

**PROGRAMA DE ESTUDIOS POR COMPETENCIAS
PROGRAMACIÓN PARALELA Y DISTRIBUIDA**

I. IDENTIFICACIÓN DEL CURSO

ORGANISMO ACADÉMICO: Facultad de Ingeniería						
PROGRAMA EDUCATIVO: Ingeniería en Computación				ÁREA DE DOCENCIA: Programación e Ingeniería del Software		
APROBACIÓN POR LOS H.H. CONSEJOS ACADÉMICO Y DE GOBIERNO		FECHA:		PROGRAMA ELABORADO POR: Ing. Luis E. Ledezma Fuentes Ing. Élfego Gutiérrez Ocampo		PROGRAMA REVISADO POR: Integrantes de la Academia de Programación e Ingeniería de Software
				FECHA DE ELABORACIÓN : Enero 2011		FECHA DE REVISIÓN : Mayo 2011
CLAVE	HORAS DE TEORÍA	HORAS DE PRÁCTICA	TOTAL DE HORAS	CRÉDITOS	TIPO DE UNIDAD DE APRENDIZAJE	NÚCLEO DE FORMACIÓN
L41057	2	3	5	8	Curso-Laboratorio	Sustantivo
PRERREQUISITOS: Programación Avanzada, Estructuras de Datos		UNIDAD DE APRENDIZAJE ANTECEDENTE: Ninguna		UNIDAD DE APRENDIZAJE CONSECUENTE: Ninguna		
PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE: Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)						

II. PRESENTACIÓN

De manera general, la computación en paralelo comprende arquitecturas de computadoras paralelas, algoritmos paralelos y programación en paralelo.

La estructura planteada consta de seis unidades de competencia. La primera estudia las bases del cómputo paralelo en general, como necesidades de máquinas de alto rendimiento, conceptos generales, taxonomías de arquitecturas (SIMD, MIMD, SISD, MISD), las unidades funcionales del pipeline, topologías de redes de conexión y mediciones de rendimiento. La segunda se centra en el estudio de los sistemas de arquitecturas y modelos de computadoras en paralelo (SMP, MPP, COW, DSM entre otras), las estrategias de acceso a memoria. Una tercera unidad de competencia explica lo relativo a modelos y diseños de algoritmos computacionales en paralelo. La cuarta se enfoca al análisis de rendimiento y balanceo de carga computacional en algoritmos paralelos, la quinta a diseño de algoritmos haciendo uso de memoria compartida bajo el software OpenMp o similar, y la sexta a desarrollo e implementación de programas en sistemas de memoria distribuida bajo MPI.

La evaluación debe considerar tanto la parte teórica como la práctica, y se marcan como conocimientos mínimos indispensables los siguientes: arquitectura paralela de computadoras, estrategias de diseño de algoritmos paralelos, estrategia de acceso en memoria, resolución de un problema haciendo uso de programación en paralelo, entendiéndose como mínimos indispensables aquellos que son condición si ne qua non para poder aprobar la asignatura. Aún cuando el alumno superase por puntuación el límite de aprobado no se concederá el mismo si no demuestra pericia en los tópicos marcados como mínimos indispensables.

Se recomienda que el alumno practique su programación ya sea en lenguaje C ó en java previo a cursar ésta materia.

III. LINEAMIENTOS DE AL UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE :
<ul style="list-style-type: none">• Establecer las políticas del curso al inicio del mismo.• Respetar el horario del curso y la forma de evaluarlo.• Cumplir el temario y el número de horas asignadas al curso o justificar la ausencia por adelantado (asistencia a conferencias, etc.)• Asesorar y guiar el trabajo de las unidades de aprendizaje.	<ul style="list-style-type: none">• Contar con la asistencia establecida en el reglamento de Facultades:<ul style="list-style-type: none">○ 80% para examen ordinario○ 60% para examen extraordinario○ 30% para examen a título de suficiencia• Cumplir con las actividades encomendadas entregando con calidad en tiempo y forma los trabajos requeridos.

<ul style="list-style-type: none"> • Retroalimentar el trabajo de los alumnos. • Fomentar la creatividad en los alumnos a través del desarrollo de proyectos. • Evaluar y Calificar a los alumnos. • Preparar el material didáctico para las clases y prácticas. 	<ul style="list-style-type: none"> • Participar activa y críticamente en el proceso de enseñanza-aprendizaje. • Hacer uso adecuado de las instalaciones y equipo de cómputo. • Realizar las evaluaciones que se establezcan. • Mantener unas pautas de comportamiento socialmente aceptables cuando se encuentre en clases y laboratorio. • Cuando se requiera, entregar a tiempo y forma los trabajos requeridos.
--	---

IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Encuadrado en el Plan F2 por Competencias de la UAEMex, presentar al alumno con la tecnología de vanguardia en el diseño de algoritmos y programación en paralelo haciendo uso de computadoras con 2 o más procesadores o haciendo uso de procesadores de doble núcleo con vistas a capacitar al estudiante a su egreso en el análisis, diseño, desarrollo y construcción de sistemas de resolución de problemas de gran envergadura en donde se requiera el uso de arquitecturas paralelas o sistemas distribuidos.

V. COMPETENCIAS GENÉRICAS

Tal y como se establece en el apartado 4.2.1.1 Saberes del Plan Flexible 2004 por Competencias

- Analizar y diseñar algoritmos y programas haciendo uso de técnicas de programación en paralelo aplicables a la tecnología computacional.
- Analizar y diseñar proyectos donde se requiera programación en paralelo y distribuida.
- Comunicarse con expertos de otras áreas.
- Utilizar eficazmente computadoras con 2 o más procesadores.
- Analizar soluciones del entorno y problemas propios de ser tratados mediante sistemas computacionales paralelos y distribuidos.
- Proponer soluciones eficaces y eficientes.
- Crear nuevas ideas para la solución de problemas.
- Aplicar los conocimientos mediante prácticas.
- Conocer la temática básica de la profesión que desempeña en la práctica.
- Especificar arquitecturas de computadoras de arquitectura paralelas o sistemas distribuidos.

- Diseñar, desarrollar y dar mantenimiento a sistemas paralelos o distribuidos.
- Conocer la temática básica sobre programación paralela y distribuida.

VI. AMBITO DE DESEMPEÑO

- Empresas públicas y privadas
- Investigación de nuevas soluciones computacionales
- Docencia a cualquier nivel de aprendizaje escolarizado
- Desarrollo de proyectos.
- Análisis, diseño, implementación y mantenimiento de sistemas computacionales

VII. ESCENARIOS DE APRENDIZAJE

- Salón de Clases
- Sala de Cómputo

VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

1. Dar a conocer las bases del cómputo paralelo en general.
2. Dar a conocer las distintas formas de conectar procesadores con procesadores, procesadores con memoria.
3. Comprender y entender técnicas de diseño de algoritmos en paralelo y distribuidos.
4. Comprender y aplicar técnicas de balanceo de carga computacional durante el diseño de programas en paralelo y distribuidos.
5. Modelar el comportamiento de un sistema de memoria compartida en paralelo.
6. Modelar el comportamiento de un sistema de memoria distribuida en paralelo.

IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Paradigmas y bases del cómputo paralelo y distribuido	<ul style="list-style-type: none"> - Conceptos generales. -Paradigmas de la Programación en paralelo -Constitución de una computadora paralela. - Modelos de arquitectura (MIMD, SIMD, SISD, MISD). - Cómputo Paralelo y Programación en Paralelo. - Memoria Compartida y Distribuida. - Red de Conexión 	Comprender qué es el paralelismo, cómo se logra y aplicaciones.	Receptiva Analítica Responsabilidad para cumplir con las tareas asignadas Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: <ul style="list-style-type: none"> – Presentaciones acompañadas de apuntes preparados por el profesor. – Revisión y análisis de material bibliográfico – Solución de ejercicios – Desarrollo de prácticas de programación en laboratorio 	RECURSOS REQUERIDOS Pizarrón Libro de texto y apuntes del docente Laboratorio de prácticas con un lenguaje de programación modular Videoprojector	TIEMPO DESTINADO 11 horas teóricas 12 horas práctica	

CRITERIOS DE DESEMPEÑO I	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
Resolución de problemas	Ejercicios teóricos	Funcionamiento de una computadora paralela, calculo de número de operaciones
Práctica de laboratorio	Programas	Programación de problemas relacionados con series y sistemas de ecuaciones para su posterior distribución de carga computacional.

UNIDAD DE COMPETENCIA II	ELEMENTOS DE COMPETENCIA		
	Conocimientos	Habilidades	Actitudes/ Valores
Arquitecturas Paralelas y Sistemas de interconexión	<ul style="list-style-type: none"> - Sistemas de arquitectura (SMP, MPP, COW, DSM). - Modelos de acceso a memoria (UMA, NUMA, COMA, NORMA). - Ley de Amdahl. - Ley de Gustafson. 	<ul style="list-style-type: none"> • Comprender las diferentes arquitecturas paralelas • Analizar sistemas de acceso memoria avanzados para una computadora paralela. • Comprender distintas estrategias de acceso a memoria para lectura y escritura. • Medir el rendimiento de máquinas en paralelo 	Receptiva Analítica Responsabilidad para cumplir con las tareas asignadas Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un

		dependiendo de la parte secuencia de un programa.	comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: <ul style="list-style-type: none"> – Presentaciones acompañadas de apuntes preparados por el profesor. – Revisión y análisis de material bibliográfico – Solución de ejercicios – Desarrollo de prácticas de programación en laboratorio 		RECURSOS REQUERIDOS Pizarrón Libro de texto y apuntes del docente Laboratorio de prácticas con un lenguaje de programación Videoprojector	TIEMPO DESTINADO 18 horas teóricas 18 horas práctica
CRITERIOS DE DESEMPEÑO II		EVIDENCIAS	
		DESEMPEÑO	PRODUCTOS
Resolución de problemas		Ejercicios teóricos	Medición de rendimiento de computadoras paralelas y modelos de acceso a memoria.
Práctica de laboratorio		Programas	Análisis de coherencia de algoritmos, determinación del código que debe ser paralelizado.

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Técnicas de Diseño de algoritmos paralelos y distribuidos	Técnicas de algoritmos paralelos (PRAM, APRAM, C3).	<ul style="list-style-type: none"> • Conocer formas de analizar el grado de paralelización de 	Receptiva Analítica Responsabilidad para

	Particionamiento, Comunicación, Aglomeración y Mapeo (PCAM)	algoritmos.	cumplir con las tareas asignadas Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: <ul style="list-style-type: none"> - Presentaciones acompañadas de apuntes preparados por el profesor. - Revisión y análisis de material bibliográfico - Solución de ejercicios - Desarrollo de prácticas de programación en laboratorio 		RECURSOS REQUERIDOS Pizarrón, Libro de texto y apuntes del docente Laboratorio de prácticas con un lenguaje de programación modular Videoprojector	TIEMPO DESTINADO 6 horas teóricas 6 horas práctica
CRITERIOS DE DESEMPEÑO III		EVIDENCIAS	
		DESEMPEÑO	PRODUCTOS
Resolución de problemas		Desarrollo de aplicaciones utilizando las diferentes estrategias de diseño de algoritmos paralelos	Estrategias de diseño de algoritmos paralelos y su implementación.

UNIDAD DE COMPETENCIA IV	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Balanceo de carga	<ul style="list-style-type: none"> -Medición de balance de carga. -Asignación dinámica de procesos. -Balanceo de carga dinámico, robusto y no centralizado 	<ul style="list-style-type: none"> • Aprender y/o aplicar software de soporte para realización de balanceo de carga computacional 	Receptiva Analítica Responsabilidad para cumplir con las tareas asignadas Tolerancia y participación Desarrollar la capacidad analítica ante nuevos problemas Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable
ESTRATEGIAS DIDÁCTICAS: <ul style="list-style-type: none"> – Presentaciones acompañadas de apuntes preparados por el profesor. – Revisión y análisis de material bibliográfico – Solución de ejercicios – Desarrollo de prácticas de programación en laboratorio 	RECURSOS REQUERIDOS Pizarrón, Libro de texto y apuntes del docente Laboratorio de prácticas con un lenguaje de programación modular Videoprojector	TIEMPO DESTINADO 6 horas teóricas 6 horas práctica	
CRITERIOS DE DESEMPEÑO IV	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
Resolución de problemas	Desarrollo de programas haciendo uso de técnicas de balanceo de carga.	Distribución de carga computacional hacia diferentes procesos	

UNIDAD DE COMPETENCIA V	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Sistema de memoria compartida	<p>Explorar la intuición usada en crear un algoritmo paralelo diseñándolo y realizándolo en un hardware con memoria compartida.</p> <p>Uso de Java y OpenMP</p>	<ul style="list-style-type: none"> Evaluación operacional en un sistema de memoria compartida 	<p>Receptiva</p> <p>Analítica</p> <p>Responsabilidad para cumplir con las tareas asignadas</p> <p>Tolerancia y participación</p> <p>Desarrollar la capacidad analítica ante nuevos problemas</p> <p>Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable</p>
<p>ESTRATEGIAS DIDÁCTICAS:</p> <ul style="list-style-type: none"> Presentaciones acompañadas de apuntes preparados por el profesor. Revisión y análisis de material bibliográfico Solución de ejercicios Desarrollo de prácticas de programación en laboratorio 	<p>RECURSOS REQUERIDOS</p> <p>Pizarrón,</p> <p>Libro de texto y apuntes del docente</p> <p>Laboratorio de prácticas con un lenguaje de programación modular</p> <p>Videoprojector</p>	<p>TIEMPO DESTINADO</p> <p>2 horas teóricas</p> <p>10 horas práctica</p>	
CRITERIOS DE DESEMPEÑO V	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
	Mostrar el motivo por el cual	Herramienta de	

Resolución de problemas	ciertos problemas son resueltos mediante algoritmos paralelos para memoria compartida	programación de memoria compartida OpenMp o similar
-------------------------	---	---

UNIDAD DE COMPETENCIA VI	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Sistema de memoria distribuida	<p>Explorar la intuición usada en crear un algoritmo paralelo diseñándolo con memoria distribuida.</p> <p>Instrucciones de paralelización (mpi, pvm, java) aplicados a problemas numéricos.</p>	<ul style="list-style-type: none"> Evaluación operacional en un sistema de memoria distribuida 	<p>Receptiva</p> <p>Analítica</p> <p>Responsabilidad para cumplir con las tareas asignadas</p> <p>Tolerancia y participación</p> <p>Desarrollar la capacidad analítica ante nuevos problemas</p> <p>Respetar al docente y a los compañeros mediante un comportamiento socialmente aceptable</p>
<p>ESTRATEGIAS DIDÁCTICAS:</p> <ul style="list-style-type: none"> Presentaciones acompañadas de apuntes preparados por el profesor. Revisión y análisis de material bibliográfico Solución de ejercicios Desarrollo de prácticas de programación en laboratorio 		<p>RECURSOS REQUERIDOS</p> <p>Pizarrón,</p> <p>Libro de texto y apuntes del docente</p> <p>Laboratorio de prácticas con un lenguaje de programación modular</p> <p>Videoprojector</p>	<p>TIEMPO DESTINADO</p> <p>2 horas teóricas</p> <p>10 horas práctica</p>

CRITERIOS DE DESEMPEÑO V	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
	Resolución de problemas	Mostrar el motivo por el cual ciertos problemas son resueltos mediante algoritmos paralelos para memoria distribuida

X. EVALUACIÓN Y ACREDITACIÓN

Para tener derecho a examen ordinario requiere al menos un 80% de asistencia.

Para tener derecho a presentar el examen extraordinario se requiere al menos un 60 % de asistencia.

Para tener derecho a presentar el examen a título de suficiencia se requiere al menos un 30% de asistencia.

Se realizarán dos o más exámenes parciales con valor del 40%

Se realizarán uno, dos o tres Proyectos Parciales con valor del 60%

Si el estudiante finaliza el proyecto de manera satisfactoria previo a la fecha de examen ordinario y la calificación es mayor o igual a 80% el alumno exenta.

La calificación final ordinaria estará compuesta por 40 % de examen escrito final y un 60 % de un proyecto acumulativo práctico

La calificación final extraordinaria estará compuesta por 30 % del examen escrito final correspondiente y un 70 % de un proyecto final práctico

La calificación final a título de suficiencia estará compuesta por 30 % del examen escrito final correspondiente y un 70 % de un proyecto final práctico.

XII. REFERENCIAS

BIBLIOGRAFÍA BÁSICA:

- ✓ Francisco Almeida, Domingo Giménez, José Miguel Mantas, Antonio M. Vidal. (2008) "Introducción a la programación paralela". Paraninfo Cengage Learning
- ✓ Jordan Harry, Alaghband Gita. (2004) "Fundamentals of Parallel Processing". Prentice Hall
- ✓ Camera Hughes, Tracey Hughes. (2004) "Parallel and Distributed Programming Using C++". Addison Wesley.
- ✓ Ananth Grama, Anshul Gupta, George Karypis & Vipin Kumas. (2003) "Introduction to Parallel Computing". Pearson Addison Wesley.

BIBLIOGRAFÍA COMPLEMENTARIA:

- ✓ Andrew S. Tanenbaum, Maarten Van Steen. (2003) "Distributed Systems : Principles and Paradigms". Prentice Hall.
- ✓ M. L. Liu. (2004) "Computación Distribuida : Fundamentos y Aplicaciones". Pearson Addison Wesley.
- ✓ OpenMPI. <http://www.open-mpi.org/>.
- ✓ OpenMP. <http://www.openmp.org/blog/>