



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

Clave: 08MSU0017H

**FACULTAD DE INGENIERÍA**

Clave: 08USU4053W

**PROGRAMA DEL CURSO:**

**Análisis y Diseño de Algoritmos  
(DI 615)**

<b>DES:</b>	Ingeniería
<b>Programa Educativo:</b>	Doctorado
<b>Tipo de materia (Obligatoria/Optativa):</b>	Optativa
<b>Clave de la materia:</b>	DI615
<b>Semestre:</b>	
<b>Área en plan de estudios</b>	Inteligencia Computacional
<b>Créditos</b>	6
<b>Total de horas por semana:</b>	6
<i>Teoría: Presencial o Virtual</i>	4
<i>Laboratorio o Taller:</i>	0
<i>Prácticas:</i>	2
<i>Trabajo extra-clase:</i>	0
<b>Créditos Totales:</b>	6
<b>Total de horas semestre (x 16 sem):</b>	96
<b>Fecha de actualización:</b>	Octubre de 2017
<b>Prerrequisito (s):</b>	Ninguna

**Propósito del curso :**

El curso aporta los principios fundamentales del diseño de algoritmos para comprender como surgen en una amplia gama de aplicaciones, cómo utilizar e implementar varias técnicas de diseño de algoritmos y cómo los algoritmos resuelven de manera eficiente estos problemas.

COMPETENCIAS	DOMINIOS COGNITIVOS	RESULTADOS DE APRENDIZAJE
<p><b>CE1:Fundamentos Avanzados para Investigación en Ingeniería:</b> Desarrolla e implementa métodos, modelos, simulaciones, teorías y herramientas tecnológicas como fundamentos para la innovación y propuesta a la solución del amplio rango de problemas que resuelve la ingeniería, especialmente en lo referente a la optimización del diseño, la operación, el control y la cuantificación de la incertidumbre para la toma de decisiones dentro del ejercicio profesional y de investigación en el campo disciplinar específico.</p>	<ol style="list-style-type: none"> <li><b>1. Introducción al Diseño y Análisis de Algoritmos.</b> <ol style="list-style-type: none"> <li>1.1. Definiciones de eficiencia.               <ol style="list-style-type: none"> <li>1.1.1. Tiempo de ejecución en el peor caso.</li> <li>1.1.2. Tiempo polinómico.</li> </ol> </li> <li>1.2. Orden de crecimiento asintótico.</li> <li>1.3. Ejemplos de tiempos comunes de ejecución.               <ol style="list-style-type: none"> <li>1.3.1. Tiempo lineal.</li> <li>1.3.2. Tiempo <math>O(n \log n)</math>.</li> <li>1.3.3. Tiempo cuadrático.</li> <li>1.3.4. Tiempo cúbico.</li> <li>1.3.5. Tiempo <math>O(n^k)</math></li> <li>1.3.6. Más allá del tiempo polinómico.</li> <li>1.3.7. Tiempo sublineal.</li> </ol> </li> </ol> </li> <li><b>2. Grafos</b> <ol style="list-style-type: none"> <li>2.1. Definición y aplicaciones de los grafos.</li> <li>2.2. Conectividad y navegación de los grafos.</li> <li>2.3. Implementando navegación de grafos usando colas y pilas.</li> <li>2.4. Algoritmo para determinar si un grafo es bipartita.</li> <li>2.5. Conectividad en los grafos dirigidos.</li> <li>2.6. Grafos dirigidos sin ciclos y ordenamiento topológico.</li> </ol> </li> <li><b>3. Algoritmos voraces</b> <ol style="list-style-type: none"> <li>3.1. Calendarización de intervalos.</li> <li>3.2. Caches óptimos.</li> </ol> </li> </ol>	<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 conclusiones válidas.</p>

	<ul style="list-style-type: none"> <li>3.3. Caminos más cortos en un grafo.</li> <li>3.4. El problema del árbol más pequeño.</li> <li>3.5. El algoritmo de Kruskal.</li> <li>3.6. Clustering.</li> <li>3.7. Codificación de Huffman y Compresión de datos.</li>   <li><b>4. Algoritmos del tipo "Divide y Vencerás"</b> <ul style="list-style-type: none"> <li>4.1. El algoritmo de MergeSort.</li> <li>4.2. Relaciones de recurrencia.</li> <li>4.3. Conteo de acciones de inversión.</li> <li>4.4. Encontrando el par de puntos más cercano.</li> <li>4.5. Multiplicación de enteros.</li> </ul> </li>   <li><b>5. Técnicas de programación dinámica.</b> <ul style="list-style-type: none"> <li>5.1. Calendarización con intervalos ponderados.</li> <li>5.2. Principios de programación dinámica. <ul style="list-style-type: none"> <li>5.2.1. Memorización</li> <li>5.2.2. Iteración sobre los sub-problemas.</li> </ul> </li> <li>5.3. Cálculo de mínimos cuadrados.</li> <li>5.4. Subconjuntos de sumas.</li> <li>5.5. Alineación de secuencias.</li> </ul> </li>   <li><b>6. Redes de Flujos</b> <ul style="list-style-type: none"> <li>6.1. El problema de flujo máximo y el algoritmo de Ford-Fulkerson.</li> <li>6.2. Flujos máximos y cortes mínimos en una red.</li> <li>6.3. Seleccionando buenos caminos de aumentación.</li> <li>6.4. Solución del problema de apareamiento máximo bipartito.</li> <li>6.5. Caminos desunidos en grafos dirigidos y no dirigidos.</li> </ul> </li>   <li><b>7. Algoritmos de Aproximación</b> <ul style="list-style-type: none"> <li>7.1. Problema de selección de centros.</li> <li>7.2. Cobertura de conjuntos: Heurística voraz.</li> <li>7.3. Método de asignación de precios: Cobertura de vértices.</li> <li>7.4. Programación lineal y redondeo.</li> </ul> </li>   <li><b>8. Búsqueda local</b> <ul style="list-style-type: none"> <li>8.1. Espacio de búsqueda de los problemas de optimización.</li> <li>8.2. El algoritmo de la metrópolis y el recocido simulado.</li> <li>8.3. Redes neuronales de Hopfield usando búsqueda local.</li> <li>8.4. Aproximación del Corte Máximo.</li> </ul> </li> </ul>	
--	--	--

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 de programación.</li> </ol> <p>Material de Apoyo didáctico: Recursos</p> <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> <p>* Plataforma</p>	<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>
--	--	--

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
Algorithm Design Jon Kleinberg & Eva Tardos. Algorithms S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. 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. Grafos																	
3. Algoritmos voraces																	
4. Algoritmos del tipo "Divide y Vencerás"																	
5. Examen Parcial																	
6. Técnicas de programación dinámica																	
7. Redes de Flujos																	
8. Algoritmos de Aproximación																	
9. Búsqueda local																	
10. Examen final																	