Universidad Nacional de San Agustín
VICE RECTORADO ACADÉMICO
SILABO

CODIGO DEL CURSO: CS210T

1 Datos Generales

FACULTAD : Ingeniería de Producción y Servicios						
DEPARTAMENTO : Ingeniería de Sistemas e Informática			**ESCUELA :** Ciencia de la Computación			
PROFESOR : **TÍTULO :**						
ASIGNATURA : Análisis y Diseño de Algoritmos						
PREREQUISITO: CS103O,CB203	**CREDITOS:** 4		**Año**: 2010-1 **Sem:** 5^{to} Semestre.		**Total Horas:** 2 HT; 2 HP 2 HL	
Horario	Lun	Mar	Mie	Jue	Vie	Sáb
Total Semanal						
Aula						

2 Exposición de Motivos

Los algoritmos son pieza clave para la ciencia de la computación. El rendimiento de sistema de so
de dos cosas: a) La búsqueda de algoritmos y b) La eficiencia conveniente de varias capas de impl
El diseño de buenos algoritmos es por otra parte crucial para el buen funcionamiento de todo s
Más aun, el estudio de algoritmos provee el buen entendimiento de la naturaleza del problema
técnicas independientes para la posible solución, independientemente de un lenguaje de programa
programación, hardware de computador o cualquier otro aspecto de implementación.(Computing
& ACM).

2 Objetivo

- Permitir que el alumno pueda realizar el análisis y diseñ de algoritmos eficientes para la solución de proble complejos.
- Proveer al alumno de una serie de técnicas, de anáisis y diseño para la evaluación e implementación de algorit

3 Contenido Temático

3 AL/Análisis Básico de Algoritmos.(12 horas)

Objetivos Específicos	Co
▪ Explicar el uso de las notaciones Big O, *Omega* Ω y *Theta* Θ para describir la cantidad de trabajo hecha por un algoritmo. ▪ Uso de notaciones Big O, *Omega* Ω y *Theta* Θ para determinar los límites asintóticos superior, inferior y el más próximo en tiempo y espacio en complejidad de algoritmos . ▪ Determinar la complejidad de tiempo y espacio de algoritmos simples. ▪ Deducir la relación de recurrencia que describe la complejidad de tiempo de algoritmos definidos recursivamente. ▪ Solucionar relaciones de recurrencia elemental.	[4]

	Objetivos Específicos	Contenidos
3 AL/Algoritmos Fundamentales.(16 horas)	▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento $O(NlogN)$. ▪ Diseñar e implementar una función de (*hash*) apropiada para una aplicación. ▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de *hash*. ▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (*hashing*). ▪ Discutir otros factores, además de la eficiencia computacional, que influyen en la elección de los algoritmos, tales como tiempo de programación, mantenimiento y el uso de patrones específicos de aplicación en los datos de entrada. ▪ Resolver problemas usando los algoritmos de grafos fundamentales, incluyendo búsqueda por amplitud y profundidad; caminos más cortos con uno y múltiples orígenes, cerradura transitiva, ordenamiento topológico y al menos un algoritmo de árbol de expansión mínima. ▪ Demostrar las siguientes capacidades: evaluar algoritmos, seleccionar una opción de un rango posible, proveer una justificación para tal elección e implementar el algoritmo..	▪ Algoritmos numéricos sim ▪ Búsqueda secuencial y bin ▪ Algoritmos cuadráticos d miento (selección, inserció ▪ Algoritmos de tipo *C* (Quicksort, heapsort, merg ▪ Tablas de (*hash*) incluyend gias de solución para las c ▪ Árboles de búsqueda bina ▪ Representación de grafos Matrices de adyacencia). ▪ Recorridos por amplitud didad. ▪ El algoritmo del camino (algoritmos de Dijkstra y ▪ Cerradura transitiva (alg Floyd). ▪ Árbol de expansión mínima mos de Kruskal y Prim). ▪ Ordenamiento Topológico. [4], [2], [1]

3 AL/Estrategias Algorítmicas.(24 horas)

Objetivos Específicos	Contenidos
▪ Describir las desventajas de los algoritmos de fuerza bruta. ▪ Para cada una de las diferentes clases de algoritmos (fuerza bruta, voraces, dividir y conquistar, *Backtracking*, *Branch-and-bound* y heurísticos), identificar un ejemplo del comportamiento humano cotidiano que ejemplifique el concepto básico. ▪ Implementar un algoritmo voraz para resolver apropiadamente un problema. ▪ Implementar un algoritmo de divide y vencerás para solucionar apropiadamente un problema. ▪ Utilizar *Backtracking* para solucionar problemas tal como el de navegación en un laberinto. ▪ Describir varios métodos de solución de problemas heurísticos. ▪ Utilizar emparejamiento de patrones para analizar subcadenas. ▪ Utilizar aproximación numérica para resolver problemas matemáticos, tal como el de encontrar las raíces de un polinomio.	▪ Algoritmos de fuerza bruta *force*). ▪ Algoritmos voraces (*greedy*). ▪ Divide y vencerás. ▪ *Backtracking*. ▪ *Branch-and-bound*. ▪ Heurísticos. ▪ Emparejamiento de patrones ritmos de cadenas/texto. ▪ Algoritmos de aproximación rica. [4], [2], [1]

3 AL/Algoritmos Distribuidos.(4 horas)

Objetivos Específicos	Contenidos
▪ Explicar el paradigma distribuido. ▪ Explicar un algoritmo distribuido simple. ▪ Determinar cuando usar los algoritmos de consenso o elección. ▪ Distinguir entre relojes físicos y lógicos. ▪ Describir el ordenamiento relativo de eventos en un algoritmo distribuido.	▪ Consenso y elección. ▪ Detección de finalización. ▪ Tolerancia a fallas. ▪ Estabilización. [4], [2], [1]

	Objetivos Específicos	Contenidos
3 AL/Clases de Complejidad P y NP.(4 horas)	▪ Definir las clases P y NP. ▪ Explicar el significado de la NP-Completitud. ▪ Probar que un problema es NP-completo reducciendo un problema NP-Completo clásico conocido a éste.	▪ Definición de las clases ▪ NP-completitud (El teo ok). ▪ Problemas NP-complet res. ▪ Técnicas de reducción. [4], [2], [1]

4 Actividades

- Asignaciones
- Controles de Lectura
- Exposiciones

5 Recursos Materiales

- Apuntes del curso
- Libro(s) de la bibliografía

6 Metodología

- Clase Magistral.
- Taller didáctico.
- Social Constructivismo.
- Prácticas personales y en grupo.

7 Evaluación

La nota final (NF) se obtiene de la siguiente manera:

NE Nota de Exámenes 60 %, esta nota se divide en
- Exámen Parcial 40 %
- Examen Final 60 %

NT Nota de Trabajos e Intervención en clase 40 %

$$NF = 0{,}6 * NE + 0{,}4 * NT$$

Referencias

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[2] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006.

[3] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison Wesley Iberoamericana, 1994.

[4] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005.

Docente del curso