

|  |   |   |   |
|--|---|---|---|
| <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>UNIDAD ACADÉMICA INGENIERÍA</b></p> <p style="text-align: center;"><b>PROGRAMA DE LA UNIDAD DE APRENDIZAJE:</b></p> <p style="text-align: center;"><b>ANÁLISIS Y DISEÑO DE ALGORITMOS</b></p> | <p><b>DES:</b></p> <p><b>Programa Educativo:</b></p> <p><b>Tipo de materia (Obligatoria/Optativa):</b></p> <p><b>Clave de la materia:</b></p> <p><b>Semestre:</b></p> <p><b>Área en plan de estudios</b></p> <p><b>Total de horas por semana:</b></p> <p><i>Teoría: Presencial o Virtual</i></p> <p><i>Laboratorio o Taller:</i></p> <p><i>Prácticas:</i></p> <p><i>Trabajo extra-clase:</i></p> <p><b>Créditos Totales:</b></p> <p><b>Total de horas semestre (x 16 sem):</b></p> <p><b>Fecha de actualización:</b></p> <p><b>Prerrequisito (s):</b></p> | <p style="text-align: center;"><b>INGENIERÍA</b></p> <p>Maestría en Ingeniería en Computación</p> <p>Electiva</p> <p>MICE02</p> <p>1</p> <p>G,E</p> <p>4</p> <p>2</p> <p>0</p> <p>2</p> <p>2</p> <p>6</p> <p>64</p> <p>18-Oct-2017</p> <p>Ninguna</p>   |   |
|  | <p><i>Realizado por:</i></p> <p>Comité de Rediseño Curricular</p>   |   |   |
|  | <p><b>Propósito del curso :</b></p> <p>El curso aporta los principios fundamentales del análisis y diseño de algoritmos para comprender como se aplican en una amplia gama de problemas. Permite el estudio de diversos tipos de problemas y el diseño y/o aplicación de algoritmos para resolverlos de manera eficiente.</p>   |   |   |
|  | <b>COMPETENCIAS</b>   | <b>DOMINIOS COGNITIVOS</b>  | <b>RESULTADOS DE APRENDIZAJE</b>  |
|  | <p><b>Aplicación de Ciencias de la Computación:</b> <i>Aplica con ética las ciencias computacionales para resolver problemas en los sectores: industrial, gubernamental, académico y social bajo esquemas de colaboración inter, multi y transdisciplinaria</i></p> <p>1. Soluciona problemas en diversas áreas del conocimiento aplicando las ciencias</p>   | <p><b>1. Introducción al Diseño y Análisis de Algoritmos.</b></p> <p>1.1. Problemas, algoritmos y complejidad computacional</p> <p>1.2. Definiciones de eficiencia.</p> <p>1.1.1. Tiempo de ejecución en el peor caso.</p> <p>1.1.2. Tiempo polinómico.</p> <p>1.3. Orden de crecimiento asintótico.</p> <p>1.3.1. Notación O grande</p> <p>1.3.2. Notación <math>\Omega</math> grande</p> <p>1.3.3. Notación <math>\Theta</math></p> <p>1.4. Ejemplos de tiempos comunes de ejecución.</p> <p>1.4.1. Tiempo lineal.</p> <p>1.4.2. Tiempo <math>O(n \log n)</math>.</p> | <p>Aplica los fundamentos de ingeniería y de ingeniería especializada en la identificación, formulación, análisis y resolución de problemas complejos con el fin de alcanzar conclusiones fundamentadas.</p> <p>Realiza investigaciones de problemas complejos por métodos que incluyen experimentos apropiados, análisis e interpretación de datos y síntesis de la información con el fin de llegar a</p> |

|                         |  |                              |
|-------------------------|--|------------------------------|
| <p>computacionales.</p> | <ul style="list-style-type: none"> <li>1.4.3. Tiempo cuadrático.</li> <li>1.4.4. Tiempo cúbico.</li> <li>1.4.5. Tiempo <math>O(n^k)</math></li> <li>1.4.6. Más allá del tiempo polinómico.</li> <li>1.4.7. Tiempo sublineal.</li> <li>1.5. Tipos de problemas según su complejidad <ul style="list-style-type: none"> <li>1.5.1. Maquinas de Turing determinísticas y la Clase P</li> <li>1.5.2. Problemas No Determinísticos y la clase NP</li> <li>1.5.3. <math>P=NP?</math></li> <li>1.5.4. Transformaciones polinomiales y los problemas NP-Completos</li> <li>1.5.5. Problemas NP-Hard</li> </ul> </li> <li><b>2. Algoritmos “Divide y Venceras” (Divide and Conquer)</b> <ul style="list-style-type: none"> <li>2.1. Relaciones de Recurrencia</li> <li>2.2. Métodos para resolver recurrencias <ul style="list-style-type: none"> <li>2.2.1. Árboles de Recursiones</li> <li>2.2.2. El método Maestro</li> </ul> </li> <li>2.3. El modelo “Divide y Venceras”</li> <li>2.4. Algoritmo MergeSort</li> <li>2.5. Algoritmo para encontrar el par de puntos mas cercano</li> <li>2.6. Algoritmo de la multiplicación</li> </ul> </li> <li><b>3. Teoria de Grafos</b> <ul style="list-style-type: none"> <li>3.1. Definiciones básicas y aplicaciones. <ul style="list-style-type: none"> <li>3.1.1. Representacion de Grafos <ul style="list-style-type: none"> <li>3.1.1.1. Matriz de adyacencia</li> <li>3.1.1.2. Listas adyacentes</li> </ul> </li> <li>3.1.2. Grafos No Dirigidos</li> <li>3.1.3. Caminos</li> <li>3.1.4. Ciclos</li> <li>3.1.5. Árboles</li> </ul> </li> <li>3.2. Conectividad</li> <li>3.3. Búsqueda en Anchura (BFS)</li> <li>3.4. Búsqueda en Profundidad (DFS)</li> <li>3.5. Grafos Bipartitas (provando la característica de Bipartición)</li> <li>3.6. Grafos Aciclicos Dirigidos (DAG's) y el Orden Topológico</li> </ul> </li> <li><b>4. Algoritmos Ambiciosos (Greedy)</b> <ul style="list-style-type: none"> <li>4.1. La estrategia “Greedy”</li> <li>4.2. Árboles de Mínima Expansión</li> <li>4.3. Códigos de Huffman</li> <li>4.4. Calendarización de Intervalos</li> <li>4.5. Caminos mas cortos en grafos (Algoritmo de Dijkstra)</li> <li>4.6. Algoritmo de Kruskal</li> <li>4.7. Closterizaciones (Clustering)</li> </ul> </li> </ul> | <p>conclusiones válidas.</p> |
|-------------------------|--|------------------------------|

|  |  |  |
|--|--|--|
|  | <p><b>5. Programación Dinámica.</b></p> <p>5.1. Principios de programación dinámica.</p> <p>5.1.1. Memorización</p> <p>5.1.2. Iteración sobre los sub-problemas.</p> <p>5.2. Calendarización de Intervalos Ponderados</p> <p>5.3. Método de Mínimos cuadrados segmentados.</p> <p>5.4. El problema de la subsecuencia mas larga.</p> <p>5.5. Revisitando el problema de los caminos mas cortos en grafos.</p> <p>5.6. El problema de la mochila (Knapsack)</p> <p>5.7. Alineación de secuencias.</p> <p><b>6. Flujos en Redes</b></p> <p>6.1. Flujos máximos y cortes mínimos en una red.</p> <p>6.2. El problema de flujo máximo y el algoritmo de Ford-Fulkerson.</p> <p>6.2.1. Seleccionando buenos caminos de aumentación.</p> <p>6.3. El problema del Apareamiento Bipartita (Bipartite Matching).</p> <p>6.4. Caminos desunidos en grafos dirigidos y no dirigidos.</p> <p><b>7. Algoritmos de Aproximación</b></p> <p>7.1. El problema de selección de centros.</p> <p>7.2. El problema de la cobertura de conjuntos.</p> <p>7.3. Método de asignación de precios: Cobertura de vértices.</p> <p>7.4. Programación lineal.</p> <p>7.4.1. El método Simplex</p> <p>7.5. El problema del agente viajero</p> |  |
|--|--|--|

| OBJETO DE APRENDIZAJE  | METODOLOGIA  | EVIDENCIAS DE APRENDIZAJE  |
|--|--|--|
| <ol style="list-style-type: none"> <li>1. Introducción al Diseño y Análisis de Algoritmos.</li> <li>2. Grafos.</li> <li>3. Algoritmos voraces.</li> <li>4. Algoritmos del tipo "Divide y Vencerás".</li> <li>5. Examen parcial.</li> <li>6. Técnicas de programación dinámica.</li> <li>7. Redes de Flujos.</li> <li>8. Algoritmos de aproximación.</li> <li>9. Búsqueda local</li> <li>10. Examen final.</li> </ol> | <ol style="list-style-type: none"> <li>1. Para cada unidad, se presenta una introducción por parte del maestro.</li> <li>2. Para cada unidad, el maestro deja una tarea donde se aplican los conceptos vistos en clase para la resolución de problemas. La tarea requiere que el alumno revise las técnicas y concepto vistos en clase, aclare dudas y aplique las técnicas ya sea manualmente o las implemente utilizando un lenguaje de programación.</li> <li>3. La discusión y el análisis se propician a partir del planteamiento de una situación problemática, donde el estudiante aporte alternativas de solución o resolver un ejercicio en el que aplique conceptos ya analizados.</li> <li>4. En algunas unidades el maestro muestra directamente en una computadora, posiblemente con la ayuda de un proyector, cómo se implementan las técnicas vistas en clase usando un lenguaje</li> </ol> | <ul style="list-style-type: none"> <li>• Tareas escritas</li> <li>• Reportes de investigación</li> <li>• Evaluaciones parciales</li> <li>• Trabajo final integrador</li> </ul> |

|  |  |  |
|--|--|--|
|  | de programación.<br><br>Material de Apoyo didáctico: Recursos <ul style="list-style-type: none"> <li>• Talleres para realizar ejercicios</li> <li>• Materiales gráficos: artículos, libros, diccionarios, etc.</li> <li>• Cañón</li> <li>• Rotafolio</li> <li>• Pizarrón, pintarrones</li> <li>• Proyector de acetatos</li> </ul> * Plataforma |  |
|--|--|--|

| FUENTES DE INFORMACIÓN   | EVALUACIÓN DE LOS APRENDIZAJES   |
|--|--|
| 1. Algorithm Design Jon Kleinberg & Eva Tardos.<br>2. Computers and Intractability: A Guide to the Theory on NP-Completeness<br>3. Algorithms S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani.<br>4. Introduction to Algorithms Cormen, Leiserson, Rivest, & Stein. | <ul style="list-style-type: none"> <li>• Tareas escritas 20%</li> <li>• Reportes de investigación 20%</li> <li>• Evaluaciones parciales 20%</li> <li>• Trabajo final integrador 40%</li> </ul> |

### Cronograma del Avance Programático

| UNIDADES DE APRENDIZAJE                            | SEMANAS |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
|--|---------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|--|
|  | 1       | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |  |
| 1. Introducción al Diseño y Análisis de Algoritmos |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 2. Algoritmos "Divide y Venceras"                  |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 3. Teoría de Grafos                                |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 4. Algoritmos Ambiciosos (Greedy)                  |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 5. Programación Dinámica                           |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 6. Flujos en Redes                                 |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |
| 7. Algoritmos de Aproximación                      |         |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |  |