

<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE <b>CHIHUAHUA</b></p> <p style="text-align: center;"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE: CONCEPTOS AVANZADOS DE PROGRAMACIÓN</b></p>	<b>DES:</b>	<b>INGENIERÍA</b>
	<b>Programa Educativo</b>	Ingeniería en Ciencias de la Computación
	<b>Tipo de materia (Obli/Opta):</b>	Obligatoria
	<b>Clave de la materia:</b>	CI274
	<b>Semestre:</b>	2
	<b>Área en plan de estudios (G, E):</b>	Ciencias de la Ingeniería
	<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:	Febrero 2023
	<b>CONCEPTOS AVANZADOS DE PROGRAMACIÓN</b>	<i>Prerrequisito (s):</i>
	<i>Realizado por:</i>	Comité de Rediseño Curricular
<b>PROPÓSITO DEL CURSO</b>		
<p>El curso proporciona conceptos y técnicas de programación mediante las cuales el estudiante pueda escribir programas de cómputo adecuados, utilizando los diferentes paradigmas y lenguajes, así como el uso de las diversas plataformas de desarrollo. La aplicación de los diversos lenguajes, los conocimientos y las técnicas de programación, promueven en el estudiante la habilidad para desarrollar aplicaciones avanzadas y complejas en las que entren a formar parte aspectos avanzados de concurrencia, subprograma y programación defensiva.</p>		
<b>COMPETENCIAS</b> (Tipo Y Nombre de la competencias que nutre la materia y a las que contribuye)	<b>DOMINIOS COGNITIVOS</b> (Objetos de estudio, temas y subtemas)	<b>RESULTADOS DE APRENDIZAJE.</b> (Por objeto de estudio).
El curso promueve las siguientes competencias:	<b>I. EJECUCION DE PROGRAMAS</b>	Distingue entre lenguajesde

**BÁSICAS:**

**SOLUCION DE PROBLEMAS**

- Aplica las diferentes técnicas de observación para la solución de problemas.
- Distingue los diversos tipos de sistemas

- 1.1 Interpretación Vs. Compilación
- 1.2 Ejecución nativa
- 1.3 Máquinas virtuales
- 1.4 Esquema de memoria en tiempo de ejecución: call-stack, heap, static data
- 1.5 Concepto de administración de memoria: allocating, de-allocating, garbage collection.

programación interpretados y compilados.

Identifica los ambientes y plataformas para la ejecución de programas.

Ejecuta programas escritos en plataformas interpretadas y compiladas.

<ul style="list-style-type: none"> <li>• Aplica la tecnología a la solución de problemáticas</li> <li>• Aplica el enfoque sistémico en diversos contextos</li> <li>• Crea soluciones innovadoras y utiliza formas no convencionales en la solución de problemas</li> </ul>		Escribe programas simples que aplican los conceptos de administración de memoria.
<p><b>PROFESIONALES:</b></p> <p><b>CIENCIAS FUNDAMENTALES DE LA INGENIERÍA</b></p>	<p><b>II. SUBPROGRAMAS</b></p> <p>2.1 Concepto de subprograma 2.2 Referenciamiento local, alcance y tiempo de vida de parámetros. 2.3 Sobrecarga de operadores 2.4 Bloques y anidamiento</p>	<p>Define el concepto de subprograma</p> <p>Escribe programas simples usando subprogramas</p>
<ul style="list-style-type: none"> <li>• Utiliza las matemáticas como herramientas para solución de problemas en ingeniería</li> </ul>	<p><b>III. MANEJO DE EVENTOS Y EXCEPCIONES</b></p> <p>3.1 Concepto de excepción 3.2 Concepto de evento 3.3 Programación defensiva 3.3.1 Captura de excepciones 3.3.2 Aserciones</p>	<p>Distingue entre los conceptos de excepción y evento.</p> <p>Escribe programas simples utilizando los principios de programación defensiva</p>
<p><b>ESPECIFICAS:</b></p> <p><b>FUNDAMENTOS DE CIENCIAS DE LA COMPUTACIÓN</b></p>	<p><b>IV. CONCURRENCIA.</b></p> <p>4.1 Concepto de concurrencia 4.1.1 Semáforos 4.1.2 Monitores 4.1.3 Hilos de control</p>	<p>Explica el concepto de concurrencia.</p> <p>Escribe programas simples con hilos de ejecución independientes.</p>
<ul style="list-style-type: none"> <li>• Diseña y aplica algoritmos, estructuras y representación de datos para soluciones computacionales</li> <li>• Distingue los fundamentos de un sistema computacional</li> <li>• Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas</li> <li>• Aplica los tópicos de paradigmas de programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas</li> </ul>	<p><b>V. PARADIGMAS DE PROGRAMACIONY LA EVOLUCION DE LOS LENGUAJES DE PROGRAMACION</b></p> <p>5.1 Generaciones de lenguajes de programación 5.2 Paradigma estructurado 5.3 Paradigma funcional 5.4 Paradigma lógico 5.5 Paradigma Orientado a Objetos 5.6 Combinación de paradigmas</p>	<p>Identifica los diferentes paradigmas de programación y los relaciona con su aplicación</p> <p>Explica la mezcla de paradigmas para aplicaciones robustas.</p>

OBJETO DE ESTUDIO	METODOLOGIA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS DE APRENDIZAJE.
<p>I. EJECUCIÓN DE PROGRAMAS</p> <p>II. SUBPROGRAMAS</p> <p>III. MANEJO DE EVENTOS Y EXCEPCIONES</p> <p>IV. CONCURRENCIA</p> <p>V. PARADIGMAS DE PROGRAMACIÓN Y LA EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN</p>	<p>Aprendizaje interactivo (exposición del profesor)</p> <p>Grupo de discusión.</p> <p>Auto aprendizaje (búsqueda y análisis de información)</p> <p>Inductivo</p> <ul style="list-style-type: none"> <li>• Observación</li> <li>• Comparación</li> </ul> <p>Deductivo</p> <ul style="list-style-type: none"> <li>• Aplicación</li> <li>• Comprobación</li> <li>• Demostración</li> </ul> <p>Sintético</p> <ul style="list-style-type: none"> <li>• Recapitulación</li> <li>• Definición</li> <li>• Resumen</li> <li>• Esquemas</li> <li>• Modelos matemáticos</li> <li>• Conclusión.</li> </ul> <p><b>Material de Apoyo didáctico:</b></p> <p><b>Recursos</b></p> <ul style="list-style-type: none"> <li>• Manual de Instrucción</li> <li>• Materiales gráficos: artículos, libros, diccionarios, etc.</li> <li>• Cañón</li> <li>• Pintarrones</li> <li>• Equipo de computo</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas y ejercicios.</li> <li>• Escritura de programas de cómputo.</li> <li>• Exámenes escritos.</li> </ul>

FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos)
<p>1. Concepts of Programming Languages. Robert W. Sebesta. Cualquier edición.</p>	<p><b>INSTRUMENTOS:</b></p> <ul style="list-style-type: none"> <li>• Exámenes</li> <li>• Solución de ejercicios (aplicación de conocimientos)</li> <li>• Lista de cotejo (Respeto y participación al trabajo dentro del salón de clase, interés por la asignatura)</li> </ul> <p><b>CRITERIOS DE DESEMPEÑO:</b></p> <ul style="list-style-type: none"> <li>• Los programas desarrollados deberán estar completos.</li> <li>• Las tareas se deberán entregar en tiempo y forma</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Exámenes escritos:</b> Se realizan 3 exámenes escritos durante el semestre y las fechas se establecen por la secretaría académica</li> </ul> <p>Se toma en cuenta para integrar <b>calificaciones parciales:</b> Exámenes 60% Tareas y practicas 40%</p> <p>Fecha de exámenes parciales: 1º. Parcial: 2º. Parcial: 3er Parcial:</p> <p><b>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</b></p>
--	--

Cronograma del Avance Programático

**S e m a n a s**

Unidades de aprendizaje	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I. Ejecución de programas																
II. Subprogramas																
III. Manejo de eventos y excepciones																
IV. Concurrencia																
V. Paradigmas de programación y la evolución de los lenguajes de programación																