



**FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA DE SISTEMAS**

SILABO DEL CURSO

ANALISIS Y DISEÑO DE ALGORITMOS

1. DATOS GENERALES

- 1.1. Facultad : Ingeniería
- 1.2. Carrera Profesional : Ingeniería de Sistemas
- 1.3. Departamento : Ingeniería de Sistemas
- 1.4. Tipo de Curso : Obligatorio
- 1.5. Requisitos : Algoritmos y Estructuras de Datos 2
- 1.6. Ciclo de estudios : 3
- 1.7. Duración del curso : 18 semanas
 - Inicio : 17 de Marzo de 2008
 - Término : 19 de Julio de 2008
- 1.8. Extensión horaria : 4 horas semanales (02 Teoría, 02 Laboratorio)
- 1.9. Créditos : 04
- 1.10. Período lectivo : 2008-2
- 1.11. Docente responsable: César Liza Avila
 - * cla@upnorte.edu.pe
 - creadores@hotmail.com
- 1.12. Página web : www.geocities.com/cesar_liza
www.cesarliza.com

2. FUNDAMENTACION

El curso de Análisis y Diseño de Algoritmos, son una pieza clave para las ciencias de la computación, ya que la eficiencia de un software depende de lo eficiente que sea la implementación del algoritmo. El conocimiento del análisis y diseño de algoritmos permite un buen entendimiento de la naturaleza del problema, así como el estudio de su complejidad y recursos necesarios para el buen funcionamiento. En este curso también mostramos un conjunto de técnicas independientes del lenguaje, paradigma de programación y hardware, que nos permitirá afrontar con mayores posibilidades de éxito la solución de un problema a ser resuelto con computadoras.

3. COMPETENCIA

Al concluir el curso los alumnos estarán en capacidad analizar y diseñar algoritmos, esto implica calcular su complejidad, y utilizar un conjunto de técnicas como recursión y backtracking, algoritmos voraces, divide y vencerás, fuerza bruta, programación dinámica, entre otras.

4. OBJETIVOS DEL CURSO

Al finalizar el curso el alumno deberá ser capaz de:

- 4.1 Ser capaz de analizar el costo de un algoritmo, en particular planteando y resolviendo recurrencias, para los tipos más comunes.
- 4.2 Comprender el concepto de complejidad de un problema como cota inferior, y conocer técnicas elementales para demostrar cotas inferiores.
- 4.3 Comprender las técnicas de diseño de algoritmos basadas en inducción, dividir para reinar, y programación dinámica, voraz, aleatorios.
- 4.4 Comprender el concepto de análisis amortizado de algoritmos. Ser capaz de diseñar algoritmos y estructuras de datos considerando costo amortizado, y ser capaz de analizar este costo.
- 4.5 Ser capaz de diseñar algoritmos eficientes usando la finitud del dominio. Poder distinguir estas situaciones.

5. CONTENIDOS CONCEPTUALES

- 5.1 Análisis de Algoritmos
- 5.2 Algoritmos Voraces
- 5.3 Divide y conquista
- 5.4 Algoritmos recursivos y Backtracking
- 5.5 Algoritmos exhaustivos o fuerza bruta
- 5.6 Programación dinámica
- 5.7 Algoritmos probabilistas o aleatorios
- 5.8 Análisis Amortizado
- 5.9 Algoritmos para grafos
- 5.10 Árboles abarcadores mínimos
- 5.11 Digrafos etiquetados y Flujo máximo
- 5.12 Geometría computacional
- 5.13 Problemas NP-completos:
- 5.14 Algoritmos paralelos

6. CONTENIDOS PROCEDIMENTALES

- 6.1 Conoce e investiga los conceptos de análisis, cálculo de complejidad y diversas técnicas de diseño de algoritmos.
- 6.2 Realizan y proponen ejemplos utilizando las técnicas vistas en clase con su respectivo cálculo de complejidad.

7. CONTENIDOS ACTITUDINALES

- 7.1 Responsabilidad Individual y Colectiva
- 7.2 Disposición a la Investigación y a la búsqueda de Información Adicional
- 7.3 Actitud analítica y lógica en la solución de los problemas propuestos
- 7.4 Disposición al trabajo en equipo
- 7.5 Disposición a ser reflexivos y creativos

8. METODOLOGIA GENERAL DEL CURSO

Las principales estrategias técnicas son:

- 8.1 Intervenciones en clase, como medio para que el alumno aplique e investigue los conocimientos recibidos.
- 8.2 Se tomarán prácticas calificadas, pruebas individuales en clase, como medio para que el alumno aplique los conocimientos recibidos en clase, como entrenamiento para el examen y como medio para ir midiendo sus logros académicos de manera continua.
- 8.3 Práctica calificada individual en Laboratorio, como medio para que el alumno aplique los conocimientos recibidos en clase, como entrenamiento para el examen y como medio para ir midiendo sus logros académicos de manera continua. En esta práctica, recibirá la atención del docente, como facilitador y guía de la práctica.
- 8.4 Desarrollo y Exposición de trabajos, como medio para que los alumnos aplique los conocimientos recibidos en clase, así como encontrar la madurez para aplicar estos conocimientos a un caso real. En estos trabajos recibirá la asesoría y guía del docente.

9. PROGRAMACION

Unidad	Objetivos	Sem	Contenidos
1. Análisis de Algoritmos	- Dar a conocer los conceptos de complejidad algorítmica y como calcularla. - Calcular la complejidad de los algoritmos de ordenamiento y búsqueda.	1 18/08	Complejidad, funciones básicas, reglas de simplificación, reglas para el cálculo de complejidad. Análisis asintótico, Notación O grande, omega y theta.
		2 25/08	Análisis de algoritmos de búsqueda, lineal, lineal acotada, lineal recursiva. Búsqueda binaria iterativa, búsqueda binaria recursiva. Análisis algoritmos de Ordenamiento por intercambio, burbuja, inserción, selección, rápida, montículo, mezcla.
	Referencias Bibliográficas: [1] Capítulo 4, [2] Capítulo 8 y 9, [3] Capítulo 2 y 6, [4] Capítulo 2, [5] Capítulo 6		

2. Algoritmos Voraces	. Mostrar la técnica Voraz así como calcular su complejidad.	3 01/09	Definición, forma general. Ejemplos: Cambio de moneda, factores primos, ruta entre dos nodos, árbol de recubrimiento mínimo, el problema de la mochila.
Referencias Bibliográficas: [1] Capítulo 6, [2] Capítulo 10, [6] Capítulo 8			
3. Divide y conquista	- Permitir que el alumno conozca y aplique la estrategia dividir y vencer, así como mostrar como calcular su complejidad.	4 08/09	Definición, forma general. Ejemplos: Ordenamientos, Búsqueda en conjuntos ordenados, multiplicación de enteros, mediana y moda de un vector, permutaciones, potencia, MCD.
Referencias Bibliográficas: [1] Capítulo 7, [2] Capítulo 10, [6] Capítulo 8			
4. Algoritmos recursivos y Backtracking	- Lograr que el alumno utilice la técnica de "vuelta atrás" así como que sea capaz de calcular su complejidad.	5 15/09	Recursión y Backtracking, problemas que pueden ser resueltos con backtraking, esquema general, ventajas, desventajas. Ejemplos diversos: Permutaciones, variaciones, Laberintos, Sudoku, Torres de Hanoi, 8 reinas.
Referencias Bibliográficas: [1] Capítulo 9			
5. Algoritmos exhaustivos o fuerza bruta	- Que el alumno identifique que es y cuando utilizar la "fuerza bruta", así como que logre el cálculo de su complejidad.	6 22/09	Definición, aplicaciones, cuando usarla. Backtracking y fuerza bruta. Recorrido en anchura y profundidad en árboles. El ataque por fuerza bruta. Ejemplos: Permutaciones El salto del caballo, las ocho reinas, ruta crítica, el problema de la mochila.
Referencias Bibliográficas: [5] Capítulo 8, [6] Capítulo 8			
6. Programación dinámica	- Comprender la importancia de la Programación dinámica. - Identificar las condiciones para usarla así como calcular su complejidad.	7 29/09	Definición, forma general, condiciones, la programación dinámica como problema de optimización, el principio de optimalidad de Bellman. Ejemplos: Factorial de un número, ruta crítica, Fibonacci, números primos, cambio de monedas, el problema de las vacas.
Referencias Bibliográficas: [1] Capítulo 8, [2] Capítulo 10, [6] Capítulo 8			
7. Algoritmos probabilistas o aleatorios	- Comprender y utilizar la aleatoriedad para resolver problemas, medir la complejidad de algoritmos aleatorios.	8 06/10	Tiempo esperado frente a tiempo promedio, algoritmos probabilistas numéricos, algoritmos de Monte Carlo, algoritmo las vegas. Ejemplos: Integración, comprobación de primalidad, factorización de enteros grandes.
Referencias Bibliográficas: [1] Capítulo 10			
		9 13/10	EXAMEN MEDIO CICLO
8. Análisis Amortizado	- Comprender el concepto de análisis amortizado de algoritmos y que el alumno sea capaz de diseñar algoritmos y estructuras de datos considerando costo amortizado.	10 20/10	Definición, Método agregado. Método del contador. Método del potencial. Ejemplos.
Referencias Bibliográficas: [1] Capítulo 4, [4] Capítulo 10			
9. Algoritmos para grafos	- Comprender y calcular la complejidad de los algoritmos para grafos.	11 27/10	Definiciones, especificación, representación, relaciones, dígrafos, dígrafos acíclicos, búsqueda en amplitud, búsqueda en profundidad, ordenamiento topológico y conectividad. Complejidad de algoritmos sobre grafos.
Referencias Bibliográficas: [3] Capítulo 17, 18			

10. Árboles abarcadores mínimos	- Calcular la complejidad de los algoritmos mas conocidos para la obtención del árbol abarcador mínimo	12 03/11	Algoritmos voraces para encontrar el arbol abarcador mínimo. TAD Conjuntos Disjuntos. Algoritmo de Kruskal. TAD Colas de Prioridad. TAD Montículos Binomiales. TAD Montículos de Fibonacci. Algoritmo de Prim.
	Referencias Bibliográficas: [3] Capitulo 18		
11. Digrafos etiquetados y Flujo máximo	- Calcular la complejidad de los algoritmos sobre dígrafos etiquetados y flujo máximo.	13 10/11	Caminos más cortos y relajación, algoritmo de Dijkstra, algoritmo de Bellman-Ford, restricciones y caminos más cortos (programación lineal), algoritmo de Floyd-Warshall y algoritmo de Johnson. Redes de flujo y método de Ford-Fulkerson.
	Referencias Bibliográficas: [3] Capitulo 18		
12. Geometría computacion al	- Desarrollar las habilidades para construir algoritmos de tipo geométrico y su implementación, así como calcular su complejidad	14 17/11	Segmentos de línea, propiedades, intersecciones, algoritmos de búsqueda convex hull. Cota inferior Omega(NLog N), par de puntos mas cercano, capas convexas o máximas.
	Referencias Bibliográficas: [7] pagina web		
13. Problemas NP-completos:	- Comprender la importancia de los problemas NP y por que son difíciles de resolver y como aproximar una solución.	15 24/11	Clases P y NP, reducciones polinomiales, problemas NP, algunas pruebas de completitud NP, algoritmos no determinísticos y problemas NP duros.
	Referencias Bibliográficas: [1] Capitulo 12, [6] Capitulo 10		
14. Algoritmos paralelos	- Introducir al alumno en el concepto de programación paralela, así como mostrarle como calcular la complejidad de sus algoritmos.	16 01/12	Computación paralela, procesamiento paralelo. Modelo PRAM. Algoritmos PRAM. teorema de Brent y el trabajo eficiente Relación entre modelo PRAM y la teoría de la complejidad. Problemas P-Completo. Diseño de algoritmos paralelos, estrategias. Algoritmos paralelos para máquinas MIMD.
	Referencias Bibliográficas: [1] Capitulo 11		
		17 08/12	EXAMEN FIN DE CICLO
		18 15/12	EXAMEN SUSTITUTORIO

10. SISTEMA DE EVALUACIÓN DEL CURSO

El cronograma de la evaluación continua del curso es el siguiente:

ESPECIFICACIÓN DE NOTAS DE TRABAJOS			
T	Descripción	Sem.	Peso
T1	Objetivos de la evaluación: <ol style="list-style-type: none"> Permitir que el estudiante aplique los conceptos sobre análisis de algoritmos vistos en clase. Medir el entendimiento de la técnica voraz. 	1 2 3	10%
	Descripción: Práctica de laboratorio 1 : Análisis de Algoritmos 1 (15 pts) Práctica de laboratorio 2 : Análisis de Algoritmos 2 (15 pts) Práctica de laboratorio 3 : Algoritmos Voraces (15 pts) $T1 = \frac{Lab1 + Lab2 + Lab3}{2}$		

T2	Objetivos de la evaluación: 1. Aplicar las técnicas de divide y conquista, recursión y backtracking, fuerza bruta.	4 5 6	15%
	Descripción: Práctica de laboratorio 4 : Divide y conquista (15 pts) Práctica de laboratorio 5 : Algoritmos recursivos y Backtracking (15 pts) Práctica de laboratorio 6 : Algoritmos exhaustivos o fuerza bruta (15 pts) $T2 = \frac{Lab4 + Lab5 + Lab6}{2}$		
T3	Objetivos de la evaluación: 1. Medir el grado de comprensión de las técnicas Programación dinámica, algoritmos aleatorios. 2. Medir el grado de comprensión y uso adecuado del análisis amortizado.	7 8 10	20%
	Descripción: Práctica de laboratorio 7 : Programación dinámica (15 pts) Práctica de laboratorio 8 : Algoritmos probabilistas o aleatorios (15 pts) Práctica de laboratorio 10 : Análisis Amortizado (15 pts) $T3 = \frac{Lab7 + Lab8 + Lab10}{2}$		
T4	Objetivos de la evaluación: 1. Consolidar el uso de la estructura de datos grafos, así como medir la complejidad de sus algoritmos.	11 12 13	25%
	Descripción: Práctica de laboratorio 11 : Algoritmos para grafos (15 pts) Práctica de laboratorio 12 : Árboles abarcadores mínimos (15 pts) Práctica de laboratorio 13: Digrafos etiquetados y Flujo máximo (15 pts) $T4 = \frac{Lab11 + Lab12 + Lab13}{2}$		
T5	Objetivos de la evaluación: 1. Afianzar los conceptos y usos de la Geometría Computacional 2. Medir el grado de comprensión de problemas NP completos y su importancia. 3. Medir la comprensión y uso de los algoritmos paralelos y el calculo de su complejidad	14 15 16	30%
	Descripción: Práctica de laboratorio 14: Geometría computacional (15 pts) Práctica de laboratorio 15: Problemas NP-completos (15 pts) Práctica de laboratorio 16: Algoritmos paralelos (15 pts) $T5 = \frac{Lab14 + Lab15 + Lab16}{2}$		

(*) La práctica del Laboratorio 9, es parte del examen de medio ciclo

El peso de cada T es:

EVALUACIÓN	PESO (%)	ESCALA VIGESIMAL
T01	10	1,2
T02	15	1,8
T03	20	2,4
T04	25	3,0
T05	30	3,6
TOTAL	100%	12

El promedio de Trabajos (Evaluación continua) se calcula así:

$$T = (0.10)T1 + (0.15)T2 + (0.20)T3 + (0.25)T4 + (0.30)T5$$

Los pesos ponderados de las clases de evaluación son los siguientes:

EVALUACIÓN	PESO (%)	ESCALA VIGESIMAL
PARCIAL	20	4
CONTINUA	60	12
FINAL	20	4
TOTAL	100%	20

El promedio del curso se calcula de acuerdo con la siguiente fórmula:

$$\text{Promedio} = (0.6) T + EMC (0.2) + EFC (0.2)$$

La Evaluación Sustitutoria evalúa toda la temática desarrollada en el semestre y se rinde la semana consecutiva al término de los exámenes finales (**14 de Julio**) y su nota reemplazará, necesariamente, a la nota de un Examen (Parcial o Final) o a la nota de un T (Evaluación Continua), de tal manera que el resultado final sea favorable al alumno.

11. NORMAS ADICIONALES

- Es obligatoria la asistencia a las clases teóricas y prácticas programadas (70%). El alumno que no cumpla con este requisito quedará inhabilitado en el curso. El alumno que no esté presente al llamado de lista será considerado ausente. El cómputo de la asistencia se realiza desde el primer día de clases.
- No es posible la recuperación de ninguna nota parcial de la Evaluación Continua, bajo ningún concepto.
- Está **prohibido el uso de celular en clase**. Los alumnos cuyos celulares timbren en horario de clase serán penalizados con un descuento de **5 puntos**, si contestan el celular en clase serán **penalizados con 5 puntos adicionales**.
- Los alumnos que no guarden la debida compostura y respeto hacia sus compañeros o hacia su profesor serán castigados con el descuento de algunos puntos según la gravedad de la falta.
- Los alumnos que llamen por teléfono o toquen la puerta de la casa del docente no serán atendidos y tendrán una penalización de **5 puntos**. Toda consulta o reclamo debe realizarse en las **instalaciones de la Universidad o vía correo electrónico**.
- Los exámenes y prácticas deben rendirse en hojas grandes **A4** u oficio, deben indicar claramente el **tema**, el **nombre del alumno**, el **número o enunciado** de la pregunta. En caso de rendir en hojas individuales éstas deben estar correctamente engrapadas. Se descontará **2 puntos** por cada falta en alguno de estos puntos. El alumno puede obtener una calificación negativa la cual se podrá descontar en otra prueba.
- Está prohibido terminantemente usar fotocopias de los libros del docente, las cuales serán decomisadas sin derecho a reclamo.
- Excepcionalmente, los alumnos pueden entrar y salir del aula cuando lo estimen conveniente sin necesidad de pedir permiso al docente, siempre que lo hicieran con la mayor discreción y sin interrumpir el desenvolvimiento de la clase.

12. BIBLIOGRAFÍA

BÁSICA

Nro	CÓDIGO	AUTOR	TÍTULO
1	005.1/B81	Brassard, G.; Bratley, P	Fundamentos de Algoritmia. Prentice Hall, Madrid 1997.
2	005.73/A32	Aho, Alfred; Hopcroft, Jhon; Ullman, Jeffrey	Estructuras de datos y algoritmos. Addison Wesley Iberoamericana. 1988.
3		JOYANES, Luis; ZAHONERO, Ignacio.	Algoritmos y Estructura de Datos: Una perspectiva en C. Ed. McGraw-Hill, 1ra Edición, Madrid 2004.
4		Heileman, Gregory	Estructuras de Datos, Algoritmos y Programación Orientada a Objetos. Mc Graw Hill Interamericana de España, 1998
5	005.1/G19	Galve, Javier; Gonzales, Juan, Sanchez, Angel; Velásquez, Angel	Algoritmica: diseño y análisis de algoritmos funcionales e imperativos. Editorial RAMA- Addison Wesley. USA 1993.
6	Internet	CESAR LIZA AVILA.	www.geocities.com/cesar_liza y www.cesarliza.com
7	Internet		http://www.dma.fi.upm.es/docencia/segundociclo/geomco/mp/aplicaciones.html
8	Internet		http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060024/index.html

COMPLEMENTARIA

Nro	Código	Autor	Obra
9		Thomas H. Cormen	Introduction to Algorithms, Second Edition. MIT Press. ISBN 0262032937
10		Steven S. Skiena	The Algoritm Design Manual. Editorial Springer. ISBN 0387948600
11		Steven S. Skiena; Miguel A. Revilla	Programming Challenges. Editorial Springer. ISBN 0387001638
12		Steven S. Skiena; Miguel A. Revilla	CONCURSOS INTERNACIONALES DE INFORMÁTICA Y PROGRAMACIÓN MANUAL DE ENTRENAMIENTO POR INTERNET EDITA: Secretariado de Publicaciones e Intercambio. Editorial. Universidad de Valladolid. AÑO: 2006. FORMATO: 17x24 cms. 393 págs. ISBN: 84-8448-371-1