



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

## Programa de Computación de Alta Performance

### 1. NOMBRE DE LA UNIDAD CURRICULAR

Computación de Alta Performance

### 2. CRÉDITOS

10 créditos

### 3. OBJETIVOS DE LA UNIDAD CURRICULAR

El objetivo de la unidad curricular 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.

### 4. 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.

- Horas clase (teórico): 30
- Horas clase (práctico): 10
- Horas clase (laboratorio): 12
- Horas consulta: 20

Subtotal horas presenciales: 72

- Horas estudio: 20
- Horas resolución ejercicios/prácticos: 0
- Horas proyecto final/monografía: 60

Total de horas de dedicación del estudiante: 152

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.

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).

ocho

## 5. 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.

## 6. BIBLIOGRAFÍA

Tema	Básica	Complementaria
Evolución histórica de la computación paralela y distribuida.	(1)(2)	(3)
Categorización de Flynn, Arquitecturas SMP, MPP y clusters.	(1)(2)	(3)
Modelos de computación paralela y distribuida: maestro-esclavo, cliente-servidor y arquitecturas de 3 niveles.	(1)(2)	(6)(7)
Diferencia entre procesamiento paralelo y distribuido: lenguajes concurrentes = sincronización + comunicación.	(1)(2)	(6)(7)
Método de descomposición de dominio, Método de descomposición funcional, Consideraciones de load-balancing, Medidas de performance (ley de Amdahl).	(1)(2)	(6)(7)
Primitivas de comunicación entre procesos (memoria	(1)	(4)(5)

compartida, semáforos, fork, sockets, pipes). Programación multithreading.		
Interfases para desarrollar sistemas distribuidos: PVM (Parallel Virtual Machine), MPI (Message Passing Interface).	(1)	(6)(7)(8)
Tendencias actuales: MPP vs SMP, arquitecturas many-core, modelo MapReduce, network computing, computación grid y cloud.	(1)	(9)(10)
Charlas de invitados especiales, presentación de proyectos finales de años anteriores y descripción de proyectos en el área.	(1)	

### 6.1 Básica

1. Transparencias del curso: disponibles en la página web del curso.
2. Foster, Ian (1995). Designing and Building Parallel Programs. Addison-Wesley.

### 6.2 Complementaria

3. Baker, Mark (1999). Cluster Computing at a glance. Software Practice and Experience 29 (6), pp. 551-576.
4. Kerrisk, Michael (2010). The Linux Programming Interface. No Starch Press.
5. Stevens, Richard (1990). UNIX Network Programming. Prentice Hall.
6. Geist, Al et. Al (1994). PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing. USA: Massachusetts Institute of Technology.
7. Gropp, William et. Al (1994). Using MPI: Portable Parallel Programming with the Message-Passing Interface. USA: Massachusetts Institute of Technology.
8. Gropp, William et. Al (1999). Using MPI-2: Advanced Features of the Message-Passing Interface. USA: Massachusetts Institute of Technology.
9. Jeffers, Jim (2013). Intel Xeon Phi Coprocessor High-Performance Programming. USA: Morgan Kaufmann.
10. Srinivasa, K.G. (2015). Guide to High Performance Distributed Computing: Case Studies with Hadoop, Scalding and Spark. Springer.

*Wave*

Aprobado por resolución N°113 del CFI de fecha 04.07.2017

## **7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS**

**7.1 Conocimientos Previos Exigidos:** Conocimientos básicos de arquitectura de sistemas, sistemas operativos y programación en C.

**7.2 Conocimientos Previos Recomendados:** Conocimientos básicos de redes de computadoras.

## ANEXO A

### Para todas las Carreras

#### A1) INSTITUTO

Instituto de Computación.

#### A2) CRONOGRAMA TENTATIVO

Semana 1	Tema 1 (2 hs de clase). Tema 2 (2 hs de clase).
Semana 2	Tema 3 (1 hs de clase). Tema 4 (1 hs de clase). Tema 5 (2 hs de clase).
Semana 3	Tema 6 (4 hs de clase).
Semana 4	Clase práctica (2 hs de clase). Tema 7 (2 hs de clase).
Semana 5	Tema 7 (4 hs de clase).
Semana 6	Clase práctica (2 hs de clase). Tema 8 (2 hs de clase).
Semana 7	Clase práctica (2 hs de clase). Tema 8 (2 hs de clase).
Semana 8	Tema 8 (4 hs de clase).
Semana 9	Tema 9 (2 hs de clase). Tema 9 (2 hs de clase).
Semana 10	Clase práctica (4 hs de clase)
Semana 11	Clase laboratorio (4 hs de clase)
Semana 12	Clase laboratorio (2 hs de clase)
Semana 13	Clase laboratorio (2 hs de clase)
Semana 14	Clase laboratorio (2 hs de clase)
Semana 15	Clase laboratorio (2 hs de clase)

#### A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La evaluación involucra la realización de un trabajo práctico para familiarizarse con las técnicas y bibliotecas presentadas en el curso y un proyecto final que permita aplicar los conceptos estudiados en el curso a un problema de interés académico y/o profesional de los estudiantes. El trabajo práctico es calificado como aprobado o no aprobado y es de carácter eliminatorio. En cambio el proyecto final es calificado en una escala de 0 a 12 y refleja la nota final del curso. La problemática abordada en este proyecto forma parte de su evaluación y debe ser propuesta por los estudiantes. En ambas actividades se trabajará en grupos de dos personas.

10  
-  
dici

Aprobado por resolución N°113 del CFI de fecha 04.07.2017

**A4) CALIDAD DE LIBRE**

Los estudiantes no podrán acceder a la Calidad de Libre.

**A5) CUPOS DE LA UNIDAD CURRICULAR**

No tiene

**ANEXO B para la carrera Ingeniería en Computación (plan 97) y Licenciatura en Computación**

**B1) ÁREA DE FORMACIÓN**

Arquitectura, Sistemas Operativos y Redes de Computadoras

**B2) UNIDADES CURRICULARES PREVIAS**

Para el Curso: Cursos aprobados de:

Programación 4

y Sistemas Operativos

y alguno de los siguientes cursos:

Redes de computadoras o

Introducción a las Redes de Computadores o

Aspectos Básicos de Redes de Computadores

y alguno de los siguientes cursos:

Introducción a la Arquitectura de Computadores o

Arquitectura de Computadores 1 o

Arquitectura de Computadoras

Para el Examen: no aplica

11  
Enile

Aprobado por resolución N°113 del CFI de fecha 04.07.2017

**ANEXO B para la carrera Ingeniería en Computación (plan 87)**

**B1) ÁREA DE FORMACIÓN**

No corresponde

**B2) UNIDADES CURRICULARES PREVIAS**

Para el Curso: previas comunes a las electivas y  
examen de Arquitectura de Sistemas y Sistemas Operativos

Para el Examen: no aplica.

Observaciones: esta unidad curricular se corresponde con una electiva

APROB. POR RESOLUCIÓN DE SAC. ING.  
Fecha 19/2/19 Exp. 060120-001731-08