UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
INGENIERIA EN INFORMÁTICA
PLAN 2008
PROGRAMA DE ESTUDIO

Resolución 25/07/09-00 Acta 1215/07/04/2025
ANEXO 05

I. IDENTIFICACIÓN

1.	Asignatura	: Electiva - Diseño de Algoritmos Paralelos
2.	Semestre	: Según la alternativa en la que se inscribe el estudiante
3.	Horas semanales	: 7 horas
	3.1. Clases teóricas	: 3.5 horas
	3.2. Clases prácticas	: 3.5 horas
4.	Total real de horas disponibles	: 112 horas
	4.1. Clases teóricas	: 56 horas
	4.2. Clases prácticas	: 56 horas

II. JUSTIFICACIÓN

El interés creciente por resolver problemas más complejos a llevado a la industria de la computación a desarrollar sistemas cada vez más poderosos. El enfoque más importante para desarrollar estos sistemas ha sido la utilización de paralelismo en diferentes niveles. Sistemas paralelos de tipo clúster y multiprocesadores han sido desarrollados desde hace décadas, sin embargo su uso en general estuvo restringido a laboratorios especializados, universidades e instituciones de gran porte.
La necesidad de resolver problemas cada vez más complejos tanto en la industria como en la ciencia, el abaratamiento de los componentes, así como las limitaciones físicas de la computación con un único procesador, han hecho que estos sistemas paralelos sean utilizados en entornos cada vez más cercanos al usuario común. Desde 2007, la industria de la computación atraviesa un proceso de cambio acelerado desde procesadores con un solo núcleo a arquitecturas multinúcleo donde varios procesadores se comunican utilizando a través de sistemas cache compartidos.
Los sistemas paralelos se encuentran hoy de manera ubicua en todo tipo de dispositivo electrónico, desde sistemas de sensores, teléfonos, televisores, hasta grandes y poderosas supercomputadoras. La difusión de estos sistemas tiene y tendrá un efecto profundo en la forma en que se desarrolla el software a fin de poder hacer un uso eficiente de los recursos computacionales disponibles. Particularmente se deben desarrollar nuevos algoritmos paralelos adecuados a las diferentes plataformas disponibles, así como modificar algoritmos paralelos existentes para un mejor desempeño.
En el contexto anteriormente expuesto, tener conocimiento sobre los conceptos relacionados al diseño de algoritmos paralelos, conocer los diferentes modelos de cómputo paralelo y las herramientas más utilizadas para programar los algoritmos en dichos modelos son competencias que serán cada vez más necesarias.

III. OBJETIVOS

3.1. Proveer al estudiante los conocimientos básicos para el aprovechamiento de los computadores paralelos, multiprocesadores y multicomputadores.
3.2. Presentar los distintos modelos de programación para máquinas paralelas
3.3. Proporcionar algoritmos paralelos para problemas fundamentales, e introducir el diseño y análisis de algoritmos paralelos para la resolución de problemas concretos en ciertos campos de aplicación.

IV. PRE - REQUISITO

Para cursar esta asignatura, que se ofrece en la carrera como Electiva, el estudiante deberá cumplir con los prerrequisitos establecidos según la electiva en la que se inscribe, de acuerdo con la siguiente tabla:

Alternativa	Porcentaje de créditos aprobados	Cantidad de créditos requeridos
Electiva 1	55%	184
Electiva 2, Electiva 3, Electiva 4, Electiva 5	70%	235
Electiva 6, Electiva 7	80%	268

V. CONTENIDO

5.1. Unidades programáticas

5.1.1. Introducción al cómputo paralelo
5.1.2. Plataformas de programación paralela.
5.1.3. Principios de diseño de algoritmos paralelos
5.1.4. Operaciones de comunicación básicas
5.1.5. Modelos analíticos de programas paralelos
5.1.6. Programación utilizando el paradigma de pasos de mensajes
5.1.7. Programación de plataformas de memoria compartida

5.1.8. Algoritmos paralelos de matrices densas y algebra lineal.
5.1.9. Ordenamiento

5.2. Desarrollo de las unidades programáticas

5.2.1. Introducción al cómputo paralelo
5.2.1.1. Introducción al cómputo paralelo.
5.2.1.2. Motivación del paralelismo.

5.2.2. Plataformas de programación paralela.
5.2.2.1. Paralelismo implícito.
5.2.2.2. Limitaciones en el desempeño de los sistemas de memoria.
5.2.2.3. Costos en la comunicación en máquinas paralelas.
5.2.2.4. Organización física de plataformas paralelas.
5.2.2.5. Mecanismos de enrutamientos para redes de interconexión
5.2.2.6. Impacto del mapeo de procesos a procesadores y las técnicas de mapeo

5.2.3. Principios de diseño de algoritmos paralelos
5.2.3.1. Técnicas de descomposición
5.2.3.2. Características de tareas e interacciones
5.2.3.3. Técnicas de mapeo para el balance de carga
5.2.3.4. Métodos para contener sobrecargas de interacciones
5.2.3.5. Modelos de algoritmos paralelos

5.2.4. Operaciones de comunicación básicas
5.2.4.1. Broadcast uno a todos y reducción todos a uno
5.2.4.2. Reducción y broadcast todos a todos
5.2.4.3. Reducción a todos
5.2.4.4. Scatter y Gather
5.2.4.5. Comunicación personalizada todos a todos
5.2.4.6. Corrimiento circular
5.2.4.7. Mejora en la velocidad de algunas operaciones de comunicación

5.2.5. Modelos analíticos de programas paralelos
5.2.5.1. Fuentes de sobrecostos en programas paralelos
5.2.5.2. Métricas de desempeño de sistemas paralelos
5.2.5.3. El efecto de la granularidad en el desempeño
5.2.5.4. Escalabilidad de sistemas paralelos
5.2.5.5. Tiempo mínimo de ejecución y tiempo de ejecución mínimo de costo optimo
5.2.5.6. Análisis asintótico de programas paralelos
5.2.5.7. Métricas de escalabilidad

5.2.6. Programación utilizando el paradigma de pasos de mensajes
5.2.6.1. Principios de la programación de paso de mensajes
5.2.6.2. Las primitivas Send y Receive
5.2.6.3. Message passing interface - MPI
5.2.6.4. Solapamiento de comunicación con computación
5.2.6.5. Operaciones de comunicación y computación colectiva
5.2.6.6. Operaciones sobre grupos

5.2.7. Programación de plataformas de memoria compartida
5.2.7.1. Hilos, pthreads.
5.2.7.2. Creación y terminación de hilos
5.2.7.3. Primitivas de sincronización de Pthreads
5.2.7.4. Cancelación de un hilo
5.2.7.5. OpenMP

5.2.8. Algoritmos paralelos de matrices densas
5.2.8.1. Algoritmo de Cannon y DNS
5.2.8.2. Multiplicación matriz-vector
5.2.8.3. Multiplicación matriz-matriz
5.2.8.4. 8.4 Resolución de sistemas de ecuaciones lineales

5.2.9. Ordenamiento
5.2.9.1. Ordenamiento en computadoras paralelas
5.2.9.2. Redes de ordenamiento

VI. ESTRATEGIAS METODOLÓGICAS

6.1. La asignatura está concebida sobre las prácticas de laboratorio que los estudiantes irán desarrollando durante las clases, apoyados sobre la base teórica presentada en clases magistrales.
6.2. Las clases teóricas se desarrollan en clases magistrales y trabajos grupales, dirigidos por el docente. Además los estudiantes participarán activamente de las clases al realizar lecturas previas de un tema determinado, indicadas a través del sitio virtual de la Facultad.
6.3. Los estudiantes realizarán los trabajos de laboratorio, en grupos o individuales y serán supervisados por los docentes.
6.4. Presentación y defensa de memorias de prácticas de laboratorio y de artículos científicos relacionados con el área en cuestión.
6.5. Enseñanza apoyada en trabajo y evaluación continua, que incluyen el aprendizaje basado en problemas y el trabajo en grupo.
6.6. En la plataforma virtual de la Facultad se realizarán: foros de discusión, tareas individuales y grupales, videos tutoriales, talleres, entregas de memorias, etc..

VII. MEDIOS AUXILIARES

7.1 Pizarras acrílicas.
7.2 Marcadores.
7.3 Borrador de pizarra acrílica.
7.4 Computadoras. Sistemas operativo Linux, librerías OpenMP, MPI.

7.5 Proyectores multimedia.
7.6 Parlantes para multimedia.
7.7 Plataforma virtual "EDUCA".
7.8 Entornos de desarrollo y manuales de uso de los mismos.
7.9 Sala de laboratorio equipada para las prácticas.
7.10 Computadoras en red.
7.11 Acceso a internet.

VIII. EVALUACIÓN

La evaluación sobre el aprendizaje y conocimiento adquiridos por el alumno se realizará de acuerdo a lo establecido en el reglamento de la Facultad Politécnica de la UNA.

IX. BIBLIOGRAFÍA

- Grama, A., Gupta, A., Karypis, G. & Kumar, V (2003). *Introduction to Parallel Computing.* (2° Ed.). Addison Wesley.
- Pacheco, P. S. (2011). *An introduction to Parallel Programming.* Morgan Kaufmann Publishers.
- Casanova, H. & Legrand, A. (2008). Parallel Algorithms. Chapman and Hall/CRC Numerical Analy and Scient Comp. Series. CRC Press.
- Leopold, C. (2001). Parallel and Distributed Computing: A Survey of Models, Paradigms and Approaches, Wiley Series on Parallel and Distributed Computing. John Wiley y Son Inc.
- Roosta, S.H. (2000). Parallel Processing and Parallel Algorithms: Theory and Computation / S.H. Roosta: Springer, 2000.
- Mohcine, J., Contassot-Vivier, S. & Couturir, R. (2007). *Parallel Iterative Algorithms: From Sequential to Grid Computing, Chapman and Hall.* Boca Raton: CRC Numerical Analysis and Scientific Computation Series.

RECURSOS DISPONIBLES A TRAVÉS DE CICCO

- Petersen, W. P., & Arbenz, P. (2004). *Introduction to Parallel Computing: A Practical Guide with Examples in C.* Oxford: OUP Oxford.
- Introducing parallel programming to traditional undergraduate courses. (2012). 2012 Frontiers in Education Conference Proceedings, Frontiers in Education Conference (FIE), 2012, 1. doi:10.1109/FIE.2012.6462263
- Alan, S. (1995). Introduction to Parallel Programming, by Steven Brawer. *Scientific Programming, Vol 4, Iss 2, Pp 115-118 (1995)*, (2), 115. doi:10.1155/1995/473105