



**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**

Clave:



Clave:

**FACULTAD DE INGENIERÍA**

**TEORÍA DE LA COMPUTACIÓN**

<b>DES:</b>	Ingeniería
<b>Programa(s) Educativo(s):</b>	Maestría en Ciencias Básicas
<b>Tipo de materia:</b>	Optativa
<b>Clave de la materia:</b>	MCBOP214
<b>Semestre:</b>	Segundo
<b>Área en plan de estudios:</b>	Específica
<b>Créditos</b>	5
<b>Total de horas por semana:</b>	5
<i>Teoría:</i>	4
<i>Práctica</i>	
<i>Taller:</i>	
<i>Laboratorio:</i>	
<i>Prácticas complementarias:</i>	
<i>Trabajo extra clase:</i>	1
<b>Total de horas semestre:</b>	80
<b>Fecha de actualización:</b>	Septiembre 2015
<b>Materia requisito:</b>	Computación Métodos Matemáticos

**PROPÓSITO DEL CURSO**

Que el alumno defina y analice el tipo de problemas que pueden o no pueden ser resueltos por una computadora bajo términos formales.

<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).
<p><b>GENÉRICAS:</b>  <b>Gestión del conocimiento</b>            Demuestra conocimientos y habilidades para la búsqueda, análisis crítico, síntesis y procesamiento de información para su transformación conocimiento con actitud ética.</p> <p><i>Identifica y articula sus necesidades de conocimiento a partir de definir problemas de información relevante.</i></p>	<p><b>I. MÁQUINAS DE TURING</b></p> <p>1.1 Máquinas de Turing deterministas            1.1.1 Definición formal de máquina de Turing            1.1.2 Computando con una máquina de Turing</p> <p>1.2 Tipos de máquinas de Turing            1.2.1 Máquina de Turing multi-cintas            1.2.2 Máquinas de Turing no deterministas            1.2.3 Máquinas de acceso aleatorio</p> <p>1.3 Simulación de Máquinas de Turing</p>	<p>Analiza la representación abstracta de los elementos de la computación como ciencia.</p> <p>Asocia el concepto de máquina de Turing como computadora abstracta.</p> <p>Analiza los distintos tipos de máquina de Turing.</p> <p>Simula el comportamiento de una máquina de Turing en un lenguaje de programación de alto nivel.</p>

<p><b>ESPECÍFICAS:</b> <b>Modelación y simulación matemática</b></p> <p>Modela sistemas dinámicos mediante simulaciones matemáticas para generar predicciones de comportamiento que contribuyen a la solución de problemas de contexto considerando distintos escenarios de forma honesta y responsable</p> <p><i>Clasifica problemas de sistemas complejos que pueden ser modelados matemáticamente.</i></p>	<p><b>2. INTRODUCCIÓN A LA TEORÍA DE LA NP-COMPLETITUD</b></p> <p>2.1 Relación entre máquinas de Turing deterministas y la clase P 2.2 Relación entre no determinismo y la clase NP 2.3 Relación entre P y NP 2.4 La clase NP-Completo 2.5 La clase NP-Difícil 2.6 Máquina de Turing universal 2.7 El problema de parada 2.8 Teorema de Cook 2.9.1 Demostración del teorema de Cook</p>	<p>Asocia la máquina de Turing y las clases de problemas en la teoría de la computación.</p> <p>Clasifica los problemas que tienen o no tienen solución.</p> <p>Clasifica algunos problemas que no tienen solución.</p> <p>Comprueba el uso del teorema de Cook en la teoría de la NP-completitud.</p> <p>Analiza los límites de la computabilidad.</p>
<p><i>Plantea métodos matemáticos y computacionales de solución de forma responsable y ética.</i></p> <p><i>Determina soluciones particulares del sistema complejo en tiempos razonables.</i></p> <p><i>Deduce comportamientos del sistema complejo por medio de simulaciones computacionales y matemáticas de forma honesta y responsable.</i> <b>Soluciones de sistemas físicos</b></p> <p>Explica el comportamiento de sistemas dinámicos físicos para caracterizar problemas gravitacionales y de medios transparentes empleando modelos matemáticos y, simulaciones</p> <p><i>Deduce el comportamiento de los sistemas físicos para su caracterización por medio de métodos y simulaciones computacionales y matemáticos.</i></p>	<p><b>3. DEMOSTRANDO LA NP-COMPLETITUD</b></p> <p>3.1 Problemas Np-Completo 3.1.1 3-Satisfactibilidad 3.1.2 Emparejamiento 3-dimensional 3.1.3 Circuito hamiltoniano 3.1.4 Partición 3.2 Técnicas para demostrar la NP-completitud 3.2.1 Restricción 3.2.2 Reemplazo local 3.2.3 Diseño de componentes</p>	<p>Describe problemas básicos dentro de la clase NP-completo.</p> <p>Utiliza las técnicas básicas para demostrar la NP-completitud de los problemas.</p>

<b>OBJETO DE ESTUDIO</b>	<b>METODOLOGÍA</b>	<b>EVIDENCIAS DE APRENDIZAJE.</b>
--------------------------	--------------------	-----------------------------------

	(Estrategias, secuencias, recursos didácticos)	
I. Máquinas de Turing	<ol style="list-style-type: none"> <li>1. Aprendizaje interactivo (exposición del profesor)</li> <li>2. Investigación de tópicos.</li> <li>3. Se divide el grupo para búsqueda y análisis de la información.</li> <li>4. Demostraciones formales.</li> </ol>	<ul style="list-style-type: none"> <li>• Pseudo-códigos</li> <li>• Programas computacionales</li> <li>• Ejercicios resueltos y demostraciones formales.</li> </ul>
II. Introducción a la teoría de la NP-Complejidad	<ol style="list-style-type: none"> <li>1. Trabajo individual. Búsqueda de información.</li> <li>2. Presentación.</li> <li>3. Demostraciones formales.</li> </ol>	
III. Demostrando la NP-Complejidad	<ol style="list-style-type: none"> <li>1. Aprendizaje interactivo.</li> <li>2. Demostraciones formales.</li> <li>3. Solución de ejercicios.</li> </ol> <p><b>Material de Apoyo didáctico:</b> <b>Recursos</b></p> <ul style="list-style-type: none"> <li>• Literatura citada en el programa del curso.</li> <li>• Materiales gráficos: artículos y libros, entre otros</li> <li>• Cañón.</li> <li>• Pizarrón, pintarrones.</li> </ul>	

FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos)
<ol style="list-style-type: none"> <li>1. Kelley, D., Aguilar, L. J., &amp; Platas, M. L. D. (1995). <i>Teoría de autómatas y lenguajes formales</i> (Vol. 22). Prentice Hall.</li> <li>2. Sipser, M. (2012). <i>Introduction to the Theory of Computation</i>. Cengage Learning..</li> <li>3. Hopcroft, J. E., Ullman, J. D., &amp; Motwani, R. (2002). <i>Introducción a la teoría de autómatas, lenguajes y computación</i>.</li> </ol>	<p><b>INSTRUMENTOS:</b></p> <p>Examen escrito Solución de problemas Programas computacionales</p> <p><b>CRITERIOS DE DESEMPEÑO:</b></p> <p>Los exámenes por escrito: valoran el nivel de argumentación en relación al hecho que se quiere demostrar. Manejo de lenguaje técnico, coherencia</p>

