

Nombre de la asignatura: **Matemáticas Discretas**  
 Línea de trabajo: Modelación Inteligente de Sistemas, PDI y Visión Artificial  
 Tiempo de dedicación del estudiante a las actividades de  
 DOC - TIS - TPS - Horas totales. Créditos  
 48-20-100-168-6

DOC: Docencia; TIS: Trabajo independiente significativo; TPS: Trabajo profesional supervisado

**1. Historial de la asignatura.**

Fecha revisión / actualización	Participantes	Observaciones, cambios o justificación
9 de mayo de 2011	Bernabé Ortiz y Herbert	Materia de tronco común

**2. Pre-requisitos y co-requisitos.**

Ninguno

**3. Objetivo de la asignatura.**

Conocer, comprender y aplicar conceptos básicos de la matemática discreta en el estudio de modelos teóricos de la computación.

**4. Aportación al perfil del graduado.**

Esta asignatura aporta al perfil del egresado los conocimientos matemáticos para entender, inferir, aplicar y desarrollar modelos matemáticos tendientes a resolver problemas en el área de las ciencias computacionales. Esta materia es el soporte para un conjunto de áreas computacionales como análisis y diseño de algoritmos, bases de datos, inteligencia artificial, redes de computadoras, etcétera.

**5. Contenido temático.**

Unidad	Temas	Subtemas
I	Lógica proposicional.	1.1. Introducción. 1.2. Argumentos lógicos y proposiciones. 1.2.1. Introducción. 1.2.2. Argumentos lógicos. 1.2.3. Proposiciones. 1.3. Conectivos lógicos. 1.3.1. Introducción. 1.3.2. Negación. 1.3.3. Conjunción. 1.3.4. Disyunción. 1.3.5. Condicional. 1.3.6. Bicondicional. 1.4. Proposiciones compuestas. 1.4.1. Introducción. 1.4.2. Expresiones lógicas.

		<ul style="list-style-type: none"> <li>1.4.3. Análisis de proposiciones compuestas.</li> <li>1.4.4. Reglas de precedencia.</li> <li>1.4.5. Evaluación de expresiones y tablas de verdad.</li> <li>1.5. Tautologías y contradicciones. <ul style="list-style-type: none"> <li>1.5.1. Introducción.</li> <li>1.5.2. Tautologías.</li> <li>1.5.3. Contradicciones.</li> </ul> </li> <li>1.6. Equivalencias lógicas. <ul style="list-style-type: none"> <li>1.6.1. Introducción.</li> <li>1.6.2. Demostración de equivalencias lógicas con tablas de verdad.</li> <li>1.6.3. Algebra proposicional.</li> <li>1.6.4. Leyes esenciales del algebra proposicional.</li> <li>1.6.5. Formas normales.</li> </ul> </li> <li>1.7. Reglas de inferencia <ul style="list-style-type: none"> <li>1.7.1. Argumentos válidos y no validos.</li> <li>1.7.2. Demostración formal directa.</li> <li>1.7.3. Demostración por contradicción.</li> </ul> </li> </ul>
<b>II</b>	Conjuntos, relaciones y funciones.	<ul style="list-style-type: none"> <li>2.1. Introducción.</li> <li>2.2. Conjuntos <ul style="list-style-type: none"> <li>2.2.1. Operaciones sobre conjuntos.</li> <li>2.2.2. Conjunto potencia.</li> </ul> </li> <li>2.3. Relaciones. <ul style="list-style-type: none"> <li>2.3.1. Pares ordenados y producto cartesiano.</li> <li>2.3.2. Relaciones y su representación.</li> <li>2.3.3. Dominio y rango de una relación.</li> <li>2.3.4. Composición de relaciones.</li> <li>2.3.5. Relaciones reflexivas.</li> <li>2.3.6. Relaciones simétricas.</li> <li>2.3.7. Relaciones transitivas.</li> <li>2.3.8. Cerradura.</li> <li>2.3.9. Relaciones de equivalencia.</li> </ul> </li> <li>2.4. Funciones. <ul style="list-style-type: none"> <li>2.4.1. Definición y notación.</li> <li>2.4.2. Composición de funciones.</li> <li>2.4.3. Funciones inyectivas, sobreyectivas y biyectivas.</li> <li>2.4.4. Funciones inversas.</li> <li>2.4.5. Función característica.</li> </ul> </li> </ul>
<b>III</b>	Relaciones de recurrencia.	<ul style="list-style-type: none"> <li>3.1 Introducción</li> <li>3.2 Relaciones de recurrencia homogénea.</li> <li>3.3 Relaciones de recurrencia no homogénea.</li> </ul>
<b>IV</b>	Complejidad de algoritmos.	<ul style="list-style-type: none"> <li>4.1 Introducción.</li> <li>4.2 Algoritmo con tiempo polinomial.</li> <li>4.3 Algoritmos con tiempo exponencial.</li> <li>4.4 Los límites de la computabilidad.</li> <li>4.5 Análisis asintótico.</li> <li>4.6 Divide y conquistas.</li> </ul>

		4.7 Tiempo polinomial no determinístico.
V	Técnicas de conteo	5.1 Introducción. 5.2 Principio de multiplicación de conteo. 5.3 Ordenaciones con repetición. 5.4 Ordenaciones sin repetición 5.5 Permutaciones. 5.6 Combinaciones.
VI	Grafos y árboles	6.1. Introducción. 6.2. Definiciones básicas de teoría de grafos. 6.3. Algoritmos de recorrido y búsqueda. 6.3.1. El camino más corto. 6.3.2. Recorrido a lo ancho. 6.3.3. Recorrido en profundidad. 6.4. Árboles. 6.4.1. Introducción. 6.4.2. Definiciones básicas de árboles. 6.4.3. Recorrido de árboles. 6.4.3.1. Recorrido preorden. 6.4.3.2. Recorrido enorden. 6.4.3.3. Recorrido postorden. 6.4.4. Árboles balanceados.

## 6. Metodología de desarrollo del curso.

- Proporcionar al estudiante una lista de problemas de cada tema y generar practicas de laboratorio para confrontar los resultados requeridos.
- Promover grupos de discusión para y análisis sobre los conceptos previamente investigados.
- Utilizar laboratorio de cómputo con acceso a internet y software para desarrollar las aplicaciones: Matlab, Prolog y visual studio .net
- Para cada tema seleccionar un caso de estudio.
- Discutir y determinar problemas de aplicación.

## 7. Sugerencias de evaluación.

- Análisis y discusión. El alumno debe reflexionar y opinar acerca del tema que se esté tratando en clase.
- Ejercicios resueltos por el alumno.
- Investigaciones. El alumno investigará acerca de un tema particular y entregará un reporte escrito del resultado de su investigación.
- Programas. El alumno desarrollará y entregará un programa para resolver un problema dado.
- Un proyecto final donde se apliquen los conocimientos adquiridos en el curso.

## 8. Bibliografía y Software de apoyo.

- Winfried Karl Grassmann, Jean-Paul Tremblay. Logic and Discrete Mathematics (A Computer Science Perspective). Prentice Hall. 1996.
- B. Kolman, R. C. Busby, S. Ross. *Estructuras de matemáticas discretas para la computación*. Prentice Hall, Hispanoamericana. 1997.
- Jean-Paul Tremblay, Ram Manohar. *Matemáticas Discretas (Con Aplicación a las Ciencias de la Computación)*. CECSA. 1996.

- Ralph P. Grimaldi. *Matemáticas Discreta y Combinatoria (Una introducción con aplicaciones)*. Prentice Hall. Tercera edición. 1998.
- Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall (Pearson). 4ta edición. 1999.
- Seymour Lipschutz, Marc Lipson. *Matemáticas Discretas*. Mc Graw Hill (Serie Schaum). 3ra edición. 2007.
- Nicklaus Wirth. *Algorithms + Data Structures = Programs*. Prentice Hall. 1976
- David Gries, Fred B. Schneider. *A Logical Approach to Discrete Math*. Springer (Texts and monographs in computer science). 1993.
- Prolog :programming for artificial intelligence. Bratko, Ivan.

## 9. Actividades propuestas.

Unidad	Práctica
1	Empleando MSExcel calcular el valor de verdad de una proposición lógica compuesta.
1	Empleando Matlab calcular el valor de verdad de una proposición lógica compuesta
1	Emplear el simulador que aparece en la página web <a href="http://sourceforge.net/projects/circuit/files/2.6.x/2.6.2/logisim-win-2.6.2.exe/download">http://sourceforge.net/projects/circuit/files/2.6.x/2.6.2/logisim-win-2.6.2.exe/download</a> para construir tablas de verdad.
2	Empleando Power Point representar el comportamiento de las operaciones con conjuntos mediante diagramas de Venn.
2	Creación de una base de datos inteligente mediante PROLOG para ejemplificar las propiedades de una relación
2	Emplear Matlab para representar las propiedades de una relación mediante matrices.
2	Emplear Matlab para ejemplificar la propiedad de cerradura de una relación y determinar el número de caminos y recorridos en un grafo.
6	Emplear Matlab para determinar si dos grafos son isomorfos.
6	Emplear Matlab para determinar el camino más corto en un grafo.
6	Emplear Matlab para realizar el recorrido de un árbol que represente una expresión matemática y obtener su valor usando para ello el concepto de pila para almacenar resultados.
6	Emplear Matlab para crear un árbol binario a partir de una lista de números aleatorios y llevar a cabo búsquedas y ordenamiento de dichos datos.
6	Emplear Matlab para simular el recorrido de árboles para la búsqueda de información.
6	Emplear Matlab para ejemplificar el algoritmo de Kruskal y Prim
6	Emplear Matlab para ejemplificar los códigos de Huffman
6	Emplear Matlab para ejemplificar el teorema de flujo máxima
6	Emplear un simulador para ejemplificar las redes de Petri