



## Programa de asignatura por competencias de educación superior

### Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

<b>Actualización:</b>	Junio 07, 2022				
<b>Carrera:</b>	Ingeniería en Desarrollo de Software	<b>Asignatura:</b>	Computación paralela		
<b>Academia:</b>	Ciencias Computacionales y Programación /	<b>Clave:</b>	19SDS18		
<b>Módulo formativo:</b>	Programación aplicada	<b>Seriación:</b>	19SDS29 - Internet de las cosas		
<b>Tipo de curso:</b>	Presencial	<b>Prerrequisito:</b>	19SDS15 - Estructuras de datos y algoritmia		
<b>Semestre:</b>	Quinto	<b>Créditos:</b>	6.75	<b>Horas semestre:</b>	108 horas
<b>Teoría:</b>	2 horas	<b>Práctica:</b>	2 horas	<b>Trabajo indpt.:</b>	2 horas
				<b>Total x semana:</b>	6 horas

## Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	Los egresados gestionarán recursos relacionados con el desarrollo de software en alguna organización.	Los egresados podrán aplicar metodologías en el desarrollo de proyectos en el contexto laboral.	20% de los egresados aplicarán metodologías en el desarrollo de software en su contexto laboral.
OE2	Los egresados diseñarán e implementarán soluciones innovadoras mediante el uso de tecnologías de la información.	Los egresados participarán activamente en el ciclo de desarrollo e integración continuos	25% de los egresados desempeñarán labores de desarrollo e integración continuos.
OE3	Los egresados desarrollarán conocimiento especializado que les permite enfocarse en un área del conocimiento específico del desarrollo de software.	Los egresados desempeñarán actividades orientadas al aseguramiento de los activos de información de manera resiliente, la gestión de la infraestructura de redes y comunicaciones, o integrando hardware y software para crear soluciones IoT; así como el uso de inteligencia artificial para gestionar datos y reconocer patrones que determinen oportunidades de negocio en las organizaciones.	5% de los egresados desempeñarán labores en desarrollo de soluciones IoT.
OE5	Los egresados serán capaces de emprender un negocio basado en el desarrollo de un producto o servicio de tecnologías de la información, aportando valor a la generación de empleos e incrementar el bienestar económico y social, de forma ecológica y sustentable.	Los egresados serán capaces de emprender un negocio basado en el desarrollo propio de un producto o servicio de tecnologías de la información.	2% de los egresados tendrán participación en el acta constitutiva de una empresa creada a partir del desarrollo de software para ofrecer un producto o servicio.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Aplicar los conocimientos de ciencias básicas como física y matemáticas, así como las ciencias de la ingeniería para generar nuevos productos o servicios basándose en la innovación tecnológica.	Desarrollará software para resolver diferentes problemas del área de ingeniería que implique altos niveles de procesamiento, diseñando algoritmos para el procesamiento concurrente y/o paralelo, así como implementando sistemas de cómputo de alto desempeño.	<p>2 Programación concurrente.</p> <p>2.1 Programación multi-hilos.</p> <p>2.2 Funcionalidad vs rendimiento.</p> <p>2.3 Aceleración y eficiencia.</p> <p>2.4 Paso de mensajes.</p> <p>3 Procesamiento paralelo.</p> <p>3.1 RMI, sockets, CORBA.</p> <p>3.2 High Performance Computing (OpenHPC).</p> <p>3.3 NFS para HPC clusters, data centers (FTP, MySQL/MariaDB, Apache).</p>
AE2	Aplicar y analizar procesos de diseño de ingeniería para generar una experiencia de usuario que asegure cubrir las necesidades como las expectativas de clientes y partes interesadas, utilizando y gestionando la infraestructura de red necesaria.	Diseñará software para resolver diferentes problemas del área de ingeniería que implique altos niveles de procesamiento, algoritmos para el procesamiento concurrente y/o paralelo, así como implementando sistemas de cómputo de alto desempeño.	<p>1 Introducción a los procesos.</p> <p>1.1 Modelo de procesos y estados.</p> <p>1.2 Comunicación entre procesos.</p> <p>1.3 Condiciones de competencia, exclusión mutua.</p> <p>1.4 Planificación de procesos.</p> <p>1.5 Arquitectura de microprocesadores (single core y multi core).</p> <p>1.6 Memoria compartida y distribuida.</p> <p>1.7 Programación concurrente y sistemas distribuidos.</p> <p>1.8 Contextos de computación paralela (cloud, cluster, grid, P2P, load balancing, high availability).</p>
AE3	Desarrollar una experimentación adecuada para recopilar, almacenar y analizar grandes cantidades de información basándose en el juicio ingenieril para crear productos o servicios innovadores mediados por software.	Implementará mecanismo de análisis y pruebas (testing) de algoritmos y software en el ámbito de procesamiento concurrente, paralelo y cómputo de alto rendimiento.	<p>2 Programación concurrente.</p> <p>2.1 Programación multi-hilos.</p> <p>2.2 Funcionalidad vs rendimiento.</p> <p>2.3 Aceleración y eficiencia.</p>



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			2.4 Paso de mensajes.  3 Procesamiento paralelo. 3.1 RMI, sockets, CORBA. 3.2 High Performance Computing (OpenHPC). 3.3 NFS para HPC clusters, data centers (FTP, MySQL/MariaDB, Apache).
AE6	Reconocer la mejora continua como parte de su desarrollo profesional para mantener un perfil actualizado en desarrollo de software para el diseño e implementación de productos y servicios basados en tecnologías con las tendencias emergentes.	Diseñará procesos personales y de equipo de desarrollo de software que permita la evaluación y retroalimentación de su propuesta y desarrollo de sistemas relacionados con la programación concurrente, paralela y cómputo de alto rendimiento.	1 Introducción a los procesos. 1.1 Modelo de procesos y estados. 1.2 Comunicación entre procesos. 1.3 Condiciones de competencia, exclusión mutua. 1.4 Planificación de procesos. 1.5 Arquitectura de microprocesadores (single core y multi core). 1.6 Memoria compartida y distribuida. 1.7 Programación concurrente y sistemas distribuidos. 1.8 Contextos de computación paralela (cloud, cluster, grid, P2P, load balancing, high availability).  2 Programación concurrente. 2.1 Programación multi-hilos. 2.2 Funcionalidad vs rendimiento. 2.3 Aceleración y eficiencia. 2.4 Paso de mensajes.  3 Procesamiento paralelo. 3.1 RMI, sockets, CORBA. 3.2 High Performance Computing (OpenHPC).



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			3.3 NFS para HPC clusters, data centers (FTP, MySQL/MariaDB, Apache).

### Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Conocer e implementar conceptos de sistemas en ambientes de cómputo de alto rendimiento (High Performance Computing, HPC).		
Atributos (competencia específica) de la asignatura		
Diseñar algoritmos, codificación, lenguajes y tecnologías que permitan implementar sistemas de cómputo de alto desempeño.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
Conocer los conceptos relacionados con la distribución de procesamiento y la comunicación entre procesos que se requiere para el diseño e implementación de sistemas de alto desempeño.	Diseñar e implementar algoritmos relacionados con el procesamiento paralelo y sistemas de cómputo de alto rendimiento.  Utilizar las diferentes técnicas, sistemas, herramientas y lenguajes de programación diseñados para el procesamiento en paralelo y sistemas de cómputo de alto rendimiento.	Trabajo en equipo. Comunicación personal.
Producto integrador de la asignatura, considerando los avances por unidad		
Desarrollo de software implementando los conocimientos y conceptos del cómputo de alto desempeño.		

## Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Introducción a los procesos."

<b>Número y nombre de la unidad:</b> 1. Introducción a los procesos.							
<b>Tiempo y porcentaje para esta unidad:</b>		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
<b>Aprendizajes esperados:</b> Conocer los conceptos de procesamiento y las características de los procesos, para su implementación en el desarrollo de software.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1.1 Modelo de procesos y estados. 1.2 Comunicación entre procesos. 1.3 Condiciones de competencia, exclusión mutua. 1.4 Planificación de procesos. 1.5 Arquitectura de microprocesadores (single core y multi core). 1.6 Memoria compartida y distribuida. 1.7 Programación concurrente y sistemas distribuidos. 1.8 Contextos de computación paralela (cloud, cluster, grid, P2P, load balancing, high availability).	Saber: - Conocer los conceptos relacionados con el procesamiento y programación distribuida y de procesos, relacionados con la implementación de sistemas de cómputo de alto rendimiento.  Saber hacer: - Diseñar y programar algoritmos que implementan los conceptos de administración y planificación de procesos. - Desarrollar la capacidad de análisis y diseño de algoritmos.  Ser: Reflexión sobre el impacto de las	-Presentación de material didáctico a través de diferentes medios (diapositivas, videos, videoprojector, equipo de cómputo, acceso a internet, etc.) -Actividades de clasificación y análisis de información. - Proyecto de investigación y aplicación.	Evaluación formativa: -Mapas mentales. -Presentación y exposición. -Investigación de conceptos y tecnologías.  Evaluación sumativa: -Examen conceptual. -Proyecto de diseño y programación de problemas relacionados con la administración de procesos.	Diseño y programación de problemas relacionados con la administración de procesos: - Productor Consumidor. - Lectores y escritores. - Cena de los filósofos. - Barbero dormilón.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Introducción a los procesos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	problemáticas en el contexto y su vida cotidiana.			
<b>Bibliografía</b>				
<p>- Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. 2ª Edición. United States: Pearson.</p> <p>- Quinn, M. (2004). Parallel Programming in C with MPI and OpenMP. United States: McGraw Hill.</p> <p>- Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. United States: Addison Wesley.</p> <p>- Pacheco, P.S. (1997). Parallel Programming with MPI. United States: Morgan Kaufmann.</p> <p>- Tanenbaum, A. ; Van Steen, M. (2002). Distributed Systems: Principles and Paradigms. United States: Prentice Hall.</p>				

## Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Programación concurrente."

<b>Número y nombre de la unidad:</b> 2. Programación concurrente.							
<b>Tiempo y porcentaje para esta unidad:</b>		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
<b>Aprendizajes esperados:</b> Trabajar con la programación concurrente con hilos de procesamiento (threads) para su implementación en el desarrollo de software.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2.1 Programación multi-hilos. 2.2 Funcionalidad vs rendimiento. 2.3 Aceleración y eficiencia. 2.4 Paso de mensajes.	<p>Saber:</p> <ul style="list-style-type: none"> <li>- Conocer e implementar los conceptos de programación concurrente mediante hilos de procesamiento (threads).</li> </ul> <p>Saber hacer:</p> <ul style="list-style-type: none"> <li>- Diseñar y programar algoritmos que implementan los conceptos de administración y planificación de hilos de procesamiento (threads) y procesamiento concurrente.</li> <li>- Desarrollar la capacidad de análisis y diseño de algoritmos, así como planificación.</li> </ul> <p>Ser:</p> <ul style="list-style-type: none"> <li>- Reflexión sobre el impacto de las</li> </ul>	<ul style="list-style-type: none"> <li>- Presentación de material didáctico a través de diferentes medios (diapositivas, guías didácticas, videos, videoprojector, equipo de cómputo, acceso a internet, etc.)</li> <li>- Actividades de clasificación y análisis de información.</li> <li>- Proyecto de investigación y aplicación.</li> </ul>	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> <li>- Prácticas de laboratorio</li> <li>- Caso de estudio</li> </ul> <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> <li>- Examen conceptual.</li> <li>- Proyecto de implementación.</li> </ul>	<p>Desarrollo de aplicaciones implementando la programación de hilos de ejecución (threads)</p>			



Continuación: Tabla 4.2. Desglose específico de la unidad "Programación concurrente."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	problemáticas en el contexto y su vida cotidiana.			

**Bibliografía**

- Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. 2ª Edición. United States: Pearson.
- Quinn, M. (2004). Parallel Programming in C with MPI and OpenMP. United States: McGraw Hill.
- Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. United States: Addison Wesley.
- Pacheco, P.S. (1997). Parallel Programming with MPI. United States: Morgan Kaufmann.
- Tanenbaum, A. ; Van Steen, M. (2002). Distributed Systems: Principles and Paradigms. United States: Prentice Hall.

## Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Procesamiento paralelo."

<b>Número y nombre de la unidad:</b> 3. Procesamiento paralelo.							
<b>Tiempo y porcentaje para esta unidad:</b>		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
<b>Aprendizajes esperados:</b>		Desarrollar un software mediante la programación y procesamiento paralelo, para su implementación en el desarrollo de sistemas de información distribuidos.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
3.1 RMI, sockets, CORBA. 3.2 High Performance Computing (OpenHPC). 3.3 NFS para HPC clusters, data centers (FTP, MySQL/MariaDB, Apache).	<p>Saber:</p> <ul style="list-style-type: none"> <li>- Conocer los conceptos de programación distribuida mediante diferentes tecnologías de comunicación y ejecución remota de métodos, procesos y/o servicios, así como la instalación, configuración e implementación de sistemas distribuidos y de alto rendimiento.</li> </ul> <p>Saber hacer:</p> <ul style="list-style-type: none"> <li>- Diseñar, implementar, instalar, configurar y programar sistemas distribuidos y de alto rendimiento.</li> <li>- Desarrollar la capacidad de análisis y diseño, así como planificación.</li> </ul>	<ul style="list-style-type: none"> <li>-Presentación de material didáctico a través de diferentes medios (diapositivas, guías didácticas, vídeos, videoprojector, equipode cómputo, acceso a internet, etc.)</li> <li>- Actividades de clasificación y análisis de información.</li> <li>- Proyecto de investigación y aplicación.</li> </ul>	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> <li>- Prácticas de laboratorio</li> <li>- Caso de estudio</li> </ul> <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> <li>-Examen conceptual.</li> <li>- Proyecto de implementación.</li> </ul>	Desarrollo de aplicaciones implementando la programación y procesamiento paralelo ensistemas de cómputo de alto desempeño.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Procesamiento paralelo."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: Reflexión sobre el impacto de las problemáticas en el contexto y su vida cotidiana.			
<b>Bibliografía</b>				
<ul style="list-style-type: none"> <li>- Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. 2ª Edición. United States: Pearson.</li> <li>- Quinn, M. (2004). Parallel Programming in C with MPI and OpenMP. United States: McGraw Hill.</li> <li>- Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. United States: Addison Wesley.</li> <li>- Pacheco, P.S. (1997). Parallel Programming with MPI. United States: Morgan Kaufmann.</li> <li>- Tanenbaum, A. ; Van Steen, M. (2002). Distributed Systems: Principles and Paradigms. United States: Prentice Hall.</li> </ul>				



## V. Perfil docente

Tabla 5. Descripción del perfil docente

<b>Perfil deseable docente para impartir la asignatura</b>
<p>Carrera(s): Especialización, maestría y/o doctorado en el área de la materia a impartir. o carrera afín</p> <ul style="list-style-type: none"><li>- Con experiencia especializada en el campo y docente, deseable de 2 años. Con habilidades pedagógicas y uso de metodologías alternativas de enseñanza.</li><li>- Experiencia mínima de dos años</li><li>- Mínimo Maestría, deseable doctorado.</li></ul>