

## Programa de Asignatura

### Historia del programa

Lugar y fecha de elaboración	Participantes	Observaciones (Cambios y justificaciones)
Cancún, Q. Roo, 03/12/2016	Dr. David Israel Flores Granados	Se llevó a cabo una reestructuración completa al separarse los temas de TDA en una nueva asignatura.

### Relación con otras asignaturas

Anteriores	Posteriores
Algoritmos y Estructura de datos. a) Estructuras de datos estáticas. b) Estructuras de control.	Programación Orientada a Objetos a) Tipos de datos abstractos.

Nombre de la asignatura	Departamento o Licenciatura
Técnicas algorítmicas	Ingeniería en Datos e Inteligencia Organizacional

Ciclo	Clave	Créditos	Área de formación curricular
2 - 2	IT0107	8	Profesional Asociado y Licenciatura Básica

Tipo de asignatura	Horas de estudio			
	HT	HP	TH	HI
Seminario	48	16	64	64

### Objetivo(s) general(es) de la asignatura

Objetivo cognitivo

Describir las principales estrategias de algoritmia y complejidad para el diseño de algoritmos.

### Objetivo procedimental

Usar técnicas de algoritmia para dar solución a problemas computables.

### Objetivo actitudinal

Fomentar la disciplina para el análisis y diseño de soluciones a problemas computables.

## Unidades y temas

---

### Unidad I. INTRODUCCIÓN AL ANÁLISIS DE ALGORITMOS

Revisar las principales metodologías de algoritmia para la realización el análisis a los principales algoritmos de ordenamiento y búsqueda.

#### 1) Análisis de complejidad

a) Tiempo de ejecución y orden de crecimiento

b) Métodos de análisis de complejidad

c) El método maestro

#### 2) Algoritmos de Ordenamiento

a) Insertion Sort

b) Bubble Sort

c) Quicksort

d) Heapsort

#### 3) Algoritmos de Búsqueda

a) Búsqueda secuencial

b) Búsqueda binaria

## Unidad II. ALGORITMOS VORACES

Utilizar algoritmos voraces para la solución de problemas computables.

- 1) Definición
- 2) Algoritmos para grafos
  - a) Algoritmo de Kruskal
  - b) Algoritmo de Prim
  - c) Algoritmo de Dijkstra

## Unidad III. DIVIDE Y VENCERÁS

Utilizar la técnica de divide y vencerás para la solución de problemas computables.

- 1) Precedentes históricos
- 2) Recursividad y divide y vencerás
- 3) Algunos problemas utilizando divide y vencerás

## Unidad IV. PROGRAMACIÓN DINÁMICA

Utilizar la técnica de programación dinámica para la solución de problemas computables.

- 1) Elementos de la programación dinámica
- 2) Aplicaciones
  - a) Problema LCS (Longest Common Subsequence)
  - b) Problema de Calendarización de tareas.

## Actividades que promueven el aprendizaje

---

### Docente

Promover el trabajo individual en la definición de propuestas de solución a problemas determinados.

Coordinar la discusión de casos prácticos.

Realizar demostraciones de algoritmos.

Aplicar prácticas para la definición de algoritmos.

Definir estrategias para identificar las principales técnicas algorítmicas para resolver problemas computables.

### Estudiante

Realizar tareas asignadas.

Participar en el trabajo individual y en equipo.

Resolver casos prácticos.

Discutir temas en el aula.

Participar en actividades extraescolares.

## Actividades de aprendizaje en Internet

---

El estudiante deberá acceder al portal (señalar las actividades que realizarán):

Se promoverá el uso de mecanismos asíncronos (correo electrónico, grupo de noticias, WWW y tecnologías de información) como medio de comunicación.

## Criterios y/o evidencias de evaluación y acreditación

---

### Criterios

Examen

Tareas

Evidencias individuales

Evidencias grupales

Total

### Porcentajes

30

30

20

20

100

## Fuentes de referencia básica

---

### Bibliográficas

Cormen. (2009) Introduction to Algorithms. (3a Edición). EUA: MIT Press

Donald E. Knuth. (2011) The Art of Computer Programming. (1era Edición). EUA: Addison Wesley

Gilles Brassard. (1999) Fundamentos de Algoritmos. (2da Edición) EUA: Prentice Hall

Steven Skiena. (2002) Programming Challenges. (2da Edición) EUA: Springer Verlag

Weiss. (2000) Data Structures and Problem Solving Using Java. (2da Edición) EUA: Addison Wesley

### **Web gráficas**

<http://domino.research.ibm.com/comm/research.nsf/pages/r.algorithms.html> 28 de Junio de 2010

## **Fuentes de referencia complementaria**

---

### **Bibliográficas**

Robert Sedgewick. (2005) Algorithms in C++ part 5. Graph Algorithms. (1era Edición) EUA: Addison Wesley

Vazirani. (2007) Approximation Algorithms. (2da Edición). EUA: Springer.

### **Web gráficas**

.

## **Perfil profesiográfico del docente**

---

### **Académicos**

Maestría en Ciencias de la computación, Maestría en Ingeniería en Sistemas.

### **Docentes**

Tener experiencia docente a nivel superior mínima de 3 años en ingeniería.

### **Profesionales**

Tener experiencia en desarrollo de sistemas.