

Programa del curso IC-3002

## **Análisis de Algoritmos**

Escuela de Computación  
Carrera de Ingeniería en Computación, Plan 411.

## I parte: Aspectos relativos al plan de estudios

### 1 Datos generales

<b>Nombre del curso:</b>	Análisis de Algoritmos
<b>Código:</b>	IC-3002
<b>Tipo de curso:</b>	Teórico-Práctico
<b>Electivo o no:</b>	No
<b>Nº de créditos:</b>	4
<b>Nº horas de clase por semana:</b>	4
<b>Nº horas extraclase por semana:</b>	8
<b>Ubicación en el plan de estudios:</b>	Curso del III Semestre del Bachillerato de Ingeniería en Computación
<b>Requisitos:</b>	IC-2001 Estructuras de Datos MA-1404 Cálculo
<b>Correquisitos:</b>	Ninguno
<b>El curso es requisito de:</b>	IC-4700 Lenguajes de Programación
<b>Asistencia:</b>	Obligatoria
<b>Suficiencia:</b>	No
<b>Posibilidad de reconocimiento:</b>	Sí
<b>Vigencia del programa:</b>	I Semestre de 2018

## **2 Descripción general**

El curso de análisis de algoritmos dota al estudiante de la experticia y conocimientos necesarios para analizar e implementar los diversos tipos de algoritmos y estrategias existentes en la resolución de problemas. Además provee al mismo de las herramientas que lo van a fortalecer en temas avanzados de ingeniería de software y ciencias de la computación.

## **3 Objetivos**

### **Objetivo General**

Implementar algoritmos que resuelvan problemas sensibles al tamaño de la entrada, el tiempo y la eficiencia de la solución.

### **Objetivos Específicos**

1. Identificar la medición de la eficiencia de los algoritmos y la programación de algoritmos utilizando buenas prácticas y estructuración del código en capas.
2. Determinar diversas estrategias para el diseño e implementación de algoritmos.
3. Aplicar técnicas para el manejo eficiente de la memoria primaria y secundaria.
4. Examinar situaciones y ejemplos que impliquen la protección de datos e información en sistemas computacionales utilizando las tecnologías actuales para cifrado y protección.
5. Dominar conceptos de complejidad espacial y temporal, como temas avanzados de computabilidad e intratabilidad

## **4 Contenidos**

### **1. Eficiencia, medición y análisis y orden de los Algoritmos.**

- 1.1. Definiciones de análisis de algoritmos
- 1.2. Importancia del análisis
- 1.3. Definición e Importancia de la eficiencia
- 1.4. Complejidad Temporal y Espacial
- 1.5. Orden de los algoritmos
- 1.6. Técnicas de medición de algoritmos
- 1.7. Medición Analítica
- 1.8. Caso medio, Mejor caso y Peor caso
- 1.9. Familias O grande, Omega, Theta y o pequeña
- 1.10. Cálculo de funciones O de N

- 1.11. Ejemplos: cálculo de determinante , cálculo común divisor,
- 1.12. transformadas de Fourier
- 1.13. Análisis de algoritmos ejemplos
- 1.14. Burbuja
- 1.15. QuickSort
- 1.16. Búsqueda binaria
  
- 2. Intratabilidad; introducción a problemas NP.**
  - 2.1. Modelos de computación
  - 2.2. Máquinas de Turing
  - 2.3. Intratabilidad
  - 2.4. Reconsiderando el tamaño de la entrada
  - 2.5. Algoritmos polinomiales, probados como no tratables, probados como no tratables que nunca se ha encontrado un algoritmo polinomial
  - 2.6. Teoría de NP
  - 2.7. NP-Duro, NP-Fácil
  - 2.8. Equivalencia de problemas
  
- 3. Divide y conquistarás.**
  - 3.1. Enfoque general
  - 3.2. El caso de la búsqueda binaria, el merge sort y el quicksort como ejemplos
  - 3.3. Su uso en la programación modular
  
- 4. Programación dinámica.**
  - 4.1. Definición
  - 4.2. Principio de optimalidad
  - 4.3. Manifestaciones de la programación dinámica
  - 4.4. Ejemplos con algoritmo de Floyd, problema del agente viajero
  
- 5. Algoritmos “Greedy”.**
  - 5.1. Definición de algoritmo voraz
  - 5.2. Determinación de las etapas y los óptimos locales y globales
  - 5.3. Ejemplos con Spanning Trees, Arboles de Huffman entre otros
  
- 6. Backtracking.**

- 6.1. Definición de la estrategia
- 6.2. Vector solución inicial
- 6.3. Tanteos y fuerza bruta
- 6.4. Criterios de poda
- 6.5. Problema de las N reinas
- 6.6. Otros algoritmos ejemplo

**7. Algoritmos genéticos.**

- 7.1. Principios de evolución genética de los algoritmos
- 7.2. Algoritmo general genético
- 7.3. Determinación de población, función de adaptabilidad, cruce y mutación
- 7.4. Resolución de problemas utilizando algoritmos genéticos
- 7.5. Ejemplos de Algoritmos Genéticos

**8. Algoritmos probabilísticos.**

- 8.1. Tipos de algoritmos probabilísticos: Montecarlo y las vegas
- 8.2. Uso de algoritmos probabilistas para resolver problemas determinísticos de mediana complejidad
- 8.3. Uso de probabilidad y aleatoriedad para resolver problemas no polinomiales
- 8.4. Algoritmos heurísticos basados en probabilidad

**9. Complejidad computacional con problemas de búsqueda y ordenamiento.**

- 9.1. Complejidad computacional
- 9.2. Problema de ordenamiento
- 9.3. Límites inferiores
- 9.4. Límites inferiores en algoritmos de comparación de llaves
- 9.5. Análisis del algoritmo de ordenamiento de inserción, selección y heapsort
- 9.6. Problema de búsqueda
- 9.7. Límites **inferiores** en búsqueda por llave, Análisis de hashing, interpolación y B-trees

**10. Computación paralela. 2 semanas.**

- 10.1. Arquitectura de software y hardware para computación paralela

- 10.2. Algoritmos paralelos
- 10.3. Modelo de computación paralela
- 10.4. Operaciones síncronas y asíncronas
- 10.5. Eventos, delegados, escuchadores
- 10.6. Operadores de espera y activación de hilos
- 10.7. Uso de múltiples procesadores y afinidad de procesadores

#### **11. Métodos de compresión y cifrado. 2 semanas**

- 11.1. Algoritmos de LZW y de codificación
- 11.2. Algoritmos de llave privada
- 11.3. Algoritmos de llave pública
- 11.4. Algoritmos de protección con MD5

## **II parte: Aspectos operativos**

### **5 Metodología de enseñanza y aprendizaje**

Se utilizará una metodología de aula invertida en la que la persona estudiante realizará lecturas y trabajará los contenidos teóricos del curso en el tiempo extraclase a través del material provisto por la persona docente. Las horas de clase presencial se utilizarán para resolución de consultas y trabajo práctico de aplicación de los contenidos con acompañamiento de la persona docente.

### **6 Evaluación**

Se realizará una evaluación basada en tres componentes: trabajo en lecturas, tareas cortas de implementación de código y exámenes escritos.

Semanalmente se asignarán lecturas con los contenidos teóricos del curso, se evaluará la participación activa de cada estudiante en la lectura colectiva de cada una de ellas a través de la plataforma Perusall.

Las tareas cortas tienen como objetivo que la persona estudiante trabaje en el diseño e implementación de algoritmos para resolver problemas específicos aplicando las distintas técnicas estudiadas durante la semana. Algunas tareas cortas serán actividades para poner en práctica la teoría matemática estudiada en clase.

La evaluación de la comprensión de los conceptos teóricos fundamentales del análisis de algoritmos se hará a través de dos exámenes escritos.

Rubro	Porcentaje
Lecturas	20%
Tareas Cortas	30%
Examen 1	25%
Examen 2	25%
Total	100%

#### Cronograma de Actividades

Actividad	Semanas
Eficiencia, medición y análisis y orden de los Algoritmos	1 - 6
Complejidad computacional con problemas de búsqueda y ordenamiento	1 - 7
Divide y conquistarás	7
Intratabilidad; introducción a problemas NP	10
Backtracking	11
Programación dinámica	12
Algoritmos "Greedy"	13
Algoritmos genéticos	14
Algoritmos probabilísticos	14
Computación paralela	15
Métodos de compresión y cifrado	16

## 7 Bibliografía

### Obligatoria

Brassard, G., & Bratley, P. (2001). *Fundamentos de Algoritmia*. Prentice Hall.

Joyanes, L., & Zahonero, I. (1998). *Estructuras de Datos: Algoritmos, abstracción y objetos*. Mc Graw Hill.

Drozdek, A. (2001). *Data Structures and Algorithms in Java*. Estados Unidos: Brooks/Cole-Thomson Learning.

Weiss, M. A. (2006). *Data Structures & Algorithm Analysis in JAVA* (2da edición ed.). Harlow, Inglaterra: Addison Wesley Longman.

Weiss, M. A. (2006). *Data Structures & Algorithm Analysis in C++* (3ra edición ed.). Harlow, Inglaterra: Addison Wesley Longman.

Rawlins, G. J. (1991). *Compare to What? An introduction to the Analysis of Algorithms*. W. H. Freeman.

Neapolitan, R., & Naimipour, K. (2009). *Foundations of Algorithms* (4th Edition ed.). Jones and Bartlett.

### Adicional

No tiene Bibliografía adicional.

## 8 Profesor

Máster Diego Munguía Molina  
Telegram @diegomunguia  
Correo [dmunguia@itcr.ac.cr](mailto:dmunguia@itcr.ac.cr)

### Horario de consulta

Lunes 11:00 a 12:00 o cita acordada previamente  
Oficina Laboratorio Experimental - Edificio Luis Alberto Monge