

PROGRAMA DE ESTUDIOS: ANÁLISIS DE ALGORITMOS

PROTOCOLO

Fechas	Mes/año	Clave		Semestre	5°	
Elaboración	05-2010	Nivel	Licenciatura	X	Maestría	Doctorado
Aprobación		Ciclo	Integración	Básico	Superior	X
Aplicación	09-2010	Colegio	H. y C.S.	C. y T.	X	C. y H.

Plan de estudios del que forma parte: Ingeniería de Software

**Propósito(s) general(es):**

El estudiante comprenderá principios y técnicas generales del diseño de algoritmos. Analizará la complejidad computacional de diferentes algoritmos que resuelven un mismo problema con la finalidad de que adquiera las herramientas necesarias para diseñar algoritmos eficientes.

Carácter		Modalidad				Horas de estudio semestral (16 semanas)			
Indispensable	X	Seminario		Taller		Con Docen- te	Teóricas	36	Autónomas
		Curso	X	Curso-taller			Prácticas	36	Teóricas
Optativa *		Laboratori o	X	Clínica					20
						Carga horaria semanal: <b>4.5 + 3.5 = 8</b>		Carga horaria semestral:	128

**Asignaturas Previas:**

Estructuras de datos

**Asignaturas Posteriores:**

Diseño de software

**Requerimiento  
s para cursar la  
asignatura**

Conocimientos y habilidades:  
Programación en cualquier lenguaje de programación. Conocimiento de las diferentes estructuras de datos: lineales y arbóreas. Conceptos de matemáticas finitas.

**Perfil  
deseable del  
profesor:**

Ingeniero en computación o alguna carrera afín, con maestría o doctorado de preferencia, en Ciencias de la Computación, o bien, amplios conocimientos y experiencia laboral en el área. Además, amplia experiencia docente y de investigación.

**Academia responsable del programa:**

Informática

**Diseñador (es):**

Dr. Alfonso San Miguel Aguirre

\*Aquellas en las que se ofrece la posibilidad de cursar una de las asignaturas, para cubrir un requisito INDISPENSABLE será considerada INDISPENSABLE.

## INTRODUCCIÓN

La materia de análisis de algoritmos es una parte importante de la Teoría de complejidad computacional, provee estimaciones teóricas para los recursos que necesita cualquier algoritmo que resuelva un problema computacional dado. Estas estimaciones resultan ser bastante útiles en la búsqueda de algoritmos eficientes.

A la hora de realizar un análisis teórico de algoritmos es usual calcular su complejidad en un sentido asintótico, es decir, para un tamaño de entrada suficientemente grande. La cota superior asintótica y las notaciones omega y theta se usan con esa finalidad. Por ejemplo, la búsqueda binaria decimos que se ejecuta en una cantidad de pasos proporcional a un logaritmo, es decir, en  $O(\log(n))$ , o, coloquialmente hablando, “en tiempo logarítmico”.

Para su estudio se deben tener conocimientos de estructuras de datos, para que el estudiante sea capaz de implantar las distintas alternativas de solución de un problema dado. Asimismo, razonamiento en matemáticas discretas para que sea capaz de entender y analizar la complejidad de un algoritmo.

## PROPÓSITOS GENERALES

El estudiante comprenderá principios y técnicas generales del diseño de algoritmos. Analizará la complejidad computacional de diferentes algoritmos que resuelven un mismo problema con la finalidad de que adquiera las herramientas necesarias para diseñar algoritmos eficientes.

## CONTENIDOS

TEMAS Y SUBTEMAS	PROPÓSITOS ESPECIFICOS
<b>1. Análisis de eficiencia de los algoritmos</b> 1.1 Cotas inferiores y superiores 1.2 Complejidad promedio 1.3 Notación “O grande”	El estudiante será capaz de analizar, y en su caso, diseñar, algoritmos. Tendrá el conocimiento necesario para diferenciar soluciones malas y buenas (en cuanto a eficiencia).
<b>2. Algoritmos aritméticos y algebraicos.</b> 2.1 Complejidad en tiempo y en espacio 2.2 problemas P, NP y NP-completos	
<b>3. Algoritmos de búsqueda y ordenamiento</b> 3.1 Búsqueda lineal y binaria 3.2 Ordenamiento directo: selección e inserción directas 3.3 Ordenamiento eficiente: shellsort y quicksort	Buscar un valor o clave en un conjunto de datos es un problema básico en computación. Para que sea eficiente la búsqueda es conveniente que los valores se encuentren ordenados. Es así como se estudiarán distintos algoritmos de ordenamiento, no solamente para que el estudiante sea capaz de aplicarlos, sino también estos algoritmos brindan una excelente oportunidad para analizar su complejidad.

<b>4. Algoritmos de gráficas</b>	El estudiante continuará el estudio acerca del análisis y diseño de algoritmos. Concentrándose en algoritmos en gráficas, fortalecerá sus conocimientos de teoría de gráficas. También fortalecerá su formación en teoría de computación y solución de problemas de informática.
4.1 Representación en computadora	
4.2 Obtención del número cromático	
4.3 Árbol generador	
4.4 Número de independencia.	
<b>5. Técnicas de diseño de algoritmos</b>	El estudiante, como colofón del curso, tendrá conocimiento de todo un “arsenal” de técnicas de diseño de algoritmos para que, de este modo, pueda enfrentar nuevos problemas de manera rápida y eficiente.
5.1 Divide y vencerás	
5.2 Técnicas “glotonas”	
5.3 Genera y prueba	
5.4 Vuelta atrás automático backtracking)	
5.5 Montecarlo	

## METODOLOGÍA PARA EL CURSO

Curso teórico práctico. La materia se impartirá por medio de clases teóricas, prácticas y asesorías. Puesto que la aplicación en la práctica de los conocimientos adquiridos en el aula es indispensable, se tendrán sesiones en aula de computadoras una vez por semana, en ellas el estudiante implantará los algoritmos estudiados durante la teoría, para verificar su funcionamiento y complejidad. El estudiante podrá utilizar el lenguaje de programación que más le agrade, no estará obligado a utilizar alguno en particular.

La Universidad Autónoma de la Ciudad de México dispone de equipos de cómputo, PCs, para que el estudiante pueda programar los algoritmos vistos en la materia. A medida que la tecnología va avanzando es necesario ir actualizando este equipo, tal como se ha hecho hasta ahora.

## EVALUACIÓN DIAGNÓSTICA

Se aplicará un examen escrito sobre uso de la computadora y programación en algún lenguaje de programación, C, C++ C# o Java entre otros. Esto con el fin de detectar los conocimientos con los que cuentan los estudiantes y sugerir acciones para que adquieran los que no tengan, por medio de un repaso general dentro de la clase o en asesorías individuales.

## EVALUACIÓN FORMATIVA

Con el propósito de dar seguimiento al proceso de enseñanza aprendizaje, se propone aplicar tres evaluaciones formativas, la primera se aplicará al término del tema de problemas P y NP. La segunda evaluación será al terminar el tema de árboles donde los estudiantes podrán mostrar los conocimientos adquiridos sobre su representación, implementación y tratamiento. La tercera evaluación formativa se realizará al terminar el tema de técnicas de diseño de algoritmos donde el estudiante mostrará sus conocimientos acerca de cómo atacar un problema.

Los instrumentos para las evaluaciones formativas constarán tanto de preguntas sobre la teoría como de ejercicios de análisis y diseño de algoritmos. Además de las evaluaciones formativas se dejarán tareas a los estudiantes al terminar cada tema. Se hará un análisis junto con los estudiantes para evaluar los temas que presenten problemas y realizar una revisión de los mismos. En el caso de que sean pocos los estudiantes que tengan problemas con alguno de los temas, se tratarán en asesoría.

Se pondrá especial atención en el cumplimiento de las prácticas en el laboratorio, así como en la realización de las tareas, la participación y asistencia a clases, ya que estas actividades ayudan a la formación del estudiante.

## EVALUACIÓN DE CERTIFICACIÓN

El examen de certificación evaluará los conocimientos del estudiante sobre todos los temas tratados durante el curso. Este instrumento consta de dos partes: una evaluación teórica mediante examen escrito y la segunda mediante el desarrollo de un proyecto. Los temas tendrán la siguiente ponderación: Análisis de eficiencia de los algoritmos 5%, Problema P y NP 5%. Algoritmos de ordenamiento y búsqueda 10%, Algoritmos de gráficas 15%, Técnicas de diseño de algoritmos 15%. Los exámenes consistirán tanto de preguntas sobre la teoría como de ejercicios de análisis y diseño de algoritmos. Como proyecto final, que equivale al 50% de la evaluación de certificación, consistirá en resolver un problema de una manera eficiente, realizando un análisis de la complejidad del mismo y del algoritmo propuesto para solucionarlo. Deberá elaborar un programa correspondiente al algoritmo y probarlo con ejemplos representativos. Asimismo, deberá entregar un reporte escrito del proyecto.

## BIBLIOGRAFÍA

La bibliografía que se menciona a continuación es la básica para el estudiante y el profesor:

Núm.	Bibliografía	Temas para los que se recomienda
1	Aho, A., Hopcroft, A. and Ulmann, J. <b>The design and analysis of algorithms</b> . Reading, Mass., Addison Wesley, 1974.	1 -5
2	Baase, S. and Gelder, A. <b>Algoritmos computacionales: introducción al análisis y diseño</b> , Pearson Educación, 2001.	1-5
3	Brassard, G. y Bratley, P. <b>Algorítmica: Concepción y Análisis</b> . Masson 1990.	1, 2
4	Manber, U. <b>Introduction to Algorithms – A Creative Approach</b> . Addison Wesley 1989.	1-4
5	Mark Allen Weiss, <b>Estructuras de Datos y Algoritmos</b> . Addison Wesley 1995.	1, 2
6	Sedgewick, R., Flajolet and Gordon P. <b>An Introduction to the Analysis of Algorithms</b> , Addison-Wesley, 1996.	1-5

## OTROS RECURSOS

Pintarrón, computadora laptop para las presentaciones y proyector de video (cañón).