualmente imposible.

Criterio de selección de los estudiantes: En caso de que el número de estudiantes inscriptos supere el cupo propuesto para estudiantes de actualización, se realizará selección considerando el orden de inscripción de los estudiantes.

### Objetivos:

Topológicamente, un conjunto de computadoras interconectadas en una red local (LAN), puede verse como un multiprocesador con memoria distribuida, en el cual las comunicaciones entre los procesadores son lentas. El objetivo del curso es 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. Se presentarán conceptos, técnicas y herramientas de desarrollo de aplicación inmediata en la práctica, que se ilustrarán con ejemplos y proyectos concretos desarrollados en nuestra facultad. En las clases prácticas, se realizarán pruebas utilizando la infraestructura del Centro de Cálculo y de Facultad de Ingeniería (multiprocesadores de memoria compartida, redes de computadores no dedicados y clusters de computadores).

**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 las horas de clase asignadas y su distribución en horas de práctico, horas de teórico, horas de laboratorio, etc. si corresponde)

Horas clase (teórico): 30

Horas clase (práctico y laboratorio): 20

Horas consulta: 20

Horas evaluación: 60, correspondientes a desarrollo de monografía.

Horas estudio asistido: 30

Subtotal horas presenciales: 70, incluye estudio asistido.

Horas resolución ejercicios/prácticos: 0

Horas proyecto final/monografía: 60, única instancia de evaluación del curso.

Total de horas de dedicación del estudiante: 160 horas. Exposiciones teóricas semanales (total: 30 horas). Trabajo práctico y aplicaciones (total: 20 horas), Consultas (total: 20 horas). La carga horaria semanal estimada entre las actividades mencionadas es de 4,5 horas semanales (3 horas de teórico y 1,5 horas de práctico y trabajo en máquina). Desarrollo del trabajo final (total: 90 horas, incluye estudio asistido (30 horas presenciales) y evaluación (60 horas)).

### Forma de evaluación:

Realización de un proyecto al final del curso.

### Temario:

- Introducción. Evolución histórica de la computación paralela y distribuida.
- Categorización de Flynn, Arquitecturas SMP, MPP y clusters. Modelos de computación paralela: maestro-esclavo, cliente-servidor y arquitecturas de 3 niveles.
- Modelos de computación distribuida: procesamiento paralelo y distribuido,
- Lenguajes concurrentes = sincronización + comunicación
- Método de descomposición de dominio. Método de descomposición funcional
- Balance de carga. Medidas de performance. Speedup y ley de Amdahl.
- Introducción al paralelismo en lenguaje C. Primitivas de comunicación entre procesos (memoria compartida, semáforos, fork, sockets, pipes).
- Programación multithreading.
- Interfaces para desarrollar sistemas distribuidos: PVM (Parallel Virtual Machine), MPI (Message Passing Interface) y MPI-2.
- Tendencias Actuales: MPP vs SMP, influencia del lenguaje JAVA, .NET, network computing, Multi-Threading y RMI, grid y cloud computing.
- Monitoreo y administración de procesos en entornos paralelo-distribuidos.



## Facultad de Ingeniería Comisión Académica de Posgrado

- Técnicas para implementar tolerancia a fallos.
- 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)

PVM: A User's Guide and Tutorial for Networked Parallel Computing. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, y V Sunderam The MIT Press. ISBN 0-262-57108-0.

Designing and Building Parallel Programs. Ian Foster - Addison-Wesley - 1995 - ISBN 0-201-57594-9.