



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

Clave: 08MSU0017H

FACULTAD DE INGENIERÍA

Clave: 08USU4053W

PROGRAMA DEL CURSO:

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

DES:	Ingeniería
Programa Educativo:	Doctorado
Tipo de materia (Obligatoria/Optativa):	Optativa
Clave de la materia:	DI615
Semestre:	
Área en plan de estudios	Inteligencia Computacional
Créditos	6
Total de horas por semana:	6
<i>Teoría: Presencial o Virtual</i>	4
<i>Laboratorio o Taller:</i>	0
<i>Prácticas:</i>	2
<i>Trabajo extra-clase:</i>	0
Créditos Totales:	6
Total de horas semestre (x 16 sem):	96
Fecha de actualización:	Octubre de 2017
Prerrequisito (s):	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>CE1:Fundamentos Avanzados para Investigación en Ingeniería: 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"> 1. Introducción al Diseño y Análisis de Algoritmos. <ol style="list-style-type: none"> Definiciones de eficiencia. <ol style="list-style-type: none"> Tiempo de ejecución en el peor caso. Tiempo polinómico. Orden de crecimiento asintótico. Ejemplos de tiempos comunes de ejecución. <ol style="list-style-type: none"> Tiempo lineal. Tiempo $O(n \log n)$. Tiempo cuadrático. Tiempo cúbico. Tiempo $O(n^k)$. Más allá del tiempo polinómico. Tiempo sublineal. 2. Grafos <ol style="list-style-type: none"> Definición y aplicaciones de los grafos. Conectividad y navegación de los grafos. Implementando navegación de grafos usando colas y pilas. Algoritmo para determinar si un grafo es bipartita. Conectividad en los grafos dirigidos. Grafos dirigidos sin ciclos y ordenamiento topológico. 3. Algoritmos voraces <ol style="list-style-type: none"> Calendarización de intervalos. Caches óptimos. 	<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"> 3.3. Caminos más cortos en un grafo. 3.4. El problema del árbol más pequeño. 3.5. El algoritmo de Kruskal. 3.6. Clustering. 3.7. Codificación de Huffman y Compresión de datos. <p>4. Algoritmos del tipo "Divide y Vencerás"</p> <ul style="list-style-type: none"> 4.1. El algoritmo de MergeSort. 4.2. Relaciones de recurrencia. 4.3. Conteo de acciones de inversión. 4.4. Encontrando el par de puntos más cercano. 4.5. Multiplicación de enteros. <p>5. Técnicas de programación dinámica.</p> <ul style="list-style-type: none"> 5.1. Calendarización con intervalos ponderados. 5.2. Principios de programación dinámica. <ul style="list-style-type: none"> 5.2.1. Memorización 5.2.2. Iteración sobre los sub-problemas. 5.3. Cálculo de mínimos cuadrados. 5.4. Subconjuntos de sumas. 5.5. Alineación de secuencias. <p>6. Redes de Flujos</p> <ul style="list-style-type: none"> 6.1. El problema de flujo máximo y el algoritmo de Ford-Fulkerson. 6.2. Flujos máximos y cortes mínimos en una red. 6.3. Seleccionando buenos caminos de aumentación. 6.4. Solución del problema de apareamiento máximo bipartito. 6.5. Caminos desunidos en grafos dirigidos y no dirigidos. <p>7. Algoritmos de Aproximación</p> <ul style="list-style-type: none"> 7.1. Problema de selección de centros. 7.2. Cobertura de conjuntos: Heurística voraz. 7.3. Método de asignación de precios: Cobertura de vértices. 7.4. Programación lineal y redondeo. <p>8. Búsqueda local</p> <ul style="list-style-type: none"> 8.1. Espacio de búsqueda de los problemas de optimización. 8.2. El algoritmo de la metrópolis y el recocido simulado. 8.3. Redes neuronales de Hopfield usando búsqueda local. 8.4. Aproximación del Corte Máximo. 	
--	---	--

OBJETO DE APRENDIZAJE	METODOLOGIA	EVIDENCIAS DE APRENDIZAJE
-----------------------	-------------	---------------------------

<ol style="list-style-type: none"> 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. 	<ol style="list-style-type: none"> 1. Para cada unidad, se presenta una introducción por parte del maestro. 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. 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. 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. <p>Material de Apoyo didáctico: Recursos</p> <ul style="list-style-type: none"> • Talleres para realizar ejercicios • Materiales gráficos: artículos, libros, diccionarios, etc. • Cañón • Rotafolio • Pizarrón, pintarrones • Proyector de acetatos <p>* Plataforma</p>	<ul style="list-style-type: none"> • Tareas escritas • Reportes de investigación • Evaluaciones parciales • Trabajo final integrador
--	--	--

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"> • Tareas escritas 20% • Reportes de investigación 20% • Evaluaciones parciales 20% • Trabajo final integrador 40%

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																	