



UNIDAD	IZTAPALAPA	DIVISION	CIENCIAS BASICAS E INGENIERIA	1 / 3
NOMBRE DEL PLAN LICENCIATURA EN MATEMATICAS				
CLAVE	UNIDAD DE ENSEÑANZA-APRENDIZAJE		CRED.	10
2151116	ANALISIS Y DISEÑO DE ALGORITMOS		TIPO	OPT.
H. TEOR. 4.0	SERIACION 2150008 Y 72 CREDITOS DE FD		TRIM.	
H. PRAC. 2.0			IX	

OBJETIVO(S):

Objetivo General:

Al final de la UEA el alumno será capaz de:

Comprender y aplicar, los conceptos asociados a algoritmos para ejecutarse en algún sistema de cómputo.

Objetivos Específicos:

Que al final de la UEA el alumno sea capaz de:

- Estimar los recursos de cómputo (memoria y tiempo) que un algoritmo requiere para ejecutarse y determinar su complejidad.
- Comprender las principales técnicas de diseño de algoritmos para construir soluciones eficientes a diversas clases de problemas.
- Reconocer los problemas para los cuales no se conocen algoritmos polinomiales que los resuelvan y aplicar diferentes criterios que proporcionen soluciones aproximadas eficientes.

CONTENIDO SINTETICO:

1. Análisis básico. (3 semanas)
 - 1.1 Dominio asintótico (O , o y notación theta) y clases de complejidad (n , $\log(n)$, etc.).
 - 1.2 Complejidad temporal y espacial de algoritmos.
 - 1.3 Análisis de algoritmos iterativos.
 - 1.4 Análisis de algoritmos recursivos.
2. Técnicas de diseño de algoritmos exactos. (4 semanas)



APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 360

EL SECRETARIO DEL COLEGIO

NOMBRE DEL PLAN LICENCIATURA EN MATEMATICAS		2/ 3
CLAVE 2151116	ANALISIS Y DISEÑO DE ALGORITMOS	

- 2.1 Fuerza bruta.
- 2.2 Divide y vencerás.
- 2.3 Backtrack.
- 2.4 Programación dinámica.

- 3. Métodos heurísticos. (4 semanas)
- 3.1 Naturaleza y medidas de desempeño.
- 3.2 Procedimiento glotón.
- 3.3 Mejoramiento local.
- 3.4 Algoritmos genéticos.

MODALIDADES DE CONDUCCION DEL PROCESO DE ENSEÑANZA-APRENDIZAJE:

El profesor utilizará la clase magistral para exponer los temas del curso propiciando la participación activa y corresponsable en el proceso de enseñanza-aprendizaje, el pensamiento crítico, la disciplina y el rigor en el trabajo académico, así como la capacidad para aprender por sí mismo. Para lograr lo anterior se podrán desarrollar actividades tales como tareas de programación, trabajo de investigación y exposición de temas.

Se deberán desarrollar exhaustivamente ejemplos y ejercicios sobre los temas abordados.

En las horas de práctica, el profesor utilizará la modalidad de taller para que los alumnos resuelvan problemas de manera individual o grupal.

MODALIDADES DE EVALUACION:

Evaluación Global:

La evaluación global consistirá de tres evaluaciones periódicas, dos escritas y una oral y, a juicio del profesor, una evaluación terminal.

El profesor seleccionará los elementos de evaluación periódica de entre los siguientes: exámenes, participación en clase, tareas de programación, trabajos de investigación y presentaciones de temas. Los factores de ponderación quedarán a juicio del profesor y se darán a conocer al inicio del curso.

Evaluación de Recuperación:

A juicio del profesor, consistirá en una evaluación que incluya todos los contenidos teóricos y prácticos de la UEA o sólo aquellos que no fueron



UNIVERSIDAD AUTÓNOMA METROPOLITANA

APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 360

EL SECRETARIO DEL COLEGIO

cumplidos durante el trimestre.

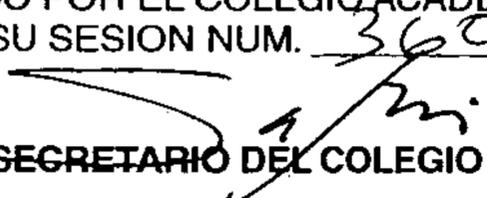
BIBLIOGRAFIA NECESARIA O RECOMENDABLE:

1. Aho, A., Hopcroft, J., Ullman J., (2002), The Design and Analysis of Computer Algorithms, Addison-Wesley Publishing Company, EUA.
2. Cormen, T., Leisserson, CH., Rivest, R., Stein C., (2009), Introduction to Algorithms, MIT Press 3a. Edición, EUA.
3. Lee, R.C.T., Tseng, S.S., Chang R.C., Tsai Y.T., (2007), Introducción al Diseño y Análisis de Algoritmos, Mc Graw Hill, México.
4. Levitin, A., (2011), Introduction to the design and Analysis of Algorithms, Addison Wesley 3a. Edición, EUA.
5. Goldberg, D.E., (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, EUA.
6. Michalewicz, Z., (1996), Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, EUA.
7. Neapolitan, R., Naimipour, K., (2011), Foundations of algorithms using C++ pseudocode, Jones and Bartlett Publishers Inc. 4a. Edición, EUA.
8. Papadimitriou, C. H., Steiglitz, K., (1998), Combinatorial Optimization: Algorithms and complexity, Dover, EUA.
9. Reeves, C. R., (1995), Modern Heuristics Techniques for Combinatorial Problems, McGraw-Hill, EUA.



UNIVERSIDAD AUTONOMA METROPOLITANA

APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 360


EL SECRETARIO DEL COLEGIO