

CARTA DESCRIPTIVA (FORMATO MODELO EDUCATIVO UACJ VISIÓN 2020)

I. Identificadores de la asignatura			
Instituto	IIT	Modalidad:	Presencial
Departamento:	Ingeniería Eléctrica y Computación	Créditos:	
Materia:	Cómputo Paralelo	Carácter:	Electiva
Programa:	Maestría en Cómputo Aplicado	Tipo:	Curso
Clave:	-----		
Nivel:	Maestría		
Horas:	48 Hrs. Totales	Teoría: 40%	Práctica: 60%

II. Ubicación	Clave:
Antecedentes: No Aplica.	
Consecuente: No Aplica.	

III. Antecedentes
Conocimientos: Lenguajes de programación, sistemas operativos y redes de computadoras.
Habilidades: Resolución de problemas en entornos nuevos o pocos conocidos en un contexto de bases de datos. Argumentación mediante lenguaje oral y escrito. Trabajo en equipo.
Actitudes y valores: Honestidad académica, puntualidad, juicio constructivo, responsabilidad, respeto, auto-aprendizaje, trabajo en equipo y personalidad emprendedora.



IV. Propósitos Generales El cómputo paralelo emerge como una alternativa para solucionar problemas complejos cuya solución algorítmica requiere de una cantidad considerable de recursos computacionales (CPU, Memoria y Almacenamiento). Esta materia presenta un enfoque teórico-práctico del cómputo paralelo. Al finalizar el curso el alumno será capaz de aplicar el cómputo paralelo en la solución de problemas.
V. Compromisos formativos
Intelectual: Describe los principios básicos del cómputo paralelo. Introduce a la complejidad computacional. Explica el ciclo de vida de una aplicación paralela. Describe el problema de asignación de tareas a procesadores. Utiliza herramientas de simulación para asignar tareas a procesadores. Utiliza herramientas para desarrollar aplicaciones paralelas. Desarrollar un caso de estudio. Realizar trabajos de investigación, ensayos y proyectos.
Humano: Respeto al trabajo intelectual de las personas, honestidad académica y respeto a sí mismo y a los demás.
Social: Compromiso social y uso ético del conocimiento
Profesional: El alumno incorpora a su formación conocimientos que le permitan comprender, analizar y dominar el ciclo de vida de una aplicación paralela.

VI. Condiciones de operación			
Espacio:	Aula de clase		
Laboratorio:	Cómputo	Mobiliario:	mesas y sillas

Población:	25 – 30
Material de uso frecuente:	A) Pizarrón y marcadores B) Proyector C) Computadora portátil
Condiciones especiales:	No aplica

Temas	Contenidos	Actividades
1. Introducción al curso. (3 horas)	<ol style="list-style-type: none"> 1. Encuadre del curso. 2. Introducción. 3. Algoritmos Secuenciales. 4. Algoritmos Paralelos. 5. Importancia del cómputo paralelo. 6. Aplicaciones actuales. 	<p>El profesor presenta el plan de estudios, las políticas del curso y la forma de evaluar.</p> <p>El profesor introduce el cómputo paralelo.</p> <p>El profesor explica los algoritmos secuenciales y su diferencia con los algoritmos paralelos.</p> <p>El profesor explica la importancia del cómputo paralelo.</p> <p>El profesor muestra aplicaciones actuales que requieren del uso del cómputo paralelo.</p>
2. Principios básicos de cómputo paralelo. (3 horas)	<ol style="list-style-type: none"> 1. Características de los algoritmos paralelos. 2. Plataformas computacionales paralelas. 3. Ley de Amdahl 4. Medidas de desempeño de arquitecturas paralelas. 5. Aceleración, Eficiencia y Redundancia. 	<p>El profesor explica las características de los algoritmos paralelos.</p> <p>El profesor explica las plataformas computacionales paralelas.</p> <p>El profesor explica la ley de Amdahl.</p> <p>El profesor describe medidas de desempeño de arquitecturas paralelas.</p> <p>El profesor explica los conceptos de aceleración, eficiencia y redundancia.</p>
6. Complejidad Computacional. (6 horas)	<ol style="list-style-type: none"> 1. Introducción Tiempos de Ejecución 2. Concepto de Complejidad 3. Órdenes de Complejidad Notación Asintótica <ul style="list-style-type: none"> * La O Mayúscula * La o Minúscula * Diferencias entre O y o 4. Problemas NP Completos 5. Problemas Intratables Problemas de Decisión 6. Algoritmos No Determinísticos 	<p>El profesor introduce el análisis de algoritmos a los estudiantes.</p> <p>El profesor explica la estimación del tiempo de ejecución de un algoritmo.</p> <p>El profesor define el concepto de complejidad computacional.</p> <p>El profesor explica las notaciones O y o en el contexto del análisis de algoritmos.</p> <p>El profesor describe la complejidad de los problemas computacionales.</p>
4. El ciclo de vida de una aplicación paralela. (3 horas)	<ol style="list-style-type: none"> 1. Problemas complejos actuales. 2. Partición de Problemas Complejos. <ul style="list-style-type: none"> - En forma de grafos. - Tareas independientes. 3. Definición de las relaciones entre tareas. 4. Asignación de tareas a procesadores. 5. Ejecución de la aplicación. 	<p>El profesor describe el ciclo de vida de una aplicación paralela.</p> <p>I El profesor muestra ejemplos de problemas complejos actuales.</p> <p>II El profesor describe el proceso de la partición de problemas.</p> <p>III El profesor explica la definición de relaciones entre tareas en una aplicación con forma de grafo.</p> <p>IV El profesor explica el problema de asignar tareas a procesadores.</p> <p>V El profesor explica la ejecución de la aplicación.</p>

<p>4. El problema de asignar tareas a procesadores. (6 horas)</p>	<ol style="list-style-type: none"> 1. Introducción. 2. Tipos de asignación <ul style="list-style-type: none"> - Estática - Dinámica 3. Estrategias de asignación. <ul style="list-style-type: none"> - Basado en listas - Basado en redundancia de tareas. - Agrupamiento de tareas. - Reactivos - Tolerantes a fallas 4. Medidas de desempeño de algoritmos de asignación. 	<p>El profesor introduce el tema. El profesor explica los tipos de asignación. El profesor describe las estrategias de asignación de tareas a procesadores: basado en listas, basado en redundancia de tareas, agrupamiento, reactivos y tolerantes a fallas. El profesor se apoya en la lectura de artículos de la literatura para comprender el tema. El profesor explica las medidas de desempeño de algoritmos de asignación.</p>
<p>5. Simulación de algoritmos de asignación de tareas. (9 horas)</p>	<ol style="list-style-type: none"> 1. Introducción. 2. Uso de Herramientas de software. 3. Prácticas. 	<p>El profesor en base a su experiencia guía al alumno en el uso de una herramienta de simulación de algoritmos de asignación de tareas a procesadores. Se sugiere utilizar una de las siguientes herramientas de la literatura: Pulpo, Simgrid, Gridsim.</p>
<p>6. Herramienta para programar una aplicación paralela. (12 horas)</p>	<ol style="list-style-type: none"> 1. Introducción. 2. Uso de Herramientas de software. 3. Prácticas. 	<p>El profesor en base a su experiencia guía al alumno en el uso de una herramienta de software para desarrollar aplicaciones paralelas. Se sugiere utilizar una de las siguientes herramientas de la literatura: MPI, PVM, OpenMP, Cuda) Se recomienda realizar prácticas con ordenamientos y búsquedas con una gran cantidad de elementos. También realizar mediciones para evaluar el cómputo secuencial vs cómputo paralelo.</p>
<p>7. Caso de estudio (6 horas)</p>	<ol style="list-style-type: none"> 1. Introducción 2. Solución del caso de estudio. 	<p>El profesor en base a su experiencia guía al alumno en la solución de un caso de estudio. Se recomienda proponer problemas que utilicen métodos numéricos.</p>

<p>VIII. Metodología y estrategias didácticas</p>
<p>Metodología Institucional:</p> <ol style="list-style-type: none"> a) Elaboración de ensayos, monografías e investigaciones (según el nivel) consultando fuentes bibliográficas, hemerográficas y en Internet. b) Elaboración de reportes de lectura de artículos en lengua inglesa, actuales y relevantes. <p>Estrategias del Modelo UACJ Visión 2020 recomendadas para el curso:</p> <ol style="list-style-type: none"> a) aproximación empírica a la realidad

- b) búsqueda, organización y recuperación de información
- c) comunicación horizontal
- d) descubrimiento
- e) ejecución-ejercitación
- f) elección, decisión
- g) evaluación
- h) experimentación
- i) extrapolación y transferencia
- j) internalización
- k) investigación
- l) meta cognitivas
- m) planeación, previsión y anticipación
- n) problematización
- o) proceso de pensamiento lógico y crítico
- p) procesos de pensamiento creativo divergente y lateral
- q) procesamiento, apropiación-construcción
- r) significación generalización
- s) trabajo colaborativo

IX. Criterios de evaluación y acreditación

a) Institucionales de acreditación:

Acreditación mínima de 80% de clases programadas

Entrega oportuna de trabajos

Pago de derechos

Calificación ordinaria mínima de 7.0

Permite examen único: no

b) Evaluación del curso

Es de acuerdo al criterio del profesor pero se sugiere:

Exámenes Parciales (20%)

Ensayos y trabajos en equipo (10%)

Tareas individuales y prácticas de laboratorio (40%)

Caso de estudio (30%)

Total: 100 %

X. Bibliografía

Grama Ananth and Karypis George, “*Introduction to Parallel Computing*”, Pearson, second edition, 2003.

Pacheco Peter, “*An Introduction to Parallel Programming*”, Morgan Kaufman, first edition, 2011.

Storti Duane and Yurtoglu Mete, “*CUDA for Engineers: An Introduction to High-Performance Parallel Computing*”, Addison-Wesley Professional, first edition, 2015.

Gropp William, Ewing Lusk and Skjellum Anthony, “*Using MPI: Portable Parallel Programming with the Message-Passing Interface*”, MIT Press, third edition, 2014.

Nielsen Frank, “*Introduction to HPC with MPI and Data Science*”, Springer, first edition, 2016.

X. Perfil deseable del docente

Preferentemente doctorado en áreas de ciencias o ingeniería de la computación o tecnologías de información.

XI. Institucionalización

Responsable del Departamento: Jesús Armando Gándara Fernández.

Coordinador del Programa: Victoria González de Moss.

Fecha de elaboración: Mayo 11, 2018.

Elaboró: Israel Hernández.