

Universidad Nacional de San Agustín
VICE RECTORADO ACADÉMICO
SILABO

CODIGO DEL CURSO: CS1030

1 Datos Generales

FACULTAD : Ingeniería de Producción y Servicios								
DEPARTAMENTO : Ingeniería de Sistemas e Informática				ESCUELA : Ciencia de la Computación				
PROFESOR :								
TÍTULO :								
ASIGNATURA : Algoritmos y Estructuras de Datos								
PREREQUISITO: CS1020		CREDITOS: 4			Año: 2010-1		Total Horas: 2 HT;	
					Sem: 4 ^{to} Semestre.		2 HT 2 HP 2 HL	
Horario		Lun	Mar	Mie	Jue	Vie	Sáb	
Total Semanal								
Aula								

2 Exposición de Motivos

El fundamento teórico de todas las ramas de la informática descansa sobre los algoritmos y estructuras de datos. Este curso brindará a los participantes una introducción a estos temas, formando así una base que servirá para los cursos en la carrera.

2 Objetivo

- Hacer que el alumno entienda la importancia de los algoritmos para la solución de problemas.
- Introducir al alumno hacia el campo de la aplicación de las estructuras de datos.

3 Contenido Temático 3 PF/Estructuras de Datos.(8 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> ▪ Describir la representación de datos numéricos y de caracteres ▪ Entender como la precisión y el redondeo puede afectar los cálculos numéricos. ▪ Discutir la representación y uso de tipos de datos primitivos y estructuras de datos incorporadas en el lenguaje. ▪ Describir aplicaciones comunes para cada estructura de datos en la lista de temas. ▪ Implementar estructuras de datos definidas por el usuario en un lenguaje de alto nivel. ▪ Comparar implementaciones alternativas de estructuras de datos considerando su desempeño. ▪ Escribir programas que usan cada una de las siguientes estructuras de datos: arreglos, registros, cadenas, listas enlazadas, pilas, colas y tablas de <i>hash</i>. ▪ Comparar y contrastar los costos y beneficios de las implementaciones dinámicas y estáticas de las estructuras de datos. ▪ Escoger la estructura de datos apropiada para modelar un problema dado. 	<ul style="list-style-type: none"> ▪ Represe ▪ Rango, s deo. ▪ Arreglos ▪ Registro ▪ Cadenas nas. ▪ Represe ▪ Adminis en tiempo ▪ Puntero ▪ Estructu ▪ Estrateg pilas, co ▪ Estrateg grafos y ▪ Estrateg ra de da <p>[1], [2]</p>

3 PF/Recursividad.(4 horas)

Objetivos Específicos	Contenidos	Horas
<ul style="list-style-type: none"> ▪ Describir el concepto de recursividad y dar ejemplos de su uso. ▪ Identificar el caso base y el caso general de un problema definido recursivamente. ▪ Comparar soluciones iterativas y recursivas para problemas elementales tal como factorial. ▪ Describir la técnica dividir y conquistar. ▪ Implementar, probar y depurar funciones y procedimientos recursivos simples. ▪ Describir como la recursividad puede ser implementada usando una pila. ▪ Discutir problemas para los cuales el <i>backtracking</i> es una solución apropiada. ▪ Determinar cuando una solución recursiva es apropiada para un problema. 	<ul style="list-style-type: none"> ▪ El concepto de recursividad. ▪ Funciones matemáticas recursivas. ▪ Funciones recursivas simples. ▪ Estrategias de dividir y conquistar. ▪ <i>Backtracking</i> recursivo. <p>[1], [2]</p>	

3 AL/Algoritmos Fundamentales.(12 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> ▪ Implementar los algoritmos cuadráticos más comunes y los algoritmos de ordenamiento $O(N\log N)$. ▪ Diseñar e implementar una función de (<i>hash</i>) apropiada para una aplicación. ▪ Diseñar e implementar un algoritmo de resolución de colisiones para tablas de <i>hash</i>. ▪ Discutir la eficiencia computacional de los principales algoritmos de ordenamiento, búsqueda y (<i>hashing</i>). ▪ 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.. 	<ul style="list-style-type: none"> ▪ Algoritmos numéricos simples. ▪ Búsqueda secuencial y binaria. ▪ Algoritmos cuadráticos de ordenamiento (selección, inserción). ▪ Algoritmos de tipo $O(N^2)$ (Quicksort, heapsort, mergesort). ▪ Tablas de (<i>hash</i>) incluyendo estrategias de solución para las colisiones. ▪ Árboles de búsqueda binaria. ▪ Representación de grafos (Matrices de adyacencia). ▪ Recorridos por amplitud y profundidad. ▪ El algoritmo del camino más corto (algoritmos de Dijkstra y Floyd). ▪ Cerradura transitiva (algoritmo de Floyd). ▪ Árbol de expansión mínima (algoritmos de Kruskal y Prim). ▪ Ordenamiento Topológico. <p>[1], [2]</p>

Objetivos Específicos	Contenidos	Horas	Fecha
<p>3 Grafos (12 horas)</p> <ul style="list-style-type: none"> Adquirir destreza para realizar una implementación correcta. Desarrollar los conocimientos para decidir cuando es mejor usar una técnica de implementación que otra. 	<ul style="list-style-type: none"> Concepto de Grafos. Grafos Dirigidos y Grafos no Dirigidos. Utilización de los Grafos. Medida de la Eficiencia. En tiempo y espacio. Matrices de Adyacencia. Matrices de Adyacencia etiquetada. Listas de Adyacencia. Implementación de Grafos usando Matrices de Adyacencia. Implementación de Grafos usando Listas de Adyacencia. Inserción, Búsqueda y Eliminación de nodos y aristas. Algoritmos de búsqueda en grafos. <p>[1], [2]</p>		

Objetivos Específicos	Contenidos	Horas
<p>3 Matrices Esparzas (8 horas)</p> <ul style="list-style-type: none"> Comprender el uso y implementación de matrices esparzas. 	<ul style="list-style-type: none"> Conceptos Iniciales. Matrices poco densas Medida de la Eficiencia en Tiempo y en Espacio Creación de la matriz esparza estática vs Dinámicas. Métodos de inserción, búsqueda y eliminación <p>[1], [2]</p>	

Objetivos Específicos	Contenidos
<p>3 Árboles Equilibrados (16 horas)</p> <ul style="list-style-type: none"> Comprender las funciones básicas de estas estructuras complejas con el fin de adquirir la capacidad para su implementación. 	<ul style="list-style-type: none"> Árboles AVL. Medida de la Eficiencia. Rotaciones Simples y Compuestas Inserción, Eliminación y Búsqueda. Árboles B, B+, B* y Patricia. <p>[1], [2]</p>

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*. MIT Press, third edition edition, 2009. ISBN: 978-0-262-53305-8.
- [2] José Fager, W. Libardo Pantoja Yépez, Marisol Villacrés, Luz Andrea Páez Martínez, Daniel Ochoa, and Ernesto Cuadros-Vargas. *Estructura de datos*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), first edition edition, 2014.

Docente del curso