

4113 – MATEMÁTICA DISCRETA

I - Datos de identificación de la asignatura

Carrera:	Licenciatura en Análisis de Sistemas		
Código:	4113	Plan:	2024
Denominación:	Matemática discreta		
Área:	Matemáticas		
Año:	Primer		
Horas con acompañamiento docente (HTD), semanal			2
Horas de Trabajo Independiente del estudiante (HTI), semanal			2
Horas semanales (HS)			4
Cantidad de sesiones			32
Total Horas de Trabajo con el docente (THTD)			64
THD teóricas	64	THD prácticas	0
Total de Horas de Trabajo Independiente del estudiante (THTI)			64
Total Horas Académicas (THA)			128
Crédito académico (CA)			5,1
Pre-requisito:	Aprobar el Curso Probatorio de Ingreso.		

II - Fundamentación

La asignatura Matemática Discreta reviste una importancia fundamental para la formación de profesionales en informática y áreas afines. Las computadoras operan con información discreta (valores finitos y separados), por lo que resulta esencial manejar conceptos y métodos matemáticos adecuados para modelar y resolver problemas mediante algoritmos. Esta materia proporciona las bases teórico-prácticas para desarrollar el pensamiento lógico-deductivo, habilidad que el estudiante de Análisis de Sistemas utilizará a lo largo de toda la carrera y en su ejercicio profesional.

En particular, Matemática Discreta aporta conocimientos sobre estructuras matemáticas (conjuntos, relaciones, funciones, lógica, entre otras) que permiten representar formalmente situaciones del mundo real en términos comprensibles por una computadora. Asimismo, introduce al alumno en los principios de la lógica matemática y la programación declarativa, sentando las bases para cursos posteriores de algoritmia, diseño de sistemas, bases de datos y áreas de inteligencia artificial. De este modo, la asignatura combina elementos conceptuales y aplicados, proporcionando una base sólida de conocimientos y la capacidad de aplicarlos en la resolución de problemas informáticos de diversa índole.

III - Competencias a desarrollar

Competencias genéricas

1. Aplicar un conjunto específico de conocimientos científicos, matemáticos y tecnológicos a un problema del área informática, tomando en consideración restricciones físicas, económicas, ambientales, humanas, éticas, políticas, legales y culturales.
2. Identificar, plantear y resolver problemas.

Competencias específicas

1. Aplicar con precisión los principios de sintaxis y semántica de la lógica proposicional, para interpretar y estructurar proposiciones lógicas con base en su significado y validez formal.
2. Utilizar con solvencia las reglas de la lógica de predicados, para analizar fórmulas compuestas y comprender su interpretación lógica en distintos contextos.
3. Desarrollar con autonomía soluciones mediante el paradigma de Programación Lógica y Declarativa, implementando algoritmos en Prolog que reflejen principios fundamentales de lógica computacional.
4. Aplicar de manera sistemática los conceptos, propiedades y operaciones de la Teoría de Conjuntos, para analizar y resolver problemas a través del razonamiento lógico-formal.
5. Emplear con criterio lógico los fundamentos del Álgebra Booleana, para operar funciones lógicas, utilizar cuantificadores y realizar inferencias en el marco del razonamiento deductivo.
6. Establecer conexiones conceptuales claras entre el Álgebra de Conjuntos, la Lógica Proposicional y el Álgebra de Boole, para interpretar y optimizar circuitos lógicos aplicados en contextos de electrónica e informática.

IV - Cuerpo de conocimientos

Unidad 1: Álgebra de Conjuntos

Contenidos:

- Conjuntos: concepto y definición. Símbolos y notaciones para la representación de conjuntos (por extensión y por comprensión). Diagramas de Venn como apoyo visual.
- Relaciones entre conjuntos: subconjunto, igualdad de conjuntos, operaciones fundamentales (unión, intersección, diferencia, complemento).
- Propiedades del Álgebra de Conjuntos (leyes conmutativas, asociativas, distributivas, de De Morgan, etc.). Principio de dualidad en conjuntos. Conjunto producto (producto cartesiano de conjuntos) y su interpretación.

Unidad 2: Relaciones

Contenidos:

- Relaciones binarias: concepto de relación en matemáticas discretas. Dominio, rango (codominio) de una relación.
- Tipos de relaciones importantes: reflexividad, simetría, transitividad y relación de equivalencia (caracterización de una relación de equivalencia a través de sus propiedades). Relación inversa de una relación dada.
- Representaciones de relaciones: representación como pares ordenados, matrices de relación, digrafos (grafos dirigidos) para visualizar relaciones. Análisis de ejemplos prácticos de relaciones en informática (por ejemplo, relaciones de orden en datos, grafos de conexiones, etc.).

Unidad 3: Funciones

Contenidos:

- Funciones: definición formal como relación especial. Dominio y codominio. Imagen de un elemento y conjunto imagen de una función.

- Tipos de funciones: inyectivas, sobreyectivas y biyectivas (explicación y criterios para cada tipo). Función inversa (existencia y cálculo de la inversa cuando la función es biyectiva).
- Representaciones gráficas y tabulares de funciones discretas. Aplicaciones de funciones en informática (p.ej., funciones de hashing como aplicaciones parciales, funciones de conversión de datos, etc.).

Unidad 4: Álgebra de Proposiciones

Contenidos:

- Lógica proposicional: concepto de proposición. Conectivos lógicos básicos (negación, conjunción, disyunción, implicación, doble implicación) y construcción de fórmulas proposicionales. Tablas de verdad para evaluar fórmulas.
- Sintaxis y semántica de la lógica proposicional: formas normales (conjuntiva FNC y disyuntiva FND), tautologías, contradicciones, satisfacibilidad y equivalencias lógicas. Propiedades del Álgebra de Proposiciones (leyes lógicas análogas a las de conjuntos y Boole). Principio de dualidad aplicado a expresiones lógicas.
- Introducción a la lógica de predicados: noción de predicado (proposición con variables libres). Cuantificadores lógicos \forall (universal) y \exists (existencial) y su significado. Expresión de fórmulas de primer orden con cuantificadores; dominio del discurso. Conversión de enunciados del lenguaje cotidiano al lenguaje formal de predicados.
- Reglas de inferencia lógica y métodos de demostración: inferencias inmediatas en lógica proposicional (Modus Ponens, Modus Tollens, silogismo disyuntivo, etc.) y nociones básicas de inferencia en lógica de predicados (instanciación universal, existencial, etc.). Método de resolución como técnica de inferencia (visión general, preparando el terreno para su uso en programación lógica).

Unidad 5: Álgebra de Boole.

Contenidos:

- Álgebra de Boole: definición y ejemplos (álgebra de conjuntos y álgebra proposicional como casos de álgebras booleanas). Operaciones booleanas básicas: AND (\wedge), OR (\vee), NOT (\neg), etc., y su interpretación en términos de conjuntos (\cap , \cup , complemento) y de lógica proposicional. Polinomio booleano y representación algebraica de funciones lógicas.
- Propiedades del Álgebra de Boole: leyes algebraicas (identidad, idempotencia, complementación, conmutatividad, asociatividad, distributividad, absorción, De Morgan) y dualidad. Relación entre el Álgebra de Boole, el Álgebra de Conjuntos y el Álgebra de Proposiciones (correspondencia entre operaciones y principios comunes).
- Circuitos lógicos digitales: puertas lógicas fundamentales (AND, OR, NOT, NAND, NOR, XOR, XNOR) y construcción de circuitos lógicos combinacionales simples. Electrónica básica del sistema binario (valores binarios 0/1 como falsedad y verdad). Tablas de verdad de circuitos lógicos con múltiples entradas y salidas.
- Simplificación de funciones booleanas: métodos de simplificación algebraica y método gráfico de mapas de Karnaugh para reducir expresiones lógicas complejas a formas más simples y eficientes (minimización de circuitos).
- Conceptos de lógica formal relacionados: satisfacibilidad de fórmulas booleanas (existencia de asignaciones de verdad que hagan verdadera una fórmula); consecuencia lógica y equivalencia lógica entre expresiones booleanas. Conversión de fórmulas booleanas a formas normales estandarizadas (FNC y FND) para facilitar la aplicación de algoritmos de resolución.
- Cláusulas de Horn y métodos de inferencia: introducción a las cláusulas de Horn (formas lógicas con a lo sumo un literal positivo) por su importancia en la lógica computacional.

Método de resolución por refutación: nociones de consistencia y completitud en sistemas lógicos. Skolemización y forma normal prenex (mención conceptual de cómo se preparan las fórmulas de primer orden para la resolución automática). Nota: Estos últimos conceptos se introducen a nivel básico para conectar con la unidad de Programación Lógica, sin profundizar en demostraciones formales complejas.

Unidad 6: Programación lógica.

Contenidos:

- Paradigma de programación lógica y declarativa: diferencias respecto al paradigma imperativo. Ventajas de la programación basada en lógica para ciertos tipos de problemas (búsqueda, IA, backtracking automático).
- Lenguaje Prolog: conceptos básicos de Prolog. Hechos y reglas como forma de representar conocimiento. Sintaxis fundamental de Prolog: átomos, variables, listas y estructuras; uso de reglas (cláusulas de Horn) y consultas.
- Mecanismo de resolución SLD en Prolog: cómo el intérprete de Prolog responde consultas mediante el principio de resolución lógica y retroceso (backtracking). Ejecución de programas lógicos por medio de prueba y refutación de objetivos. Estrategia de búsqueda en profundidad de Prolog, control de la recursión y corte (cut) como control extra (solo una mención si corresponde).
- Programación en Prolog aplicada: definición de relaciones y predicados recursivos. Resolución de problemas sencillos utilizando Prolog (ejemplos: operaciones aritméticas básicas con recursividad, manipulación de listas, árbol genealógico, puzzles lógicos simples). Interpretación de los resultados y depuración básica.
- Conexión con la lógica formal: comprensión de que Prolog opera sobre una base de conocimiento lógica. Importancia de la correctitud y terminación en programas lógicos. Cómo la teoría (lógica de predicados y cláusulas de Horn) sustenta la práctica de la programación lógica. Esta unidad cierra el curso integrando los conceptos de lógica matemática en la herramienta computacional Prolog.

V- Estrategias didácticas a ser implementadas en el proceso de enseñanza aprendizaje. (abarcando actividades de formación e investigación)

Las clases son teóricas prácticas, distribuidas en encuentros presenciales y trabajo independiente del estudiante.

Hay un momento de exposición de los temas en el pizarrón donde atendiendo a las competencias, se introducen los conceptos teóricos matemáticos, favoreciendo la comprensión, de manera que permita interpretar y vincular los mismos a la aplicación a temas específicos de informática. En este sentido se mostrarán algunas aplicaciones en los temas más relevantes.

Luego hay otro momento de trabajo en clase de consulta con el profesor y/o auxiliar, resolviendo los ejercicios prácticos y trabajando también sobre consultas de tipo general.

En algunas ocasiones se piden ejercicios para realizar en clase y entregar para ir llevando un control del avance de los alumnos y para que ellos también adviertan donde aparecen sus dificultades.

Se utilizará la plataforma virtual de enseñanza-aprendizaje como complemento del desarrollo de las unidades.

VI - Estrategias de evaluación.

La evaluación será formativa y procesual, se realizará a través de pruebas (exámenes) que podrán ser escritas, orales o de ejecución que a su vez podrá ser mediante trabajos individuales o grupales. La materia consta de dos pruebas parciales, con un recuperatorio y tres oportunidades para la prueba final.

En estos parciales, así como en el examen final, se evaluarán las competencias alcanzadas a través de actividades de contenido teórico que permitan dar cuenta del avance conceptual en los temas que se han desarrollado, se incorporan preguntas específicas tipo sobre “donde cree Ud. que es aplicable este conocimiento/método matemático” y se refleja en la corrección de las pruebas escritas del alumno.

En algunos temas se trabaja también con ejercitaciones de aplicación en clase, que requieren de un ejercicio de integración de conceptos y que complementan la evaluación a través de los parciales.

Para la obtención de calificaciones parciales y finales se tendrá en cuenta el Reglamento Académico de la universidad.

VII - Actividades de extensión y de responsabilidad social universitaria.

Rige de acuerdo al reglamento de la Universidad y el reglamento interno de la facultad.

VIII - Fuentes bibliográficas

Básica

- Lipschutz, S. & Lipcon, M. (2009). Matemáticas Discretas. Méjico: McGraw - Hill.
- Lipschutz, S. (1992). Teoría de Conjuntos y temas afines. Méjico: McGraw – Hill.
- García Valle, J. Luís. “MATEMÁTICAS ESPECIALES PARA COMPUTACIÓN”.
- Lipschutz, S. (1ª Ed.). Matemáticas para Computación. Méjico: McGraw - Hill.
- I. Bratko, “Prolog Programming for Artificial Intelligence”, Addison-Wesley, (1986).
- J. Kelly, “The Essence of Logic”. Prentice Hall (1997)

Complementaria

- M. Ben-Ari, “Mathematical Logic for Computer Science”, 3rd edition, Springer (2012)
- E. Burke, E. Foxley, “Logic and its Applications”. Prentice Hall (1996)
- C. L. Chang, R. C. T. Lee, “Symbolic Logic and Mechanical Theorem Proving”, Academic Press (1973)
- T. Hortalá, N. Martí, M. Palomino, M. Rodríguez, R. del Vado, “Lógica Matemática para Informáticos. Ejercicios resueltos”, Prentice Hall (2008)
- M. Huth, M. Ryan, “Logic in Computer Science”, Cambridge University Press (2004)
- W. F. Clocksin, C. S. Mellish, "Programming in Prolog. Using the ISO Standard" (5th ed), Springer (2003)
- J. W. Lloyd, “Foundations of Logic Programming” (2nd ed.), Springer-Verlag (1987)
- A. Nerode, R. Shore, “Logic for Applications”, Springer (1997)
- L. Sterling, E. Shapiro, “The Art Of Prolog” (2nd. ed.) The MIT Press, (1994)
- J. Wielemaker, "SWI Prolog Reference Manual. Updated for version 8.2.4", (2021)
- E. Paniagua, J. L. Sánchez, F. Martín, “Lógica computacional”, Thomson, (2003).



- A. V. Aho, J. D. Ullman. "Foundations of Computer Science" (1st ed.). W. H. Freeman & Co., (1994).
- W. Ertel, "Introduction to Artificial Intelligence", Springer (2011)
- L. Zhongwan, "Mathematical Logic for Computer Science", 2nd ed, World Scientific (1998)
- R. Caferra, "Logic for Computer Science and Artificial Intelligence", ISTE Wiley (2011)

Curso exclusivo para fines informativos