

Unidad de Aprendizaje				
Programación de Sistemas Avanzados				
Tipo de UA	Valor de créditos	Horas Semana	Horas teoría/semestre	Horas práctica/semestre
Presencial	8	4	40	40
<b>Departamento</b>		<b>Academia</b>		
Ciencias Computacionales		Software de Sistemas		
Objetivos de aprendizaje				
Desarrollar soluciones de software robustas, seguras y eficientes para sistemas computacionales avanzados, integrando técnicas modernas de programación concurrente, paralela y en tiempo real, con enfoque en plataformas distribuidas, sistemas embebidos y tecnologías emergentes como IoT y edge computing.				
Competencia de la Unidad de Aprendizaje				
CE.IC.194 Capacidad para diseñar, implementar y optimizar software de sistemas y de comunicaciones, aplicando principios de concurrencia, tolerancia a fallos, seguridad y eficiencia energética (BOE/SFIA/ CE.IC.194)				
Atributos de la competencia de UA				
Conocimientos (saber)	Habilidades (saber hacer)		Actitudes / Valores (saber ser)	
<p>C1. Fundamentos teóricos de la programación en tiempo real y sistemas distribuidos.</p> <p>C2. Complejidad algorítmica y optimización de recursos.</p> <p>C3. Programación concurrente, paralela y manejo de hilos.</p> <p>C4. Uso de herramientas modernas como Docker, Kubernetes y frameworks de paralelismo (OpenMP, MPI).</p>	<p>H1. Diseñar e implementar soluciones robustas y eficientes para sistemas avanzados.</p> <p>H2. Optimizar el rendimiento de aplicaciones en entornos paralelos y distribuidos.</p> <p>H3. Desarrollar drivers y software de bajo nivel para sistemas embebidos.</p>		<p>V1. <b>Asertividad</b> para expresarse adecuadamente y favorecer la interacción en grupos de trabajo.</p> <p>V2. <b>Resiliencia</b> para perseverar con actitud positiva ante los retos.</p> <p>V3. <b>Iniciativa, Autonomía y Responsabilidad Personal</b> que le permita responder a un mundo global y cambiante.</p> <p>V4. <b>Creatividad y pensamiento emprendedor</b> que le permita aprovechar oportunidades y apertura a nuevas opciones.</p> <p>V5. <b>Pensamiento crítico</b> para analizar e interpretar información de forma objetiva.</p> <p>V6. <b>Resolución de problemas</b> que le permita encontrar soluciones a distintos niveles</p>	

por medio de sus conocimientos especializados.

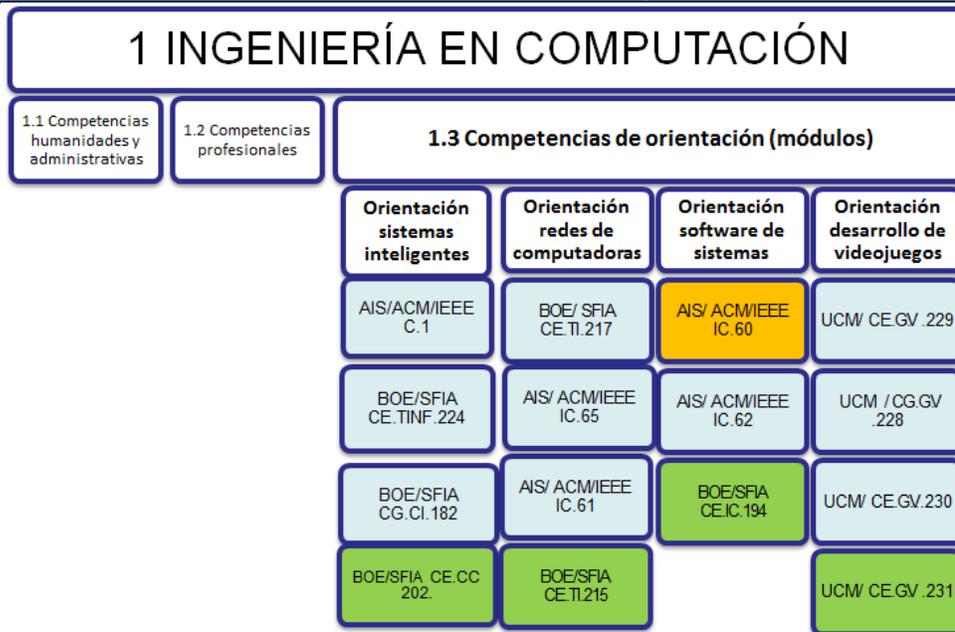
### Competencia Precedente de la Unidad de Aprendizaje

IC.62 Analizar, evaluar y seleccionar plataformas de hardware y software adecuadas para soporte de aplicaciones y sistemas embebidos en tiempo real. (AIS/ACM/IEEE IC.62)

### Competencia Consecuente de la Unidad de Aprendizaje

NO APLICA

### Estructura Conceptual



### Descripción

La presente Unidad de Aprendizaje (UA) es una asignatura teórica-práctica de la Licenciatura en Ingeniería en Computación, pertenece al Área de Formación Especializante, es la tercera UA del módulo Software de Sistemas y está diseñada para la aplicación de elementos teóricos en la programación de tiempo real: programación concurrente, comunicación y sincronización, fiabilidad y tolerancia a fallos, manejo de excepciones, uso del tiempo.

En esta UA trabajará el desarrollo de habilidades en el uso del lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo.

Esta UA aporta al Perfil de Egreso los conocimientos y habilidades necesarias para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente en los sistemas computacionales avanzados, y comprender los fundamentos teóricos de la programación en tiempo real (sistemas real-time), abordando aspectos clave como la temporización, la planificación de tareas y la optimización del uso del tiempo de CPU.

Además, analizará la complejidad algorítmica y su impacto en el rendimiento, comparando entornos secuenciales y paralelos para seleccionar las soluciones más eficientes. Dominará los conceptos de programación concurrente y paralela, diferenciando entre concurrencia (progreso simultáneo de múltiples tareas lógicas) y paralelismo (ejecución física simultánea en múltiples núcleos o procesadores). Evaluará diversas arquitecturas paralelas y distribuidas (multi-hilo, multinúcleo, clústeres y sistemas embebidos) para elegir la plataforma más adecuada según el problema a resolver. Utilizará herramientas y técnicas de sincronización, como semáforos, locks y monitores, para coordinar hilos y procesos, evitando condiciones de carrera y garantizando la consistencia de los datos compartidos. Asimismo, desarrollará habilidades de pensamiento crítico y resolución de problemas, aplicando creatividad e innovación para diseñar soluciones eficientes en programación de sistemas avanzados, fomentando el trabajo colaborativo y efectivo en equipo.

El curso: es una estrategia de tipo teórica, basada en un modelo de enseñanza aprendizaje que promueve en los estudiantes la estructuración consciente de su forma de aprehender, reflexionar, actuar, y organizar su conocimiento; el docente guía y comunica ciertos conocimientos para el logro de los objetivos educativos; requiere de una planeación previa en cuanto al objeto de estudio en particular y su importancia dentro del perfil del egresado, además, diseña las estrategias idóneas y selecciona los materiales necesarios para lograr la formación integral de los estudiantes (conocimientos, habilidades y actitudes) de conformidad al perfil del egresado.

El taller: es una estrategia de enseñanza grupal orientada a aprender mediante la acción, “aprender haciendo”, en la cual se privilegia el aprendizaje sobre la enseñanza, con el propósito de favorecer el desarrollo de habilidades sobre la base de conocimientos previos. Se requiere de metodologías participativas en la que se enseñe y aprenda a través de una tarea conjunta, para promover saberes de tipo cognitivo, procedimental y actitudinal como atributos de competencias de comunicación, trabajo colaborativo, resolución de problemas y de logro profesional.

El **curso-taller** es una mezcla de ambos conceptos.

Contenidos	Atributos			Productos del aprendizaje
	Saber	Saber hacer	Saber ser	
1. Arquitectura de sistemas paralelos y distribuidos	C1	H1	V1	Diseñar e implementar una solución acorde al cómputo distribuido y paralelo que cumpla las especificaciones eficiencia.
2. Programación concurrente y paralela	C1,C2	H1	V1, V2	
3. Programación en tiempo real	C3	H2	V3	Documentación detallada de los proyectos desarrollados, incluyendo análisis de rendimiento y optimización.
4. Tecnologías emergentes	C4	H2	V5	Documentación detallada de los proyectos desarrollados, incluyendo análisis de rendimiento y optimización.
5. Optimización y rendimiento	C4	H3	V6	Diseño e implementación de un sistema paralelo o distribuido que resuelva un problema real, utilizando tecnologías modernas

				como Docker, Kubernetes o frameworks de paralelismo.
Estrategias de enseñanza-aprendizaje				
Estrategias	Se utiliza para	Sección		
Aprendizaje basado en problemas ABP	Adquirir conocimientos, habilidades y actitudes en grupos pequeños para determinados objetivos de aprendizaje o resolución de problemas.	x		
Relatorías	Adquirir vocabulario, argumentar ideas y fomentar el pensamiento crítico.			
Seminarios	Ampliar información a profundidad, asignar distintos roles, promover las habilidades para la comunicación asertiva.			
Taller Reflexivo	Cohesión de grupo, análisis y organización de información, cambio de actitud o hábitos.			
Simulación de procesos	Construcción de conocimientos, desarrollo de habilidades y de actitudes en situaciones simuladas de la realidad.	x		
Panel	Exponer ideas de un tema sobre la base del diálogo y la comunicación asertiva. Estimular el pensamiento crítico a partir del intercambio de ideas y puntos de vista distintos.			
Mapas mentales	Favorecer la memorización, organización y representación de la información.			
Investigación de tópicos y problemas específicos	Formular problemas, confrontar hipótesis, planificar actividades, socializar conclusiones y resultados.	x		
Mapas y redes conceptuales	Incorporar nuevos conceptos, la construcción grupal y revisión de conocimientos o procedimientos, exposición y relaciones semánticas entre los conceptos.	x		
Resúmenes	Lectura y comprensión de información, para su organización sintética a partir de la identificación de ideas principales y sus nexos. Desarrolla la memorización y la organización adecuada de información.	x		
Método de proyectos	Organizar conocimiento, teóricos y prácticos, así como as relaciones entre hechos, conceptos, procedimientos, demostración y diseño de modelos, búsqueda y manejo de información, dependiendo del tipo de proyecto.	x		
Elaboración de artículos	Organizar y comunicar información sobre resultados de una investigación realizada o de un planteamiento teórico o procedimental, de algún tema específico.			
Entrevista	Profundización de un tema, identificación de un problema. Favorece la comunicación asertiva, el uso adecuado del lenguaje, así como la habilidad para la escucha activa y el manejo eficaz de información.			
Ensayo	Promover el conocimiento reflexivo, la capacidad de comunicación, el análisis y conocimiento profundo de una temática.	x		
Estudio de casos	Estudio de un fenómeno o un problema, precisa de un proceso de búsqueda o indagación.	x		

Otras			
Estrategias para la Evaluación de Saberes			Selección
Saber			
Evaluación de conceptos, principios, teorías y leyes	Nivel de comprensión y aplicación	Ensayos	<b>x</b>
		Entrevistas	
		Lista de cotejo	
		Trabajos prácticos o de ejecución	<b>x</b>
		Otros	
Saber hacer			
Evaluación de habilidades	Nivel de dominio de una técnica o actividad	Autoevaluación	<b>x</b>
		Escala de actitudes	
		Lista de cotejo	
		Pruebas de ejecución	
		Pruebas orales	
		Técnicas de observación	
		Trabajos prácticos	
		Otros	<b>x</b>
Saber ser			
Evaluación de actitudes y valores	Nivel de adquisición o	Escala de observación	<b>x</b>
		Instrumentos de auto-informe	<b>x</b>
		Lista de control	
		Registro anecdótico	
		Rúbricas	<b>x</b>
		Escala de actitudes tipo Likert	
		Otros	
Bibliografía			
<b>Libros:</b>			
Prasad, Gupta, Rosenberg, Sussman, Weems. Topics in Parallel and Distributed Computing: Introducing Concurrency in Undergraduate Courses. Morgan Kaufmann, 2015.			
Beck, Leland. Software de Sistemas. Addison-Wesley.			
Herlihy, Maurice, and Nir Shavit. The Art of Multiprocessor Programming. Morgan Kaufmann, 2021.			
<b>Recursos en Línea:</b>			
Documentación oficial de Docker y Kubernetes.			
Tutoriales y guías de OpenMP y MPI.			
Cursos en línea sobre programación en GPU con CUDA.			
<b>Artículos Científicos:</b>			
Investigaciones recientes sobre computación en la nube, edge computing y sistemas embebidos.			

<b>Criterios de evaluación</b>
Proyecto Final: 40%
Exámenes Teóricos y Prácticos: 30%
Participación en Talleres y Seminarios: 20%
Evaluación de Actitudes y Trabajo en Equipo: 10%
<b>Fecha de elaboración</b>
Marzo de 2025
<b>Participantes de la elaboración</b>
<b>Nombre</b>
Jose Juan Meza Espinosa
Martín García Hernández
Michel Emanuel López Franco