

## IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

<b>Unidad académica:</b> Instituto de Investigación en Ciencias Básicas y Aplicadas							
<b>Plan de estudios:</b> Licenciatura en Inteligencia Artificial							
<b>Unidad de aprendizaje:</b>  ALGORITMIA				<b>Ciclo de formación:</b> Profesional <b>Eje general de formación:</b> Teórico-Técnica <b>Área de conocimiento:</b> Fundamentos teóricos de la computación <b>Semestre:</b> 6º			
<b>Elaborada por:</b> Dr. Mario Abarca Dra. Elisa Chinos Olivan				<b>Fecha de elaboración:</b> Abril, 2021			
<b>Clave:</b>	<b>Horas teóricas</b> :	<b>Horas prácticas</b> :	<b>Horas totales</b> :	<b>Créditos</b> :	<b>Tipo de unidad de aprendizaje</b> :	<b>Carácter de la unidad de aprendizaje</b> :	<b>Modalidad:</b>
AL44CP04010 9	04	01	05	09	Obligatoria	Teórico - Práctica	Escolarizada
<b>Plan (es) de estudio en los que se imparte:</b> A partir de todos los programas impartidos por el Instituto de Investigación en Ciencias Básicas y Aplicadas.							

## ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

<p><b>Presentación:</b></p> <p>Los algoritmos son fundamentales en las ciencias computacionales y en la ingeniería de software. El comportamiento práctico de un software depende de los algoritmos usados y su eficiencia (el tiempo que van a tardar estos algoritmos en la práctica), por esta razón, es completamente crucial el diseño de algoritmos eficientes. En esta unidad de aprendizaje se estudian las bases para construir y desarrollar algoritmos eficientes, así como para demostrar la validez de estos algoritmos.</p>
---



**Propósito:**

Comprenda y utilice las herramientas básicas del análisis de algoritmos mediante el estudio de los conceptos teóricos y técnicas requeridas para reconocer aquellos problemas en los cuales no existe solución algorítmica práctica, así como formular y expresar algoritmos computacionales en forma correcta.

**Competencias que contribuyen al perfil de egreso**

**Competencias genéricas:**

- CG15. Capacidad para formular y gestionar proyectos.
- CG18. Capacidad para actuar en nuevas situaciones.
- CG20 capacidad de expresión y comunicación.

**Competencias específicas:**

- CE4. Analiza soluciones computacionales mediante la aplicación de fundamentos teóricos del diseño de algoritmos y estructuras de datos adecuadas para resolver problemas con pensamiento crítico.
- CE6 Contribuye al desarrollo científico y tecnológico, mediante la aplicación de conocimientos y habilidades computacionales adquiridos para la solución innovadora de problemas planteados en el sector público, privado o social con una actitud ética y responsable.

**CONTENIDOS**

Bloques	Temas
1. Introducción informal a los algoritmos	1.1. Definición intuitiva de un algoritmo y los medios de la expresión de un algoritmo. 1.2. Descripción de alto nivel (el pseudo-código). 1.3. El tamaño de un problema y su instancia. 1.3.1 Los esquemas de codificación de los problemas.



<p>2. Demostración de validez y análisis de la complejidad de algoritmos</p>	<p>2.1. Medidas de complejidad</p> <p>2.1.1. Notación "O" y "o".</p> <p>2.1.2. Complejidad de tiempo y espacio.</p> <p>2.1.3. Introducción a los problemas (algoritmos) polinomiales e intratables.</p> <p>2.1.4 Cotas inferior y superior.</p> <p>2.1.5 Clases de complejidad: P, NP, NP-completos</p> <p>2.2. Análisis de algoritmos</p> <p>2.2.1. Algoritmos iterativos y recursivos; análisis.</p> <p>2.2.2. Estimación de costos.</p> <p>2.2.3. Divide y vencerás. Aplicaciones: búsqueda binaria, árboles binarios, n-arios.</p> <p>2.2.4. Algoritmo de Euclides.</p> <p>2.2.5. Cobertura de árboles: preorder, inorder y postorder</p> <p>2.2.6 Algoritmos de búsqueda: Bubblesort, Mergesort, Quick sort</p>
<p>3. Algoritmos voraces</p>	<p>3.1. Definición de los algoritmos voraces.</p> <p>3.2. Cobertura mínima de árboles ("minimalspanningtrees"): los algoritmos de Prim y Kruskal.</p> <p>3.3. Problema de los tiempos de espera.</p> <p>3.4. Problema de la mochila no-entera.</p> <p>3.5 Calendarización de trabajos a plazos.</p>
<p>4. Introducción a la programación dinámica</p>	<p>4.1. El método general</p> <p>4.2. Rutas más cortas en grafos con pesos no-negativos: el algoritmo de Dijkstra.</p> <p>4.3. El problema de la mochila 0/1.</p> <p>4.4. El problema del agente viajero.</p> <p>4.5. Problema de calendarización flow shop.</p>



## ESTRATEGIAS DE ENSEÑANZA - APRENDIZAJE

Estrategias de aprendizaje sugeridas (Marque X)			
Aprendizaje basado en problemas	( X )	Nemotecnia	( )
Estudios de caso	( )	Análisis de textos	( X )
Trabajo colaborativo	( )	Seminarios	( )
Plenaria	( )	Debate	( )
Ensayo	( )	Taller	( )
Mapas conceptuales	( )	Ponencia científica	( )
Diseño de proyectos	( X )	Elaboración de síntesis	( )
Mapa mental	( )	Monografía	( )
Práctica reflexiva	( )	Reporte de lectura	( )
Trípticos	( )	Exposición oral	( X )
Otros			
Estrategias de enseñanza sugeridas (Marque X)			
Presentación oral (conferencia o exposición) por parte del docente	( X )	Experimentación (prácticas)	( )
Debate o Panel	( )	Trabajos de investigación documental	( )
Lectura comentada	( X )	Anteproyectos de investigación	( )
Seminario de investigación	( )	Discusión guiada	( )
Estudio de Casos	( )	Organizadores gráficos (Diagramas, etc.)	( )
Foro	( )	Actividad focal	( )
Demostraciones	( )	Analogías	( )
Ejercicios prácticos (series de problemas)	( X )	Método de proyectos	( )



Interacción la realidad (a través de videos, fotografías, dibujos y software especialmente diseñado).	( )	Actividades generadoras de información previa	( X )
Organizadores previos	( )	Exploración de la web	( X )
Archivo	( )	Portafolio de evidencias	( )
Ambiente virtual (foros, chat, correos, ligas a otros sitios web, otros)	( )	Enunciado de objetivo o intenciones	( )

### CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
<ul style="list-style-type: none"> <li>• Exámenes.</li> <li>• Participación en clase.</li> <li>• Exposiciones</li> <li>• Búsqueda de información.</li> <li>• Tareas.</li> </ul>	30%
	20%
	10%
	20%
	20%
<b>Total</b>	100 %

### PERFIL DEL PROFESORADO

Maestría o Doctorado en ciencias computacionales, matemáticas o ingeniería en áreas afines a las ciencias computacionales, con experiencia docente en el área.

### REFERENCIAS

#### Básicas:

- G. Brassard, P. Brathey. Fundamentos de Algoritmia. Ian Parberry (2001). Lecture Notes on Algorithm Analysis and Computational Complexity. Fourth Edition. Department of Computer Sciences University of North Texas.
- Horowitz, E. y Sahni, S. 1978. Fundamentals of computer algorithms. Ed. Computer Science Press.
- Aho, J. Hopcroft y Ulman, J. 1974. The design and analysis of computer algorithms. Ed. Addison-Wesley.
- Dasgupta, Sanjoy, Papadimitriou, Christos y Vazirani, Umesh. 2006. Algorithms. McGraw-Hill.

#### Complementarias:

- Aho, J. Hopcroft y Ullman, J. (1988). *Estructuras de datos y algoritmos*. Ed. Addison-Wesley.
- Levitin, Anany. (2006). *Introduction to the design & analysis of algorithms*. 2a edición. Ed. Addison Wesley.

