


<p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIDAD ACADÉMICA: FACULTAD DE INGENIERÍA</p> <p style="text-align: center;"><u>PROGRAMACIÓN Y ANÁLISIS DE ALGORITMOS</u></p>	DES:	INGENIERÍA
	Programa académico	Ingeniería en Computación
	Tipo de materia (Obli/Opta):	Obligatoria
	Clave de la materia:	CO303
	Semestre:	Tercero
	Área en plan de estudios:	Específica
	Total de horas por semana:	5
	<i>Teoría: Presencial o Virtual</i>	0
	<i>Laboratorio o Taller:</i>	4
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	1
	Créditos Totales:	5
	Total de horas semestre (x sem):	80
	Fecha de actualización:	Octubre 2024
<i>Prerrequisito (s):</i>	N/A	

DESCRIPCIÓN:

La materia enseña al alumno los fundamentos de programación estructurada y algoritmos, desde tipos de datos hasta complejidad computacional. Los estudiantes aprenden y aplican algoritmos clásicos como Divide y Vencerás y Programación Dinámica para resolver problemas prácticos como la optimización de la mochila y determinación de caminos más cortos. Evalúan la eficiencia de los algoritmos y desarrollan habilidades críticas para seleccionar y aplicar las soluciones más adecuadas. La materia prepara a los estudiantes para abordar desafíos computacionales complejos, brindándoles herramientas para desarrollar soluciones óptimas en diversos contextos de programación y análisis algorítmico.

COMPETENCIAS PARA DESARROLLAR:

B4. Transformación Digital

Transforma la cultura digital en la sociedad, en las organizaciones e instituciones educativas para aprovechar al máximo el potencial de las tecnologías y herramientas digitales; propiciar su uso responsable y ético que estimule la creatividad, innovación, la comunicación efectiva y el trabajo colaborativo e interdisciplinar en la solución de problemas de la sociedad digital; promoviendo la privacidad y la seguridad, así como el respeto a los derechos de autor y la propiedad intelectual.

P1. CIENCIAS E INGENIERÍA. Aplica los conocimientos y metodologías para el planteamiento y resolución de problemas complejos de las ciencias naturales y de la ingeniería, para la toma de decisiones en un contexto de responsabilidad social y del medio ambiente.

E1. DISEÑO Y DESARROLLO DE SOFTWARE. Utilizar en el diseño y desarrollo de software, integrando algoritmos avanzados y estructuras de datos para crear soluciones de software robustas y de calidad. Implica una comprensión profunda de los principios de programación, un enfoque metódico para la solución de problemas y la capacidad de adaptar y mejorar continuamente las prácticas de desarrollo para satisfacer las cambiantes necesidades tecnológicas y las demandas de los diversos sectores.

DOMINIOS	OBJETOS DE ESTUDIO	RESULTADOS DE APRENDIZAJE	METODOLOGÍA	EVIDENCIAS
<p>P1. CIENCIAS E INGENIERÍA.</p> <p>3.Utiliza conceptos, métodos y leyes fundamentales de las ciencias básicas para dar soluciones a problemas complejos de ciencias e ingeniería analizando los resultados para emitir conclusiones en situaciones reales.</p> <p>E1. DISEÑO Y DESARROLLO DE SOFTWARE.</p> <p>1.Desarrollar código eficiente, aplicando buenas prácticas de programación y aprovechando las características avanzadas del lenguaje, adquiriendo conocimientos sólidos en la programación utilizando lenguajes y paradigmas</p>	<p><u>UNIDAD I. INTRODUCCIÓN A LA PROGRAMACIÓN ESTRUCTURADA</u></p> <p>1.1 Características de los lenguajes de programación.</p> <p>1.1.1 Definición de lenguaje de programación.</p> <p>1.1.2 Definición de programa</p> <p>1.1.3 Traductores y compiladores</p> <p>1.1.4 Tipos de código</p> <p>1.1.5 Entornos de desarrollo integrado.</p> <p>1.2 Fundamentos de programación.</p> <p>1.2.1 Tipos de datos primitivos.</p> <p>1.2.2 Palabras reservadas.</p> <p>1.2.2 Variables y constantes.</p> <p>1.2.3 Expresiones.</p> <p>1.2.4 Jerarquía de operadores.</p> <p>1.2.5. Operadores aritméticos y operadores relacionales.</p> <p>1.2.6 Operadores lógicos.</p> <p>1.3 Control de flujo.</p> <p>1.3.1 Decisiones.</p> <p>1.3.2 Selecciones.</p> <p>1.3.3 Ciclos e iteraciones.</p> <p>1.3 Arreglos</p> <p>1.3.1. Arreglos unidimensionales</p> <p>.</p> <p>1.3.2 Arreglos multidimensionales</p> <p>.</p>	<p>Adquiere conocimientos fundamentales sobre los lenguajes de programación y los conceptos básicos de la programación y aprende sobre los fundamentos de la programación estructurada.</p>	<ul style="list-style-type: none"> ● Clase interactiva a Maestro - Alumno. ● Recursos tecnológicos institucionales . ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

<p>de programación</p> <p>B4.2 Utiliza de forma responsable las tecnologías de la información, comunicación, conocimiento y aprendizaje (TICCA), en el proceso de construcción de saberes y el desarrollo de proyectos sociales innovadores en el ámbito digital.</p>	<p>1.4 Funciones.</p> <p>1.4.1 Concepto de reutilización del código.</p>			
---	---	--	--	--

<p>relevantes para la industria. Rasgos Excelencia Capacidades y habilidades profesionales para la resolución de problemas.</p>	<p>1.4.2 Aplicación práctica de las funciones.</p>			
	<p><u>UNIDAD II.</u> <u>INTRODUCCIÓN AL ANÁLISIS DE ALGORITMOS</u></p> <p>2.1 Complejidad Computacional. 2.1.1 Definición de complejidad. 2.1.2 Complejidad temporal. 2.1.3 Complejidad espacial. 2.1.4 Problemas P y NP.</p> <p>2.2 Notación asintótica. 2.2.1 Notación Θ grande (Big-Θ). 2.2.2 Notación O grande (Big-O). 2.2.3 Notación Ω grande (Big-Ω).</p>	<p>Comprende y aplica conceptos clave relacionados con la eficiencia de los algoritmos, la importancia de la medición en la evaluación de la eficiencia algorítmica, y estiman los recursos temporales de los algoritmos y aplican conceptos como el análisis asintótico de funciones para expresar los tiempos de ejecución de los algoritmos, lo que les permite evaluar y comparar diferentes enfoques algorítmicos con una comprensión más profunda y crítica.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

	<p>UNIDAD III. TIPOS DE ALGORITMOS.</p> <p>3.1 Algoritmos divide y vencerás.</p> <p>3.1.3 Algoritmo Merge- Sort.</p> <p>3.1.4 Quick-Sort.</p> <p>3.1.3 Cálculo de la complejidad (el método maestro).</p> <p>3.2 Algoritmos avaros.</p> <p>3.2.1 Calendarización de intervalos.</p> <p>3.2.2 El camino más corto.</p> <p>3.2.3 Árboles de mínima expansión.</p>	<p>Describe y aplica conceptos clave relacionados con algoritmos divide y vencerás, algoritmos avaros, programación dinámica y manejo de strings. Mediante la comprensión de algoritmos específicos como Merge-Sort, Quick- Sort, algoritmos de calendarización de intervalos, entre otros, podrán calcular la complejidad,</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales . ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.
--	--	---	---	---

	<p>3.3 Programación dinámica. 3.3.1 Ejemplo de funcionamiento. 3.3.2 Análisis de la complejidad. 3.3.3 Ejemplo de implementación.</p> <p>3.4 Manejo de strings. 3.4.1 Notación especial para strings. 3.4.2 Función Z. 3.4.3 Problema de la coincidencia de un patrón. 3.4.5 Algoritmo Knuth- Morris-Pratt (KMP).</p>	<p>implementar soluciones, y analizar la eficiencia de los algoritmos.</p>		
	<p>UNIDAD IV. APLICACIÓN DE LOS TIPOS DE ALGORITMOS.</p> <p>4.1 Problema de la mochila. 4.1.1 Versión divide y vencerás. 4.1.2 Versión programación dinámica.</p> <p>4.2 Problema del camino más corto. 4.2.1 Algoritmo de Dijkstra. 4.2.2 Algoritmo de Floyd.</p> <p>4.3 Árbol de mínima expansión. 4.3.1 Algoritmo de Prim. 4.3.2 Algoritmo de Kruskal.</p>	<p>Resuelve problemas prácticos utilizando diferentes tipos de algoritmos, entiende cómo aplicar diferentes enfoques algorítmicos en situaciones reales.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
<ul style="list-style-type: none"> • González Guerra, L. H., De la Cueva Hernández, V. M., & Pérez Murueta, P. Ó. (2022). Algoritmos: análisis, diseño e implementación. Editorial Digital del Tecnológico de Monterrey. • Kearns, M., & Roth, A. (2019). <i>The Ethical Algorithm: The Science of Socially Aware Algorithm Design</i>. Oxford University Press. ISBN 0190948221, 9780190948221. • Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. The MIT Press. • Baase, S. (2006). Algoritmos computacionales: Introducción al análisis y diseño. Addison Wesley. 	<p>Parcial 1 (30%)</p> <ul style="list-style-type: none"> • Tareas. (20%) • Proyecto. (50%) • Examen. (30%) <p>Parcial 2 (30%)</p> <ul style="list-style-type: none"> • Tareas. (20%) • Proyecto. (50%) • Examen. (30%) <p>Parcial 3 (40%)</p> <ul style="list-style-type: none"> • Tareas. (20%) • Proyecto. (50%) • Examen. (30%). <p>La calificación mínima es 7.0.</p> <p>Se usará rúbrica para la entrega de actividades o tareas a realizar.</p>

CRONOGRAMA

Objetos de estudio	Semanas																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
UNIDAD I. INTRODUCCIÓN A LA PROGRAMACIÓN ESTRUCTURADA.																	
UNIDAD II. INTRODUCCIÓN AL ANÁLISIS DE ALGORITMOS.																	
UNIDAD III. TIPOS DE ALGORITMOS.																	
UNIDAD IV. APLICACIÓN DE LOS TIPOS DE ALGORITMOS.																	