

Universidad Nacional de San Agustín  
VICE RECTORADO ACADÉMICO  
SILABO

CODIGO DEL CURSO: CS314

<b>1 Datos Generales</b>	<b>FACULTAD :</b> Ingeniería de Producción y Servicios						
	<b>DEPARTAMENTO :</b> Ingeniería de Sistemas e Informática			<b>ESCUELA :</b> Ciencia de la Computación			
	<b>PROFESOR :</b>						
	<b>TÍTULO :</b>						
	<b>ASIGNATURA :</b> Algoritmos Paralelos						
	<b>PREREQUISITO:</b> CS210T,CS225T		<b>CREDITOS:</b> 4		<b>Año:</b> 2010-1		<b>Total Horas:</b> 2 HT;
				<b>Sem:</b> 7 <sup>mo</sup> Semestre.		2 HT 2 HP 2 HL	
<b>Horario</b>		Lun	Mar	Mie	Jue	Vie	Sáb
<b>Total Semanal</b>							
<b>Aula</b>							

**2 Exposición de Motivos** Las arquitecturas de computadores están tendiendo a incluir cada vez más núcleos y/o procesadores método de incrementar la capacidad computacional de cada unidad. La posibilidad de realizar múltiplicamente mediante hardware no es inmediatamente traducida al software, pues las aplicaciones para aprovechar estas nuevas capacidades, mediante el uso de hebras y/o procesos.

- 2 Objetivo**
- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente máquinas con múltiples núcleos.
  - Que el alumno sea capaz de comparar aplicaciones secuenciales y paralelas.
  - Que el alumno sea capaz de convertir, cuando la situación lo amerite, aplicaciones secuenciales a paralelas de forma eficiente.

**3 Contenido Temático 3 CN/Computación Paralela.(5 horas)**

Objetivos Específicos	Contenido
<ul style="list-style-type: none"> <li>▪ Comparar y contrastar para computación paralela reconociendo las fortalezas y debilidades de cada una.</li> <li>▪ Comparar y contrastar paradigma de programación paralela reconociendo las fortalezas y debilidades de cada una.</li> <li>▪ Identificar las propiedades básicas de ancho de banda, latencia, escalabilidad, granularidad.</li> <li>▪ Diseñar, código, hacer test y depuración de computación paralela.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Revisi</li> <li>▪ Model</li> <li>▪ Tipos</li> <li>▪ Tareas</li> <li>▪ Datos</li> <li>▪ Cluste</li> <li>▪ Paradi lela.</li> <li>▪ Hebra</li> <li>▪ Paso c</li> <li>▪ Técnica</li> <li>▪ Arquitect (<i>MapReduce</i>).</li> <li>▪ Comp <i>ting</i>).</li> <li>▪ Comu tación ...).</li> </ul>
	[3]

<b>3 AR/Multiprocesamiento.(5 horas)</b>	<table border="1"> <thead> <tr> <th data-bbox="634 113 1138 149">Objetivos Específicos</th> <th data-bbox="1138 113 1624 149">Contenidos</th> </tr> </thead> <tbody> <tr> <td data-bbox="634 149 1138 1056"> <ul style="list-style-type: none"> <li>▪ Discutir el concepto de procesamiento paralelo y la relación entre paralelismo y desempeño.</li> <li>▪ Apreciar que los tipos de datos multimedia (ej. audio y datos visuales de 8/16 bits) pueden ser procesados en paralelo en registros de 64 bits para mejorar el desempeño.</li> <li>▪ Entender como el desempeño puede ser mejorado incorporando múltiples procesadores en un único chip.</li> <li>▪ Apreciar la necesidad de expresar algoritmos en una forma que permita la ejecución en procesadores paralelos.</li> <li>▪ Entender como los procesadores gráficos de propósito especial (GPUs) pueden acelerar el desempeño de aplicaciones gráficas.</li> <li>▪ Entender la organización de estructuras computacionales que puedan ser electronicamente configuradas y reconfiguradas.</li> </ul> </td> <td data-bbox="1138 149 1624 1056"> <ul style="list-style-type: none"> <li>▪ La ley de Amdahl.</li> <li>▪ Procesamiento en vectores pequeños (operaciones multimedia).</li> <li>▪ Procesadores Multinúcleos y Multihebras.</li> <li>▪ La taxonomía de Flynn: Estructuras multiprocesador y arquitecturas.</li> <li>▪ Sistemas de programación de múltiples procesadores.</li> <li>▪ GPU y procesadores gráficos de propósito especial.</li> <li>▪ Introducción a la lógica reconfigurable y procesadores de propósito especial.</li> </ul> <p>[3]</p> </td> </tr> </tbody> </table>	Objetivos Específicos	Contenidos	<ul style="list-style-type: none"> <li>▪ Discutir el concepto de procesamiento paralelo y la relación entre paralelismo y desempeño.</li> <li>▪ Apreciar que los tipos de datos multimedia (ej. audio y datos visuales de 8/16 bits) pueden ser procesados en paralelo en registros de 64 bits para mejorar el desempeño.</li> <li>▪ Entender como el desempeño puede ser mejorado incorporando múltiples procesadores en un único chip.</li> <li>▪ Apreciar la necesidad de expresar algoritmos en una forma que permita la ejecución en procesadores paralelos.</li> <li>▪ Entender como los procesadores gráficos de propósito especial (GPUs) pueden acelerar el desempeño de aplicaciones gráficas.</li> <li>▪ Entender la organización de estructuras computacionales que puedan ser electronicamente configuradas y reconfiguradas.</li> </ul>	<ul style="list-style-type: none"> <li>▪ La ley de Amdahl.</li> <li>▪ Procesamiento en vectores pequeños (operaciones multimedia).</li> <li>▪ Procesadores Multinúcleos y Multihebras.</li> <li>▪ La taxonomía de Flynn: Estructuras multiprocesador y arquitecturas.</li> <li>▪ Sistemas de programación de múltiples procesadores.</li> <li>▪ GPU y procesadores gráficos de propósito especial.</li> <li>▪ Introducción a la lógica reconfigurable y procesadores de propósito especial.</li> </ul> <p>[3]</p>	
	Objetivos Específicos	Contenidos				
<ul style="list-style-type: none"> <li>▪ Discutir el concepto de procesamiento paralelo y la relación entre paralelismo y desempeño.</li> <li>▪ Apreciar que los tipos de datos multimedia (ej. audio y datos visuales de 8/16 bits) pueden ser procesados en paralelo en registros de 64 bits para mejorar el desempeño.</li> <li>▪ Entender como el desempeño puede ser mejorado incorporando múltiples procesadores en un único chip.</li> <li>▪ Apreciar la necesidad de expresar algoritmos en una forma que permita la ejecución en procesadores paralelos.</li> <li>▪ Entender como los procesadores gráficos de propósito especial (GPUs) pueden acelerar el desempeño de aplicaciones gráficas.</li> <li>▪ Entender la organización de estructuras computacionales que puedan ser electronicamente configuradas y reconfiguradas.</li> </ul>	<ul style="list-style-type: none"> <li>▪ La ley de Amdahl.</li> <li>▪ Procesamiento en vectores pequeños (operaciones multimedia).</li> <li>▪ Procesadores Multinúcleos y Multihebras.</li> <li>▪ La taxonomía de Flynn: Estructuras multiprocesador y arquitecturas.</li> <li>▪ Sistemas de programación de múltiples procesadores.</li> <li>▪ GPU y procesadores gráficos de propósito especial.</li> <li>▪ Introducción a la lógica reconfigurable y procesadores de propósito especial.</li> </ul> <p>[3]</p>					
<b>3 AL/Algoritmos Paralelos.(3 horas)</b>	<table border="1"> <thead> <tr> <th data-bbox="634 1056 1138 1092">Objetivos Específicos</th> <th data-bbox="1138 1056 1624 1092">Contenidos</th> </tr> </thead> <tbody> <tr> <td data-bbox="634 1092 1138 1451"> <ul style="list-style-type: none"> <li>▪ Describir la implementación de listas enlazadas en un PRAM.</li> <li>▪ Usar operaciones paralelas para implementar cálculos simples eficientes en paralelo.</li> <li>▪ Explicar el teorema de Brent y su relevancia.</li> </ul> </td> <td data-bbox="1138 1092 1624 1451"> <ul style="list-style-type: none"> <li>▪ El modelo PRAM.</li> <li>▪ Lecturas y escrituras exclusivas y concurrentes.</li> <li>▪ Salto de punteros.</li> <li>▪ El teorema de Brent y el trabajo eficiente.</li> </ul> <p>[3]</p> </td> </tr> </tbody> </table>	Objetivos Específicos	Contenidos	<ul style="list-style-type: none"> <li>▪ Describir la implementación de listas enlazadas en un PRAM.</li> <li>▪ Usar operaciones paralelas para implementar cálculos simples eficientes en paralelo.</li> <li>▪ Explicar el teorema de Brent y su relevancia.</li> </ul>	<ul style="list-style-type: none"> <li>▪ El modelo PRAM.</li> <li>▪ Lecturas y escrituras exclusivas y concurrentes.</li> <li>▪ Salto de punteros.</li> <li>▪ El teorema de Brent y el trabajo eficiente.</li> </ul> <p>[3]</p>	
	Objetivos Específicos	Contenidos				
<ul style="list-style-type: none"> <li>▪ Describir la implementación de listas enlazadas en un PRAM.</li> <li>▪ Usar operaciones paralelas para implementar cálculos simples eficientes en paralelo.</li> <li>▪ Explicar el teorema de Brent y su relevancia.</li> </ul>	<ul style="list-style-type: none"> <li>▪ El modelo PRAM.</li> <li>▪ Lecturas y escrituras exclusivas y concurrentes.</li> <li>▪ Salto de punteros.</li> <li>▪ El teorema de Brent y el trabajo eficiente.</li> </ul> <p>[3]</p>					
<b>3 Modelos de Threads con PTHREADS (0 horas)</b>	<table border="1"> <thead> <tr> <th data-bbox="792 1451 1300 1486">Objetivos Específicos</th> <th data-bbox="1300 1451 1624 1486">Contenidos</th> </tr> </thead> <tbody> <tr> <td data-bbox="792 1486 1300 1837"> <ul style="list-style-type: none"> <li>▪ Entender los distintos modelos de programación paralela.</li> <li>▪ Conocer ventajas y desventajas de los distintos modelos de programación paralela.</li> </ul> </td> <td data-bbox="1300 1486 1624 1837"> <ul style="list-style-type: none"> <li>▪ ¿Qué es una hebra?</li> <li>▪ ¿Qué es pthread?</li> <li>▪ Diseñando programas</li> <li>▪ Creación y manejo de</li> <li>▪ Sincronización de hebras</li> </ul> <p>[4], [3]</p> </td> </tr> </tbody> </table>	Objetivos Específicos	Contenidos	<ul style="list-style-type: none"> <li>▪ Entender los distintos modelos de programación paralela.</li> <li>▪ Conocer ventajas y desventajas de los distintos modelos de programación paralela.</li> </ul>	<ul style="list-style-type: none"> <li>▪ ¿Qué es una hebra?</li> <li>▪ ¿Qué es pthread?</li> <li>▪ Diseñando programas</li> <li>▪ Creación y manejo de</li> <li>▪ Sincronización de hebras</li> </ul> <p>[4], [3]</p>	
	Objetivos Específicos	Contenidos				
<ul style="list-style-type: none"> <li>▪ Entender los distintos modelos de programación paralela.</li> <li>▪ Conocer ventajas y desventajas de los distintos modelos de programación paralela.</li> </ul>	<ul style="list-style-type: none"> <li>▪ ¿Qué es una hebra?</li> <li>▪ ¿Qué es pthread?</li> <li>▪ Diseñando programas</li> <li>▪ Creación y manejo de</li> <li>▪ Sincronización de hebras</li> </ul> <p>[4], [3]</p>					

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Implementar programas multihebras por medio de OpenMP.</li> <li>▪ Entender y aplicar conceptos de sincronización y trabajo compartido.</li> </ul>	<ul style="list-style-type: none"> <li>▪ ¿Qué es OpenMP?</li> <li>▪ El modelo de programación OpenMP.</li> <li>▪ Directivas de OpenMP.</li> <li>▪ Constructores de trabajo compartido.</li> <li>▪ Constructores de Tareas.</li> <li>▪ Constructores de sincronización.</li> <li>▪ Manejo de datos privados.</li> </ul> <p>[1], [3]</p>

### 3 Modelos de Threads con OpenMP (0 horas)

Objetivos Específicos
<ul style="list-style-type: none"> <li>▪ Implementar programas multihebras por medio de OpenMP.</li> <li>▪ Entender y aplicar conceptos de sincronización y trabajo compartido.</li> </ul>

### 3 Modelo de programación mediante paso de Mensajes con MPI (0 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Entender y aplicar el modelo de datos paralelos utilizando la herramienta TBB.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Bucles Simples Paralelos.</li> <li>▪ Bucles Complejos Paralelos.</li> <li>▪ Cancelación y Exepciones.</li> <li>▪ Contenedores paralelos.</li> </ul> <p>[5], [3]</p>

### 3 *Threading Building Blocks (TBB)* (0 horas)

#### 4 Actividades

- Asignaciones
- Controles de Lectura
- Exposiciones

#### 5 Recursos Materiales

- Apuntes del curso
- Libro(s) de la bibliografía

## 6 Metodología

- Clase Magistral.
- Taller didáctico.
- Social Constructivismo.
- Prácticas personales y en grupo.

## 7 Evaluación

La nota final ( $NF$ ) se obtiene de la siguiente manera:

**NE** Nota de Exámenes 60 %, esta nota se divide en

- Exámen Parcial 40 %
- Examen Final 60 %

**NT** Nota de Trabajos e Intervención en clase 40 %

$$NF = 0,6 * NE + 0,4 * NT$$

## Referencias

- [1] Rohit Chandra, Ramesh Menon, David Kohr Leo Dagum, Dror Maydan, and Jeff McDonald. *Parallel Programming in OpenMP*. Morgan Kaufmann, 1 edition, 2000.
- [2] George Em Karniadakis and Robert M. Kirby II. *Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation*. Cambridge University Press, 1 edition, 2003.
- [3] Calvin Lin and Larry Snyder. *Principles of Parallel Programming*. Addison Wesley, 1 edition, 2008.
- [4] Bradford Nichols, Dick Buttlar, and Jacqueline Proulx Farrell. *Pthreads Programming: A POSIX Standard for Better Multiprocessing*. O'Reilly Media, Inc., 1 edition, 1996.
- [5] James Reinders. *Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism*. O'Reilly Media, Inc., 1 edition, 2007.

---

Docente del curso