Centro Universitario de Tonalá

PROGRAMA DE ESTUDIOS					
Nombre de la unidad de aprendizaje					
Programación Paralela (Arquitectura multicore)					
Modalidad:					
Presencial					
Departamento:					
Departamento de Ciencias de la Información y Desarrollos Tecnológicos					
Academia					
Academia de Programación					
Área de Formación					
Área de Formación Especializante Obligatoria					
Clave de la materia:	**Nivel:**	**Prerrequisitos**	**Co-requisitos**	**Tipo de asignatura**	**Tipo de curso:**
I4902	Licenciatura	I4313 (Programación I), I4861 (Álgebra Lineal)			C= curso
Hrs. /semestre	**Horas semana**	**Horas de teoría:**	**Horas de práctica:**	**Total de horas:**	**Valor de créditos:**
	4	**60**	**20**	**80**	**9**

Objetivo de la asignatura

El alumno realizará programas con distintos niveles de paralelismo, utilizando de manera eficiente tanto las estructuras heterogéneas de hardware (CPUs y GPUs), así como los distintos niveles de memoria.

Aportación de la asignatura al perfil de egreso

Campo de aplicación profesional

El alumno podrá desempeñarse en organizaciones donde se requiera procesar e interpretar datos masivos que provengan de más de una fuente, con el fin de tomar decisiones acertadas en tiempos cortos.

Perfil deseable del docente para impartir la asignatura

1. Promueve la enseñanza y el aprendizaje en la labor docente mediante una visión constructivista y centrado en el estudiante.
2. Utiliza tanto las Tecnologías de la Información y la Comunicación (TIC) como los métodos tradicionales en la labor docente.
3. Apoya al estudiante en la construcción de competencias profesionales y

genéricas mediante la planeación correcta de las actividades a desarrollar en cada Unidad de Aprendizaje.

4. Es experto en el área de programación utilizando cómputo paralelo multinúcleo.

UNIDAD 1	
Conceptos Básicos	
OBJETIVO	
El alumno entenderá los conceptos fundamentales relacionados con el cómputo paralelo multinúcleo, diferenciándolo del cómputo paralelo tradicional.	
CONTENIDO	
1.1	Introducción
1.2	Plataformas actuales
1.3	Terminología
1.4	Modelos de programación concurrente y paralelo
1.5	Tipos de paralelismo
Referencias a fuentes de información básicas	
[1] Shane Cook, (2013), CUDA Programming, a developer's guide to parallel computing with GPUs, Elsevier-Morgan Kaufmann. [2] CUDA C Programming Guide, NVIDIA.	
Referencias a fuentes de información complementarias	
[1] Rob Farber, (2011), CUDA Application design and Development, Elsevier-Morgan Kaufmann. [2] Norman Matloff, (2016), Parallel Computing for Data Science, CRC Press.	

UNIDAD 2	
Evolución del cómputo paralelo usando GPU	
OBJETIVO	
El alumno conocerá los principales conceptos de la programación con hardware gráfico, sus inicios, así como también su modelo de programación.	
CONTENIDO	
2.1	Ley de Moore
2.2	Cambio de paradigma: incremento en el rendimiento partiendo del software
2.3	Breve taxonomía de lenguajes de programación en paralelo
2.4	El estándar CUDA
2.5	Hilos, memoria compartida y kernel
2.6	El ambiente de ejecución
2.7	Niveles de memoria
Referencias a fuentes de información básicas	
[1] Shane Cook, (2013), CUDA Programming, a developer's guide to parallel computing with GPUs, Elsevier-Morgan Kaufmann. [2] CUDA C Programming Guide, NVIDIA.	
Referencias a fuentes de información complementarias	
[1] Rob Farber, (2011), CUDA Application design and Development, Elsevier-Morgan Kaufmann. [2] Norman Matloff, (2016), Parallel Computing for Data Science, CRC Press.	

UNIDAD 3	
Hardware y CUDA	
OBJETIVO	
El alumno conocerá algunas de las distintas plataformas multinúcleo utilizadas para programación de cálculos intensivos.	
CONTENIDO	
3.1	Arquitecturas basadas en Central Processing Units (CPUs) multinúcleo
3.2	Arquitecturas basadas en General Purpose Graphic Processing Units (GPGPUs)
3.3	Arquitecturas embebidas
3.4	Arquitecturas de hardware heterogéneas
Referencias a fuentes de información básicas	
[1] Shane Cook, (2013), CUDA Programming, a developer's guide to parallel computing with GPUs, Elsevier-Morgan Kaufmann. [2] CUDA C Programming Guide, NVIDIA, 2018.	
Referencias a fuentes de información complementarias	
[1] Rob Farber, (2011), CUDA Application design and Development, Elsevier-Morgan Kaufmann. [2] Norman Matloff, (2016), Parallel Computing for Data Science, CRC Press.	

UNIDAD 4	
Modelo de concurrencia y modelo de ejecución	
OBJETIVO	
El alumno profundizará en el conocimiento de los distintos niveles de memoria utilizados en los modelos de concurrencia y ejecución.	
CONTENIDO	
4.1	Modelo de concurrencia
4.2	Modelo de tiempo de ejecución
4.3	Grupos de trabajo y el dominio de ejecución
4.4	Sincronización
4.5	Modelo de memoria del huésped
4.6	Modelo de memoria del dispositivo
Referencias a fuentes de información básicas	
[1] Shane Cook, (2013), CUDA Programming, a developer's guide to parallel computing with GPUs, Elsevier-Morgan Kaufmann. [2] CUDA C Programming Guide, NVIDIA, 2018.	
Referencias a fuentes de información complementarias	
[1] Rob Farber, (2011), CUDA Application design and Development, Elsevier-Morgan Kaufmann. [2] Norman Matloff, (2016), Parallel Computing for Data Science, CRC Press.	

UNIDAD 5	
Manejo de datos	
OBJETIVO	
El alumno entenderá el impacto de los distintos tipos de datos en el uso de la memoria del dispositivo.	
CONTENIDO	

5.1	Manejo de memoria
5.2	Transferencia de datos en ambientes discretos
5.3	Colocación de datos en ambientes de memoria compartida

Referencias a fuentes de información básicas

[1] Shane Cook, (2013), CUDA Programming, a developer's guide to parallel computing with GPUs, Elsevier-Morgan Kaufmann.
[2] CUDA C Programming Guide, NVIDIA, 2018.

Referencias a fuentes de información complementarias

[1] Rob Farber, (2011), CUDA Application design and Development, Elsevier-Morgan Kaufmann.
[2] Norman Matloff, (2016), Parallel Computing for Data Science, CRC Press.

Actividades de aprendizaje

Las sesiones son teórico-prácticas, con explicaciones y ejemplos de los conceptos fundamentales del cómputo paralelo, para posteriormente realizar modificaciones, menores pero significativas para el aprendizaje del alumno, de tales ejemplos mostrados en clase.

Material y ambiente del aprendizaje

Se considera el uso de computadoras con al menos una unidad de procesamiento gráfico de marca NVIDIA instalada, así como el software para desarrollo CUDA TOOLKIT (versión más reciente). Además, se considera el uso de la Guía de programación de CUDA del año en curso, o cuando menos del año anterior.

Evaluación del aprendizaje

Exámenes y proyectos parciales

Criterio de evaluación	Porcentaje
Proyectos parciales con reportes	60%
Exámenes departamentales y parciales	20%
Proyecto final con reporte	20%

Participantes en la elaboración del programa

Código	Nombre completo	Fecha de elaboración del programa
2907062	JOSÉ VALENTÍN OSUNA ENCISO	28/AGO/2018

Aprobó y revisó la academia de:	Fecha de aprobación	Fecha de próxima revisión
PROGRAMACIÓN	13/NOV/2017	12/NOV/2018