

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad académica: Instituto de Investigación en Ciencias Básicas y Aplicadas							
Plan de estudios: Licenciatura en Inteligencia Artificial							
Unidad de aprendizaje: SISTEMAS PARALELOS				Ciclo de formación: Profesional Eje general de formación: Teórico-Técnica Área de conocimiento: Fundamentos de la computación aplicada Semestre: 4º			
Elaborada por: Dra. Lorena Díaz González M.I.C.A. Yainier Labrada Nueva				Fecha de elaboración: Abril, 2021			
Clave:	Horas teóricas :	Horas prácticas :	Horas totales :	Créditos :	Tipo de unidad de aprendizaje :	Carácter de la unidad de aprendizaje :	Modalidad:
SP29CP03020 8	03	02	05	08	Obligatoria	Teórico - Práctica	Escolarizada
Plan (es) de estudio en los que se imparte: A partir de todos los programas impartidos por el Instituto de Investigación en Ciencias Básicas y Aplicadas.							

ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

Presentación: Durante la unidad de aprendizaje se estudian los principios y fundamentos de la computación en paralelo, tales como métricas de desempeño, arquitecturas paralelas, clasificación de algoritmos paralelos, diseño de algoritmos paralelos mediante diversas técnicas paralelas basadas en estructuras de datos avanzadas y lenguajes de programación.



Propósito:

Conozca, identifique y aplique los conocimientos sobre los principios fundamentales de los sistemas paralelos, mediante un conjunto de prácticas sobre entornos paralelos y tarjetas NVIDIA que soporten la programación sobre CUDA, para desarrollar sistemas de software con ética y de forma creativa.

Competencias que contribuyen al perfil de egreso

Competencias genéricas:

- CG10. Habilidades en el uso de la tecnología de la información y de la comunicación.
- CG16. Capacidad para identificar, plantear y resolver problemas.
- CG23. Capacidad de trabajo en equipo.

Competencias específicas:

- CE7. Aplica diferentes métodos computacionales mediante el uso de herramientas modernas para la resolución de problemas reales de la ciencia y la industria de una manera colaborativa
- CE8. Implementa correctamente los algoritmos considerando aspectos prácticos, beneficios y ventajas de los distintos lenguajes de programación para el desarrollo de aplicaciones, comprometiéndose con la calidad.

CONTENIDOS

Bloques	Temas
1. Introducción a la Computación de Alto Desempeño	1.1. Sistemas Paralelos <ul style="list-style-type: none"> 1.1.1. Hardware de un Sistema Paralelo 1.1.2. Software de un Sistema Paralelo 1.2. Modelos Paralelos Estándares: Paradigmas <ul style="list-style-type: none"> 1.2.1. Paralelismo de Datos 1.2.2. Paralelismo de Tareas 1.2.3. Paralelismo Anidado de Datos 1.3. Paradigmas orientados a la Arquitectura <ul style="list-style-type: none"> 1.3.1. Paradigma de Memoria Compartida



	<p>1.3.2. Paradigma de Pasaje de Mensaje</p> <p>1.4. Resumen</p> <p>1.5 Ejercicios</p>
2. Introducción a GPU	<p>2.1. CPU y GPU</p> <p>2.2. Evolución Histórica de la GPU</p> <p>2.3. Inicio de la GPU</p> <p> 2.3.1. Estructura del Pipeline Gráfico</p> <p> 2.3.2. Pipeline en CPU y en GPU</p> <p> 2.3.3. GPU: Arquitectura Fijas vs. Unificadas</p> <p>2.4. Arquitecturas Unificadas de GPU</p> <p> 2.4.1. Arquitectura G80</p> <p> 2.4.2. Arquitectura GT200</p> <p> 2.4.3. Arquitectura GF100</p> <p>2.5. GPGPU: Computación de Propósito General en GPU</p> <p>2.6. Resumen</p> <p>2.7. Ejercicios</p>
3. Programación de GPU: Modelo de Programación CUDA	<p>3.1. Introducción a CUDA</p> <p>3.2. Arquitectura y Modelo de Programación CUDA</p> <p> 3.2.1. Arquitectura de GPU según CUDA</p> <p> 3.2.2. Modelo de Programación CUDA</p> <p>3.3. Generalidades de la Programación con CUDA</p> <p>3.4. Estructura de un Programa en CUDA</p> <p> 3.4.1. Transferencia de datos CPU-GPU</p> <p> 3.4.2. Función kernel y Threads</p> <p> 3.4.3. Multi-Bloques y Multi-Threads</p> <p> 3.4.4. Sincronización de Threads</p> <p>3.5. Modelo de Ejecución</p>



	<p>3.5.1. Administración de Threads</p> <p>3.6. Otro ejemplo: Multiplicación de Matrices</p> <p>3.7. Resumen</p> <p>3.8. Ejercicios</p>
4. Jerarquía de Memoria	<p>4.1. Modelo de Memoria de GPU</p> <p>4.2. Memoria Global</p> <p>4.3. Memoria Compartida</p> <p>4.4. Memoria de Registros</p> <p>4.5. Memoria Local</p> <p>4.6. Memoria de Constante</p> <p>4.7. Memoria de Texturas</p> <p>4.8. La Memoria como Límite del Paralelismo</p> <p>4.9. Ejemplos del uso de la Jerarquía de Memorias</p> <p style="padding-left: 40px;">4.9.1. Producto Punto 112</p> <p style="padding-left: 40px;">4.9.2. Multiplicación de Matrices usando Memoria Compartida</p> <p>4.10. Resumen</p> <p>4.11. Ejercicios</p>
5. Análisis de Rendimiento y Optimizaciones	<p>5.1. Ejecución de Threads</p> <p>5.2. Memoria Global</p> <p style="padding-left: 40px;">5.2.1. Organización de Accesos</p> <p style="padding-left: 40px;">5.2.2. Prefetching de Datos</p> <p>5.3. Rendimiento de las Instrucciones</p> <p style="padding-left: 40px;">5.3.1. Mezcla de Instrucciones</p> <p style="padding-left: 40px;">5.3.2. Granularidad</p> <p>5.4. Asignación de los recursos de un SM</p> <p>5.5. Resumen</p> <p>5.6. Ejercicios</p>



ESTRATEGIAS DE ENSEÑANZA - APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque X)			
Aprendizaje basado en problemas	(X)	Nemotecnia	()
Estudios de caso	()	Análisis de textos	()
Trabajo colaborativo	(X)	Seminarios	()
Plenaria	()	Debate	(X)
Ensayo	()	Taller	()
Mapas conceptuales	()	Ponencia científica	()
Diseño de proyectos	(X)	Elaboración de síntesis	()
Mapa mental	()	Monografía	()
Práctica reflexiva	()	Reporte de lectura	()
Trípticos	()	Exposición oral	(X)
Otros			
Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	(X)	Experimentación (prácticas)	()
Debate o Panel	()	Trabajos de investigación documental	()
Lectura comentada	(X)	Anteproyectos de investigación	()
Seminario de investigación	()	Discusión guiada	()
Estudio de Casos	(X)	Organizadores gráficos (Diagramas, etc.)	()
Foro	()	Actividad focal	()
Demostraciones	()	Analogías	()
Ejercicios prácticos (series de problemas)	(X)	Método de proyectos	()



Interacción la realidad (a través de videos, fotografías, dibujos y software especialmente diseñado).	()	Actividades generadoras de información previa	()
Organizadores previos	()	Exploración de la web	()
Archivo	()	Portafolio de evidencias	(X)
Ambiente virtual (foros, chat, correos, ligas a otros sitios web, otros)	()	Enunciado de objetivo o intenciones	()

CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
• Exámenes	30%
• Proyecto final	40%
• Resolución de problemas y estudios de casos.	20%
• Reporte de lecturas	10%
Total	100 %

PERFIL DEL PROFESORADO

Licenciatura, Maestría o Doctorado en ciencias computacionales o ingeniería en áreas afines a las ciencias computacionales, con experiencia docente en el área.

REFERENCIAS

Básicas:

- White O'Reilly, Tom. (2009). *Hadoop: the definitive guide*. Ed. Media.
- Pacheco, Peter S. (2011). *An introduction to parallel programming*. Ed. Morgan Kaufmann Publishers.
- Gebali, Fayez. *Algorithms and parallel computing*. Ed. John Wiley & Sons.
- Quinn, Michael J. (2003). *Parallel computing, theory and practice*. 2a edición. Ed. McGraw-Hill.
- J. Sanders and E. Kandrot. (2010). "CUDA by Example, An introduction to general-purpose GPU programming," pp. 2-11.
- David B. Kirk, Wen-mei W. Hwu. (2012). *Programming Massively Parallel Processors: A Hands-on Approach*, 2nd ed., Morgan Kaufmann.
- J. K. Keller. (2001). *Practical PRAM programming*. John Wiley & Sons inc.
- B. R. Kernighan. (1978). *The C programming language*. Prentice-Hall.
- E. N. Lindholm. (2008). "NVIDIA TESLA: A UNIFIED GRAPHICS AND COMPUTING ARCHITECTURE". In IEEE Computer Society, 39-45.



- S. J. Pennycook. (2011). "Performance analysis of a hybrid MPI/CUDA implementation of the NASLU benchmark".
- InSIGMETRICS Perform. Eval. Rev., 38(4), 23-29.
- M. Quinn. (1994). *Parallel Computing. Theory and Practice*. Second edition. McGraw-Hill. Inc.

Complementarias:

- NVidia. (2021). NVidia CUDA Programming Guide. Available: http://developer.download.Nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_Programming_Guide_2.3.pdf

