

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA  UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA FACULTAD DE INGENIERIA PROGRAMA DE LA UNIDAD DE APRENDIZAJE: <u>ANÁLISIS Y DISEÑO DE ALGORITMOS</u>	DES:	Ingeniería
	Programa académico	Maestría en Ingeniería en Computación
	Tipo de materia (Obli/Opta):	Electiva
	Clave de la materia:	MICE2302
	Semestre:	1
	Área en plan de estudios (G y E):	G, E
	Total de horas por semana:	6
	Teoría: Presencial o Virtual	2
	Laboratorio o Taller:	0
	Prácticas:	2
	Trabajo extra-clase:	2
	Créditos Totales:	6
	Total de horas semestre (x 16 sem):	64
	Fecha de actualización:	Agosto 2023
	Prerrequisito (s):	Ninguno
DESCRIPCIÓN DEL CURSO: 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.		
COMPETENCIAS A DESARROLLAR: INTELIGENCIA ARTIFICIAL. Aplica la Inteligencia Artificial para resolver problemas en los sectores industrial, gubernamental, académico y social bajo esquemas de colaboración ética y multidisciplinaria. GESTIÓN DE PROYECTOS. Coordina y administra de forma responsable, proyectos que atienden criterios de sustentabilidad para contribuir a la mejora de la calidad de vida. GESTIÓN DEL CONOCIMIENTO. Demuestra conocimientos y habilidades para la búsqueda, análisis crítico, síntesis y procesamiento de información para su transformación en conocimiento, con actitud ética. COMUNICACIÓN CIENTÍFICA. Difunde con responsabilidad ética y social el conocimiento científico, tecnológico, artístico y/o humanístico que produce de forma objetiva para aportar ideas y hallazgos científicos. INVESTIGACIÓN. Desarrolla investigación original, tecnología y/o innovaciones en procesos, servicios o productos que contribuyan a la solución de problemas, mejoren la convivencia, generen oportunidades para el desarrollo sustentable y propicien una mejor calidad de vida.		

DOMINIOS	OBJETOS DE ESTUDIO	RESULTADOS DE APRENDIZAJE	METODOLOGÍA	EVIDENCIAS
Soluciona problemas en diversas áreas del conocimiento aplicando las ciencias computacionales Comprueba los resultados obtenidos de un prototipo contra las investigaciones recientes a fin de identificar nuevas contribuciones	1. Introducción al Diseño y Análisis de Algoritmos. 1.1. Problemas, algoritmos y complejidad computacional 1.2. Definiciones de eficiencia. 1.1.1. Tiempo de ejecución en el peor caso. 1.1.2. Tiempo	• Aplica los fundamentos de ingeniería y de ingeniería especializada en la identificación, formulación, análisis y resolución de problemas	• Para cada unidad, se presenta una introducción por parte del maestro. • Para cada unidad, el maestro deja una tarea donde se aplican los conceptos vistos en clase para la resolución de problemas. La tarea	• Reportes de investigación • Proyectos

Accede a diferentes fuentes de información (journals o revistas científicas, bases de datos, índices, etc.) de calidad. Analiza y recupera información pertinente mediante diversas estrategias de búsqueda de datos científicos. Evalúa de manera crítica la información, considerando su calidad y pertinencia. Se comunica en forma oral y escrita con propiedad, relevancia, oportunidad y ética. Aplica los elementos fundamentales de la redacción científica. Interpreta y expresa ideas, sentimientos, teorías y corrientes de pensamiento con un enfoque ecuménico (universal). Muestra un desempeño abierto, sencillo, tolerante, congruente y objetivo al comunicar el saber científico Aplica procesos metodológicos para el desarrollo de investigación o intervención, en congruencia con el planteamiento y objetivos del proyecto a abordar. Muestra habilidad para la observación del fenómeno u objeto de estudio en su campo atencional	polinómico. 1.3. Orden de crecimiento asintótico. 1.3.1. Notación O grande 1.3.2. Notación Ω grande 1.3.3. Notación θ 1.4. Ejemplos de tiempos comunes de ejecución. 1.4.1. Tiempo lineal. 1.4.2. Tiempo $O(n \log n)$. 1.4.3. Tiempo cuadrático. 1.4.4. Tiempo cúbico. 1.4.5. Tiempo $O(n^k)$ 1.4.6. Más allá del tiempo polinómico. 1.4.7. Tiempo sublineal. 1.5. Tipos de problemas según su complejidad 1.5.1. Máquinas de Turing determinísticas y la Clase P 1.5.2. Problemas No Determinísticos y la clase NP 1.5.3. ¿P=NP? 1.5.4. Transformaciones polinomiales y los problemas NP-Completo 1.5.5. Problemas NP-Hard 2. Algoritmos "Divide y Vencerás" (Divide and Conquer) 2.1. Relaciones de Recurrencia 2.2. Métodos para resolver recurrencias 2.2.1. Árboles de Recursiones 2.2.2. El método Maestro 2.3. El modelo "Divide y Vencerás" 2.4. Algoritmo MergeSort 2.5. Algoritmo para encontrar el par de puntos mas cercano 2.6. Algoritmo de la multiplicación 3. Teoría de Grafos 3.1. Definiciones básicas y aplicaciones. 3.1.1. Representación de Grafos 3.1.1.1. Matriz de adyacencia	complejos con el fin de alcanzar conclusiones fundamentadas. 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.	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. - 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. - 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. - Material de Apoyo didáctico: Recursos - Talleres para realizar ejercicios - Materiales gráficos: artículos, libros, diccionarios, etc. - Cañón - Rotafolio - Pizarrón, pintarrones - Proyector de acetatos * Plataforma	
---	---	---	---	--

	3.1.1.2. Listas adyacentes 3.1.2. Grafos No Dirigidos 3.1.3. Caminos 3.1.4. Ciclos 3.1.5. Árboles 3.2. Conectividad 3.3. Búsqueda en Anchura (BFS) 3.4. Búsqueda en Profundidad (DFS) 3.5. Grafos Bipartitas (probando la característica de Bipartición) 3.6. Grafos Acíclicos Dirigidos (DAG's) y el Orden Topológico 4. Algoritmos Ambiciosos (Greedy) 4.1. La estrategia "Greedy" 4.2. Árboles de Mínima Expansión 4.3. Códigos de Huffman 4.4. Calendarización de Intervalos 4.5. Caminos más cortos en grafos (Algoritmo de Dijkstra) 4.6. Algoritmo de Kruskal 4.7. Clustering 5. Programación Dinámica. 5.1. Principios de programación dinámica. 5.1.1. Memorización 5.1.2. Iteración sobre los subproblemas. 5.2. Calendarización de Intervalos Ponderados 5.3. Método de Mínimos cuadrados segmentados. 5.4. El problema de la subsecuencia más larga. 5.5. Revisitando el problema de los caminos más cortos en grafos. 5.6. El problema de la mochila (Knapsack) 5.7. Alineación de secuencias. 6. Flujos en Redes 6.1. Flujos máximos y cortes mínimos en una red.			
--	--	--	--	--

	6.2. El problema de flujo máximo y el algoritmo de Ford-Fulkerson. 6.2.1. Seleccionando buenos caminos de aumentación. 6.3. El problema del Apareamiento Bipartita (Bipartite Matching). 6.4. Caminos desunidos en grafos dirigidos y no dirigidos. 7. Algoritmos de Aproximación 7.1. El problema de selección de centros. 7.2. El problema de la cobertura de conjuntos. 7.3. Método de asignación de precios: Cobertura de vértices. 7.4. Programación lineal. 7.4.1. El método Simplex El problema del agente viajero			
--	--	--	--	--

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
1. Algorithm Design Jon Kleinberg & Eva Tardos. Pearson 2021.	Tareas escritas 25%
2. Computers and Intractability: A Guide to the Theory on NP-Completeness. 1979	Reportes de investigación 25%
3. Algorithms S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. 2006.	Proyectos 50%
4. Introduction to Algorithms Cormen, Leiserson, Rivest, & Stein. 4ta edicion. 2022.	

CRONOGRAMA

[illegible]