


<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p style="text-align: center;"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</b></p> <p style="text-align: center;"><b>ARQUITECTURA DE COMPUTADORAS II</b></p>	<b>DES:</b>	<b>INGENIERÍA</b>
	<b>Programa Educativo</b>	Ingeniería en Sistemas Computacionales en Hardware
	<b>Tipo de materia (Obli/Opta):</b>	Obligatoria
	<b>Clave de la materia:</b>	413
	<b>Semestre:</b>	4
	<b>Área en plan de estudios (G, E):</b>	Ingeniería aplicada
	<b>Total de horas por semana:</b>	4
	<i>Teoría: Presencial o Virtual</i>	2
	<i>Laboratorio o Taller:</i>	2
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	0
	<b>Créditos Totales:</b>	4
	<b>Total de horas semestre (x 16 sem):</b>	64
	Fecha de actualización:	Enero 2023
<i>Prerrequisito (s):</i>	Arquitectura de computadoras I, (313)	
<i>Realizado por:</i>	Comité de Rediseño Curricular	

**DESCRIPCIÓN:**

El curso proporciona las herramientas fundamentales para que el estudiante sea capaz de desarrollar programas de bajo nivel, aplicando algunas de las operaciones básicas que los sistemas de software desarrollan para la gestión y control de los recursos computacionales. Los conocimientos y habilidades adquiridos en este curso, son antecedentes para que los estudiantes aborden en cursos posteriores que niveles abstractos de la organización de las computadoras y los sistemas operativos componentes elementales de cualquier sistema computacional.

<b>DOMINIOS</b> (Se toman de las competencias)	<b>OBJETOS DE ESTUDIO</b> (Contenidos necesarios para desarrollar cada uno de los dominios)	<b>RESULTADOS DE APRENDIZAJE</b> (Se plantean de los dominios y contenidos)	<b>METODOLOGÍA</b> (Estrategias, secuencias, recursos didácticos)	<b>EVIDENCIAS</b> (Productos tangibles que permiten valorar los resultados de aprendizaje)
<p><b>Específicas.</b></p> <p><b>Sistemas Informáticos y Computación.</b></p> <p><b>Descripción:</b></p> <p>Aplica el conocimiento, metodologías, procesos y técnicas, para el análisis, diseño, modelado y desarrollo de sistemas informáticos y de cómputo.</p>	<p><b>UNIDAD I.</b></p> <p><b>ARQUITECTURA DEL MICROPROCESADOR 8088</b></p> <p>1.1. Evaluación de la arquitectura del procesador. 1.2. Registros y memoria. 1.3. Utilización de los registros. 1.4. Organización de la memoria. 1.5. Modos de direccionamiento.</p>	<p>Identifica la interrelación que existe entre la arquitectura interna de un microprocesador y los dispositivos, externos al microprocesador, como memoria y elementos de entrada/salida, cuando se ejecutan tareas del sistema operativo o cuando se ejecutan programas desarrollados por el usuario.</p>	<p>Lección magistral.</p> <p>Estudio de casos.</p> <p>Revisión bibliográfica. Discusión dirigida. Herramientas de simulación.</p> <p>Uso de recursos como: a. presentaciones b. mapas conceptuales c. desarrollo de pseudocódigo.</p> <p>Aprendizaje basado en problemas.</p>	<ul style="list-style-type: none"> <li>● Generación de ensayos por literatura revisada.</li> <li>● Presentación informática por lectura asignada.</li> <li>● Cuadro comparativo de contribuciones a la computación.</li> <li>● Cuadro comparativo de la evolución de las arquitecturas de computadoras.</li> <li>● Reporte de prácticas.</li> </ul>

<p><b>Diseño electrónico.</b></p> <p><b>Descripción:</b></p> <p>Emplea diversas tecnologías de censado, arquitecturas de hardware y recursos informáticos en la propuesta de soluciones a problemas del entorno, con impacto en la sociedad, en el sector industrial y de servicios.</p>	<p><b>UNIDAD III.</b> <b>PROGRAMACIÓN BASE DE ENTRADA-SALIDA</b></p> <p>3.1. Servicio del DOS para dispositivo de Entrada. 3.2. Servicios del DOS para dispositivo de salida.</p>	<p>Describe, diseña e implementa programas que le permitan comprender algunas de las tareas básicas desarrolladas por sistemas operativos monousuario.</p>		<ul style="list-style-type: none"> <li>● Generación de ensayos por literatura revisada.</li> <li>● Presentación informática por lectura asignada.</li> <li>● Reporte de prácticas.</li> </ul>
<p><b>Dominio:</b></p> <p>Aplica los lenguajes de programación de alto y bajo nivel en ambientes de sistemas HW/SW.</p>	<p><b>UNIDAD IV.</b> <b>MODULARIDAD</b></p> <p>4.1. Procedimientos. 4.2. Macros. 4.3. Multimódulos.</p>	<p>Desarrolla técnicas para una programación eficiente en ensamblador.</p>		
	<p><b>UNIDAD V.</b> <b>PROGRAMACIÓN AVANZADA DE ENTRADA-SALIDA</b></p> <p>5.1. Servicios del BIOS para dispositivo de entrada. 5.2. Servicios del BIOS para dispositivo de salida. 5.3. Área de datos del BIOS para dispositivo de entrada. 5.4. Video mapeado en memoria.</p>	<p>Aplica e implementa programas de comunicación con los dispositivos de entrada-salida, utilizando servicios del BIOS y mapeo de memoria.</p>		
	<p><b>UNIDAD VI.</b> <b>TÓPICOS AVANZADOS</b></p> <p>6.1. Ensamblador y lenguajes de alto nivel. 6.2. Programación residente.</p>	<p>Desarrolla programas para comunicar eventos de bajo nivel con programas desarrollados en lenguajes de alto nivel. Así mismo, aplica la utilización de programas que los sistemas operativos y algunos controladores de hardware cargan de manera permanente en la memoria de un equipo de cómputo.</p>		

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos)
<p><i>Assembly Language for x86 processors.</i> Prentice Hall. Kip R. Irvine. (2014). <i>Assembly Programming and Computer Architecture for Software Engineers</i> Brian R. Hall and Kevin J. Slonka, CRC Press <i>Modern X86 Assembly Language Programming: 32-bit, 64-bit, SSE, and AVX.</i> Daniel Kusswurm. (2014), Apress. <i>Low Level Programming</i> Igor Zhirkov. (2017), O'Reilly Press</p> <p>Simuladores (disponibles a la fecha):</p> <p>CPU Fetch-Decode-Execute <a href="https://www.hartismere.com/20398/CPU-Fetch-Decode-Execute-Animation">https://www.hartismere.com/20398/CPU-Fetch-Decode-Execute-Animation</a></p> <p>MARIE.js - A Simple CPU Simulator Project <a href="https://marie-js.github.io/MARIE.js/about.html">https://marie-js.github.io/MARIE.js/about.html</a></p> <p>x86 like cpu simulator - 8-bit and 256 bytes of memory <a href="https://schweigi.github.io/assembler-simulator/">https://schweigi.github.io/assembler-simulator/</a></p> <p>MIPS CPU Pipelined Simulation <a href="http://bellerofonte.dii.unisi.it/index.asp">http://bellerofonte.dii.unisi.it/index.asp</a></p>	<p>El curso se evalúa en 3 momentos, las fechas se establecen por la secretaría académica:</p> <p><b>INSTRUMENTOS:</b></p> <p>Examen escrito/práctico Informes escritos Presentaciones informáticas Reportes de prácticas Proyecto de fin de curso</p> <p>Conocimientos: 40 % (aspectos teóricos) Habilidades: 45 % (análisis, argumentación, redacción, uso de tecnología, comunicación, efectiva, resolución de ejercicios con aplicación metodológica) Valores y actitudes: 15% (colaboración, orden, lenguaje apropiado, respeto, puntualidad).</p> <p><b>CRITERIOS DE DESEMPEÑO:</b></p> <p>Los informes por escrito: valoran el nivel de argumentación en relación al hecho que se quiere demostrar. Manejo de lenguaje técnico, coherencia entre párrafos y global, redacción, ortografía y presentación. Se utiliza una rúbrica para evaluación.</p> <ul style="list-style-type: none"> <li>● Exposición: presentadas en orden lógico:       <ol style="list-style-type: none"> <li>1. Introducción resaltando el objetivo a alcanzar</li> <li>2. Desarrollo temático, responder preguntas y aclarar dudas</li> <li>3. Concluir.</li> </ol> </li> <li>● Los trabajos extracurriculares</li> </ul> <p>Toda actividad complementaria al curso se podrá llevar a cabo en forma individual o por equipo según amerite el tema. Estos se reciben únicamente en tiempo y forma previamente establecidos.</p> <ul style="list-style-type: none"> <li>● Prácticas de Laboratorio</li> </ul> <p>Las actividades de práctica pueden ser de dos tipos, problemas teóricos o aquellos que pueden estudiarse utilizando alguna herramienta de simulación/diseño.</p> <ul style="list-style-type: none"> <li>● Proyecto de fin de curso</li> </ul> <p>Consolida los conocimientos, habilidades y aptitudes del curso. Motiva la creatividad, refuerza la disciplina al trabajo y la generación de productos dentro del contexto del área de la organización y arquitectura de las computadoras. La evaluación del proyecto se integra por metodología utilizada para su desarrollo, nivel de funcionalidad/operatividad/desempeño del modelo o prototipo, la calidad del reporte técnico elaborado para el mismo.</p> <p><b>ACREDITACIÓN:</b></p> <p>Se utilizan los porcentajes de 30%, 30% y 40% para la evaluación parcial I, II y III respectivamente. Los diferentes rubros que integran cada una de las evaluaciones parciales se encuentra disponible en la coordinación del programa educativo, sin embargo, cada profesor al inicio del ciclo escolar lo presenta ante el grupo.</p> <p><b>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</b></p> <p>Nota 1: El plagio no es tolerado y puede afectar seriamente la calificación.</p>

