



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Estudios Superiores Aragón
Plan de Estudios



Ingeniería en Computación
Diseño y Análisis de Algoritmos

Clave	Semestre	Créditos	Área	
	5	9.0	Programación e Ingeniería de Software	
Modalidad	Curso		Tipo	Teórico
Carácter	Obligatorio			
Horas				
Semana			Semestre	
Teóricas	4.5		Teóricas	72.0
Prácticas	0.0		Prácticas	0.0
Total	4.5		Total	72.0

Seriación indicativa

Asignaturas antecedentes	Ninguna
Asignaturas subsecuentes	Ingeniería de Software

Objetivo general: Obtener los conocimientos necesarios para la abstracción de problemas y poderles dar solución a través de algoritmos optimizados y eficientes.

Índice temático

No.	Tema	Horas Semestre	
		Teóricas	Prácticas
1	INTRODUCCIÓN A LOS ALGORITMOS	9.0	0.0
2	PROBLEMAS Y SU MODELADO	9.0	0.0
3	METODOLOGÍA EN EL DISEÑO DE ALGORITMOS	12.0	0.0
4	COSTO COMPUTACIONAL	12.0	0.0
5	ALGORITMOS COMPUTACIONALES	12.0	0.0
6	TEORÍA DE GRAFOS	12.0	0.0
7	ALGORITMOS NO DETERMINISTAS	6.0	0.0
Total		72.0	0.0
Suma total de horas		72.0	



CONSEJO ACADÉMICO DEL ÁREA DE LAS
 CIENCIAS FÍSICO-MATEMÁTICAS
 Y DE LAS INGENIERÍAS

Contenido Temático	
1. INTRODUCCIÓN A LOS ALGORITMOS	
Objetivo: Plantear algoritmos detallados para problemas sencillos.	
1.1	Algoritmos.
1.2	Pasos básicos en el desarrollo completo de un algoritmo.
1.3	Desarrollo completo de un algoritmo.
1.4	Implementación de algoritmos por medio de lenguajes de programación.
1.5	Depuración de errores.
2. PROBLEMAS Y SU MODELADO	
Objetivo: Crear modelos de problemas reales.	
2.1	Modelos y realidad.
2.2	Propiedades de los modelos.
2.3	Construcción del modelo.
2.4	Necesidad de modelar.
2.5	Métodos elementales.
3. METODOLOGÍA EN EL DISEÑO DE ALGORITMOS	
Objetivo: Conocer las estrategias básicas para resolver problemas.	
3.1	Subjetivas, ensayo, error y trabajo en retrospectiva.
3.2	Heurística.
3.3	Programación directa.
3.4	Recursividad.
4. COSTO COMPUTACIONAL	
Objetivo: Calcular el costo computacional de diferentes algoritmos.	
4.1	Concepto de costo computacional.
4.2	Órdenes de crecimiento ($n \log(n)$, n , $\log(n)$, n^2 , $n!$, etc).
4.3	Ejemplos de costo computacional en diferentes algoritmos.
5. ALGORITMOS COMPUTACIONALES	
Objetivo: Resolver los problemas más comunes del área de cómputo.	
5.1	Ordenamiento.
5.2	Búsqueda.
5.3	Expresiones lógicas y aritméticas.
5.4	Programación en paralelo.
6. TEORÍA DE GRAFOS	
Objetivo: Conocer la teoría básica de grafos, así como los costos de recorrerlos, ordenarlos y otras operaciones básicas.	
6.1	Representaciones de grafos.
6.2	Caminos.
6.3	Caminos con pesos.
6.4	Árboles

7. ALGORITMOS NO DETERMINISTAS

Objetivo: Conocer los algoritmos no deterministas básicos.

- 7.1 Caos, fractales y modelos de la naturaleza.
- 7.2 Redes neuronales.
- 7.3 Algoritmos genéticos.

Estrategias didácticas		Evaluación del aprendizaje		Recursos	
Exposición	(X)	Exámenes parciales	(X)	Aula interactiva	()
Trabajo en equipo	(X)	Examen final	(X)	Computadora	(X)
Lecturas	(X)	Trabajos y tareas	(X)	Plataforma tecnológica	(X)
Trabajo de investigación	(X)	Presentación de tema	(X)	Proyector o Pantalla LCD	(X)
Prácticas (taller o laboratorio)	()	Participación en clase	(X)	Internet	(X)
Prácticas de campo	()	Asistencia	()		
Aprendizaje por proyectos	()	Rúbricas	()		
Aprendizaje basado en problemas	(X)	Portafolios	()		
Casos de enseñanza	()	Listas de cotejo	()		
Otras (especificar)		Otras (especificar)		Otros (especificar)	

Perfil profesiográfico	
Título o grado	<ul style="list-style-type: none"> • Poseer un título a nivel licenciatura en Ingeniería en Computación, Ciencias de la Computación, Matemáticas Aplicadas a la Computación o carreras cuyo perfil sea afín al área de Programación e Ingeniería de Software.
Experiencia docente	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia profesional relacionados con los contenidos de la asignación a impartir. • Tener la vocación para la docencia y una actitud permanentemente educativa a fin de formar íntegramente al alumno: <ul style="list-style-type: none"> ○ Para aplicar recursos didácticos. ○ Para motivar al alumno. ○ Para evaluar el aprendizaje del alumno, con equidad y objetividad.
Otra característica	<ul style="list-style-type: none"> • Poseer conocimientos y experiencia pedagógica referentes al proceso de enseñanza-aprendizaje. • Tener disposición para su formación y actualización, tanto en los conocimientos de su área profesional, como en las pedagógicas. • Identificarse con los objetivos educativos de la institución y hacerlos propios. • Tener disposición para ejercer su función docente con ética profesional: <ul style="list-style-type: none"> ○ Para observar una conducta ejemplar fuera y dentro del aula. ○ Para asistir con puntualidad y constancia a sus cursos. ○ Para cumplir con los programas vigentes de sus asignaturas.

Bibliografía básica	Temas para los que se recomienda
Baase, S. (2002). <i>Algoritmos Computacionales: Introducción al análisis y diseño</i> . México: Pearson.	1,2,3,4,5,6 y 7
Bisbal, J. (2009). <i>Recursividad, complejidad y diseño de algoritmos</i> . España: Editorial UOC.	1,2,3,4,5,6 y 7
Cormen, T. (2009). <i>Introduction to Algorithms</i> . USA: The MIT Press.	1,2,3,4,5,6 y 7
Drozdek, A. (2007). <i>Estructuras de datos y algoritmos con Java</i> . México: Thomson Learning.	1,2,3,4,5,6 y 7



Joyanes, A. L. (2004). <i>Algoritmos y estructuras de datos. Una perspectiva en C.</i> España: McGraw-Hill.	1,2,3,4,5,6 y 7
Kenneth, B. (2005). <i>Algorithms: Sequential, Parallel, and Distributed.</i> U.S.A: Ed. Thomson.	5
Mehlhorn, K. (2010). <i>Algorithms and Data Structures: The Basic Toolbox.</i> USA: Springer.	1,2,3,4,5,6 y 7
Skiena, S. (2010). <i>The Algorithm Design Manual.</i> New York: Springer-Verlag.	1,2,3,4,5,6 y 7

Bibliografía complementaria	Temas para los que se recomienda
Gutttag, J. (2013). Introduction to Computation and Programming Using Python. USA: MIT Press.	1,2,3,4,5 y 6