

Nombre de la asignatura: **Análisis y Diseño de Algoritmos**
 Línea de trabajo: Modelación inteligente de sistemas, PDI y Visión Artificial
 Tiempo de dedicación del estudiante a las actividades de
 DOC - TIS - TPS - Horas totales. Créditos
 48-20-100-168-6

DOC: Docencia; **TIS:** Trabajo independiente significativo; **TPS:** Trabajo profesional supervisado

1. Historial de la asignatura.

Fecha revisión / actualización	Participantes	Observaciones, cambios o justificación
14 de Mayo de 2011	Nombres de los participantes Saul Martínez Díaz	Reestructuración del plan y programas de estudio de la Maestría en Sistemas Computacionales

2. Pre-requisitos y co-requisitos.

Ninguno.

3. Objetivo de la asignatura.

Analizar, diseñar e implementar algoritmos eficientes y correctos para la solución computacional de problemas de diferentes clases de complejidad.

4. Aportación al perfil del graduado.

La materia contribuye a la conformación de criterios científicos y críticos para el análisis y diseño de algoritmos eficientes, en diferentes aplicaciones.

5. Contenido temático.

Unidad	Temas	Subtemas
I Fundamentos de análisis de algoritmos	1.1 Problemas y algoritmos. 1.2 Métodos de análisis y diseño de algoritmos. 1.3 Crecimiento de funciones. 1.4 Métodos de solución de recurrencias.	1.1.1 Algoritmos como tecnologías. 1.2.1 Análisis de algoritmos. 1.2.2 Análisis del peor caso y caso promedio. 1.2.3 Orden de crecimiento. 1.2.4 Algoritmo divide y vencerás. 1.3.1 Notación asintótica. 1.3.2 Notación asintótica en ecuaciones y desigualdades. 1.4.1 Recurrencias lineales. 1.4.2 Método de sustitución. 1.4.3 Método del árbol de recursión

		1.4.4 Método maestro.
II Métodos de ordenamiento	<ul style="list-style-type: none"> 2.1 Heapsort. 2.2 Quicksort. 2.3 Ordenamiento lineal. 2.4 Estadística de orden. 	<ul style="list-style-type: none"> 2.1.1 El algoritmo Heapsort. 2.2.1 Descripción del Quicksort. 2.2.2 Análisis del Quicksort. 2.2.3 Versión aleatoria del Quicksort. 2.3.1 Cota inferior para el ordenamiento. 2.3.2 Counting sort. 2.3.3 Radix sort. 2.4.1 Mínimo y máximo. 2.4.2 Selección con tiempo lineal esperado. 2.4.3 Selección con el peor caso de tiempo lineal.
III Teoría de grafos y árboles.	<ul style="list-style-type: none"> 3.1 Estructura de datos elementales. 3.2 Árboles de búsqueda binaria. 3.3 Árboles rojo-negro. 3.4 Fundamentos de algoritmos para grafos. 3.5 Árboles de esparcimiento mínimo. 	<ul style="list-style-type: none"> 3.1.1 Pilas, colas y listas ligadas. 3.1.2 Representación de árboles. 3.2.1 Inserción y eliminación en árboles binarios. 3.2.2 Búsqueda en árbol binario. 3.3.1 Propiedades de los árboles rojo-negros. 3.3.2 Rotaciones. 3.3.3 Inserción y eliminación. 3.4.1 Representación de grafos. 3.4.2 Recorrido en anchura. 3.4.3 Recorrido en profundidad. 3.4.4 Ordenamiento topológico. 3.4.5 Componentes fuertemente conectados. 3.5.1 Creación de un árbol de esparcimiento mínimo. 3.5.2 Algoritmo de Kruskal. 3.5.3 Algoritmo de Prim.
IV Complejidad computacional	<ul style="list-style-type: none"> 4.1 Tiempo polinomial. 4.2 Verificación en tiempo polinomial. 4.3 NP-Complejidad y reducibilidad. 4.4 Pruebas de complejidad. 	<ul style="list-style-type: none"> 4.1.1 Problemas abstractos y concretos. 4.1.2 Codificación. 4.1.3 Complejidad de la clase P. 4.2.1 Problemas verificables en tiempo polinomial. 4.2.2 Algoritmos de verificación. 4.2.3 Complejidad de la clase NP. 4.2.4 Ciclo Hamiltoniano. 4.3.1 Reducibilidad.

		4.3.2 NP-Complejidad. 4.3.3 El problema del circuito SAT 4.4.2 3-CNF SAT
V Algoritmos de aproximación.	5.1 Problema de cubrimiento de vértices. 5.2 Problema del comerciante viajero. 5.3 Problema de cubrimiento de conjuntos.	5.1.1 Algoritmo de aproximación para el cubrimiento de vértices. 5.2.1 Problema del agente viajero con la desigualdad del triángulo. 5.2.2 Problema general del comerciante viajero. 5.2.3 Algoritmo voraz para el cubrimiento de conjuntos.

6. Metodología de desarrollo del curso.

El curso se enfocará en el análisis de algoritmos clásicos y las técnicas de diseño de algoritmos.

7. Sugerencias de evaluación.

Se evaluará la participación y cooperación de los estudiantes, así como su competencia en el análisis y diseño de algoritmos.

8. Bibliografía y Software de apoyo.

- Cormen, T., Leiserson, D., Rivest, R., y C. Stein. *Introduction to Algorithms*, MIT Press, 2001.
- Skiena, Steven S. *The Algorithm design manual*. Santa Clara, CA : TELOS-the Electronic Library of Science, 1998
- G. Brassard y P. Bratley. *Fundamentos de Algoritmia*. Prentice Hall.
- Aho, Hopcroft y Ullmann, *Algorithms and Data Structures*, 1974
- Niklaus Wirth. *Algorithms & Data Structures*. Prentice Hall.
- Sedgewick, R., Flajolet P. *An Introduction to the Analysis of Algorithms*, Addison-Wesley, 1996

9. Actividades propuestas.

Unidad	Prácticas
I.	Realizar un análisis comparativo del tiempo de ejecución de algoritmos de diferente orden.
II.	Análisis de diferentes algoritmos de ordenamiento
III.	Exposición de algoritmos con grafos y sus aplicaciones
IV.	Solución de problemas de reducibilidad
V	Investigación sobre aplicaciones prácticas de los algoritmos de aproximación