



**UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA**

Clave:



Clave:

FACULTAD DE INGENIERÍA

ALGORÍTMICA

| | |
|-----------------------------------|------------------------------------|
| DES: | Ingeniería |
| Programa(s) Educativo(s): | Maestría en Ciencias Básicas |
| Tipo de materia: | Optativa |
| Clave de la materia: | MCBOP211 |
| Semestre: | Segundo |
| Área en plan de estudios: | Específica |
| Créditos | |
| Total de horas por semana: | 5 |
| <i>Teoría:</i> | 4 |
| <i>Práctica</i> | |
| <i>Taller:</i> | |
| <i>Laboratorio:</i> | |
| <i>Prácticas complementarias:</i> | |
| <i>Trabajo extra clase:</i> | 1 |
| Total de horas semestre: | 80 |
| Fecha de actualización: | Febrero 2016 |
| Materia requisito: | Computación Métodos Matemáticos |

PROPÓSITO DEL CURSO

Aprender los principios fundamentales del análisis y el diseño de los algoritmos. Conocer y analizar diferentes clases de algoritmos. Comprende los conceptos de complejidad y eficiencia de un algoritmo. Compara diferentes algoritmos y selecciona "el mejor". Manifiesta valores y actitudes, como el respeto a las ideas de otros, responsabilidad de su aprendizaje, disciplina en el aula, cooperación, criticidad y creatividad con carácter científico a través del trabajo en equipo. Comunica en forma oral y escrita sus ideas e interpretaciones, respecto a los algoritmos estudiados, así como expone juicios de valor respecto a la relación que estos guardan con su vida y el mundo que le rodea.

| COMPETENCIAS (Tipo y nombre de la competencias que nutre la materia y a las que contribuye) | DOMINIOS COGNITIVOS (Objetos de estudio, temas y subtemas) | RESULTADOS DE APRENDIZAJE. (Por objeto de estudio). |
|--|--|---|
| <p>GENÉRICAS: Gestión del conocimiento 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>ESPECÍFICAS:</p> | <p>I. PROGRAMACIÓN DINÁMICA</p> <p>1.1 Método general 1.2 Calendarización de intervalos con pesos 1.3 Cadena poligonal de mínimos cuadrados. 1.4 Multiplicación de matrices 1.5 Alineación de cadenas</p> | <p>Analiza las bases de la programación dinámica para problemas de optimización.</p> <p>Identifica problemas que pueden ser modelados mediante programación dinámica</p> <p>Determina soluciones de problemas que pueden ser resueltos con programación dinámica.</p> |

| | | |
|--|---|---|
| <p>Modelación y simulación matemática</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><i>Determina soluciones particulares del sistema complejo en tiempos razonables.</i></p> <p><i>Plantea métodos matemáticos y computacionales de solución de forma responsable y ética.</i></p> <p><i>Deduce comportamientos del sistema complejo por medio de simulaciones computacionales y matemáticas de forma honesta y responsable.</i></p> <p><i>Es sensible en la resolución de problemas en su entorno.</i></p> | | |
| <p>Soluciones de sistemas físicos</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>Deduca el comportamiento de los sistemas físicos para su caracterización por medio de métodos y simulaciones computacionales y matemáticos.</i></p> | <p>II. ALGORITMOS HEURÍSTICOS</p> <p>2.1 Problema de selección de centros 2.2 Cobertura de conjuntos 2.3 Método de asignación de precios 2.4 Programación lineal y redondeo</p> | <p>Clasifica problemas de sistemas complejos que pueden ser modelados mediante algoritmos heurísticos.</p> <p>Deduca comportamientos de sistemas por medio de simulaciones computacionales con algoritmos heurísticos.</p> <p>Deduca el comportamiento de algunos sistemas físicos para su caracterización por medio de métodos y simulaciones computacionales y matemáticos usando programación lineal</p> <p>Distingue las ventajas y desventajas al utilizar algoritmos de aproximación para encontrar solución a problemas que no tienen una solución exacta.</p> |
| | <p>III. ALGORITMOS ESTOCÁSTICOS</p> | <p>Asocia las bases fundamentales de la teoría de</p> |

| | | |
|--|--|---|
| | <p>3.1 Probabilidad básica 3.2 Algoritmo de corte mínimo 3.3 Pareja de puntos más cercana</p> | <p>probabilidad para aplicar algoritmos estocásticos.</p> <p>Analiza y diseña algoritmos de aproximación.</p> <p>Determina soluciones para problemas que involucren parejas de puntos más cercana.</p> <p>Deduce comportamientos de problemas que involucren el algoritmo de corte mínimo, como flujos en redes.</p> |
| | <p>IV. ALGORITMOS EXACTOS</p> <p>4.1 Algoritmo de retroceso (<i>Backtracking</i>) 4.1.1 El problema de las 8 reinas 4.1.2 Suma de subconjuntos 4.1.3 Coloración de grafos 4.1.3 Ciclos hamiltonianos 4.1.4 El problema de la mochila (Knapsack problem) 4.2 Búsqueda exhaustiva ingenua (<i>Naive Search</i>)</p> | <p>Compara las ventajas y desventajas de utilizar algoritmos exactos vs. algoritmos de aproximación.</p> <p>Identifica problemas particulares que tienen solución con algoritmos exactos en tiempo razonable.</p> <p>Deduce comportamientos de sistemas de su entorno que pueden tener su analogía con problemas de los algoritmos exactos vistos en clase.</p> |

| OBJETO DE ESTUDIO | METODOLOGÍA (Estrategias, secuencias, recursos didácticos) | EVIDENCIAS DE APRENDIZAJE. |
|---|---|---|
| <p>I. Programación dinámica II. Algoritmos de aproximación III. Algoritmos aleatorios IV. Algoritmos exactos</p> | <p><u>Estrategias:</u> Aprendizaje interactivo (exposición del profesor, ejercicios resueltos por el estudiante). Investigación de tópicos Demostraciones formales de problemas matemáticos. Exposiciones de temas extras, teóricos y experimentales, por parte del estudiante. Resolución de ejercicios en corrillos. Discusión grupal de tópicos especiales. Análisis de algoritmos y su implementación en computadora</p> | <ul style="list-style-type: none"> ● Pseudo-códigos ● Programas computacionales ● Ejercicios resueltos ● Demostraciones formales. |

| | | |
|--|---|--|
| | <p><u>Recursos:</u> pintarrones, proyectores, literatura citada (textos, artículos científicos), recursos de TI, software de simulación especializado (Mathematica, Matlab, Maple)</p> <p><u>Secuencias:</u> activación de aprendizajes previos, planteamiento del objetivo, introducción al tema y conceptos centrales, trabajo autónomo del estudiante, monitoreo docente, balance de actividades, objetivos de la siguiente sesión, retroalimentación.</p> | |
|--|---|--|

| FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas) | EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos) |
|--|--|
| <ol style="list-style-type: none"> 1. Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. (2006). <i>Algorithms</i>. McGraw-Hill, Inc. 2. Cormen, T. H. (2009). <i>Introduction to algorithms</i>. MIT press. 3. Kleinberg, J., & Tardos, É. (2006). <i>Algorithm design</i>. Pearson Education India. 4. Horowitz, E., & Sahni, S. <i>Fundamentals of computer algorithms</i>. (1978). Pitman, London. | <p>INSTRUMENTOS:</p> <p>Examen escrito Solución de problemas Programas computacionles</p> <p>CRITERIOS DE DESEMPEÑO:</p> <p>Los exámenes 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 autoevaluación y heteroevaluación.</p> <p>Los problemas y las demostraciones: valoran el conocimiento teórico aplicado a la resolución de un ejercicio, debe contener el procedimiento y el resultado correcto. Se utiliza lista de cotejo para autoevaluación y heteroevaluación.</p> <p>Los programas computacionales: valoran la eficiencia del código y el conocimiento teórico aplicado para la programación del problema. Se utiliza una rúbrica para autoevaluación y heteroevaluación.</p> <p>La acreditación del curso:</p> <ul style="list-style-type: none"> ● Examen intermedio: 30% ● Examen final: 40% ● Proyecto final: 30% <p>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</p> |

| | |
|--|---|
| | Nota: la calificación mínima aprobatoria es de 80. |
|--|---|

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. Programación dinámica | | | | | | | | | | | | | | | | |
| II. Algoritmos de aproximación | | | | | | | | | | | | | | | | |
| III. Algoritmos aleatorios | | | | | | | | | | | | | | | | |
| IV. Algoritmos exactos | | | | | | | | | | | | | | | | |